KEEx: A Peer-to-Peer Tool for Distributed Knowledge Management

Matteo Bonifacio[1], Paolo Bouquet[1], Paolo Busetta[2], Alberto Danieli[3], Antonia Donà[3], Gianluca Mameli[3], Michele Nori[3]

[1]University of Trento, Dept. of Information and Communication Technology, Trento, Italy
bouquet@dit.unitn.it;bonifaci@dit.unitn.it
[2]ITC-Irst, Povo, Trento, Italy
busetta@itc.it
[3]Distributed Thinking Spa., Rovereto (Trento), Italy
{a.danieli, a.dona, g.mameli, m.nori}@dthink.biz

Abstract. Distributed Knowledge Management is an approach to Knowledge Management based on the principle that the multiplicity (and heterogeneity) of perspectives within complex organizations should not be viewed as an obstacle to knowledge exploitation, but rather as an opportunity that can foster innovation and creativity. Despite a wide agreement on this principle, most current KM systems are based on the idea that all perspectival aspects of knowledge should be eliminated in favour of an objective and general representation of knowledge. In this paper we propose a peer-to-peer system (called KEEx), which embodies the principle above in a quite straightforward way: (i) each peer provides all the services needed to create and organize "local" knowledge from an individual's or a group's perspective, and (ii) social structures and protocols of meaning negotiation are defined to achieve semantic coordination among autonomous peers.

1 Introduction

Distributed Knowledge Management (DKM), as described in [1], is an approach to KM based on the principle that the multiplicity (and heterogeneity) of perspectives within complex organizations should not be viewed as an obstacle to knowledge exploitation, but rather as an opportunity that can foster innovation and creativity.

The fact that different individuals and communities may have very different perspectives, and that these perspectives affect their representation of the world (and therefore of their work) is widely discussed – and generally accepted – in theoretical research on the nature of knowledge. Knowledge representation in artificial intelligence and cognitive science has produced many theoretical and experimental evidences of the fact that what people know is not a mere collection of facts; indeed, knowledge always presupposes some (typically implicit) interpretation schema, which provide an essential component in sense-making (see, for example, the notions of context [2], [3], [4], mental space [5], partitioned representation [6]); studies on the social nature of knowledge stress the social nature of interpretation schemas, viewed as the outcome of a special kind of "agreement" within a community of knowing (see, for example, the notions of scientific paradigm [7], frame [8]), thought world [9], perspective [10]).

Despite this large convergence, it can be observed that the high level architecture of most current KM systems in fact does not reflect this vision of knowledge (see [11] [12] [1] for a detailed discussion of this claim). The fact is that most KM systems embody the assumption that, to share and exploit knowledge, it is necessary to implement a process of knowledge-extraction-and-refinement, whose aim is to eliminate all subjective and contextual aspects of knowledge, and create an objective and general representation that can then be reused by other people in a variety of situations. Very often, this process is finalized to build a central knowledge base, where knowledge can be accessed via a knowledge portal. This centralized approach – and its underlying objectivist epistemology – is one of the reasons why so many KM systems are deserted by users, who perceive such systems either as irrelevant or oppressive [13].

In this paper we propose a peer-to-peer (P2P) architecture, called KEEx, which is coherent with the vision of DKM (See Figure 2). Indeed, P2P systems seem particularly suitable to implement the two core principles of DKM, namely the principle of autonomy (communities of knowing should be granted the highest possible degree of semantic autonomy to manage their local knowledge), and the principle of coordination (the collaboration between autonomous communities must be achieved through a process of semantic coordination, rather than through a process of semantic homogenization) [1]. In KEEx, each community of knowing (or Knowledge Nodes (KN), as they are called in [11]) is represented by a peer, and the two principles above are implemented in a quite straightforward way: (i) each peer provides all the services needed by a knowledge node to create and organize its own local knowledge (autonomy), and (ii) by defining social structures and protocols of meaning negotiation in order to achieve semantic coordination (e.g., when searching documents from other peers).

The paper goes as follows. In Section 2, we describe the main features of KEEx, and argue why they provide a useful support to DKM; in Section 3, we describe the semantic matching algorithm; finally, we draw some conclusions and future work.

2 KEEx: a P2P system for DKM

KEEx is a P2P system which allows a collection of KNs to search and provide documents on a semantic basis without presupposing a beforehand agreement on how documents should be categorized, or on a common language for representing semantic information within the system. In the following sections, we describe the high-level architecture of KEEx, and explain what role each element plays in a DKM vision. KEEx is defined as a collection of peers, each of which represents a KN, namely an individual's or a group's perspective on a given body of knowledge.

KEEx system allows to represent an organization as a set of peers that can group spontaneously, or can be forced to group. Peers can discover and interact with other peers and groups of peers in the P2P network. The set created from all local knowledge, managed from each peer, compose the whole organization's knowledge. In order to do this KEEx solution has different peers:

- **PKM peer** and **Source peer** that represents KN.
- Service peer: **Normalization peer**, to support the knowledge exchange between peers, **Super peer** and **Rendez-Vous peer** to support KEEx system deployment in an organization.

2.1.1 PKM peers

PKM (Personal Knowledge Manager) peer allows a user to manage his own local knowledge, and at the same time to share it in the KEEx P2P network with other users that have a PKM peer. Each PKM peer can play two main roles: *provider* and *seeker*. A PKM peer acts as a provider when it "publishes" in the system a body of knowledge, together with an explicit perspective on it (called *context*, e.g. a topic hierarchy used to categorized local documents [14]); a PKM peer acts as a seeker when it searches for information by making explicit part of its own perspective, and negotiates it with other PKM peers. Below we illustrate the main modules and functionalities.

Document Repository. A *Document Repository* is where each KN stores its own local knowledge. We can imagine a private space in which the KN maintains its document and data, possibly using a local semantic schema (e.g., a file-system structure, or a database schema), or a document management system in order to organize and access them.

Context Repository. Following [15], we define a context as a partial and approximate representation of the world from an individual's or a group's perspective. The reason why we adopt this notion of context is that it provides a robust formal framework (called Local Models Semantics [4]) for modelling both contexts and their relationships.

In order to use contexts in KEEx, we adopted a web-oriented syntax for contexts, called CTXML. It provides an XML-Schema specification of context for document organization and classification. In KEEx, each context plays the role of a category system for organizing and classifying documents, or any other kind of digital information identifiable by a URI, stored in a document repository. Each peer can use more than one context to classify local knowledge and peer's contexts are stored in a *context repository*.

From the standpoint of DKM, contexts are relevant in two distinct senses:

– on the one hand, they have an important role within each KN, as they provide a dynamic and incremental explicitation of its semantic perspective. Once contexts are reified, they become cognitive artifacts that contribute to the process of perspective making [10], namely the consolidation of a shared view in a KN, continuously subject to revision and internal negotiation among its members;

– on the other hand, contexts offer a simple and direct way for a KN to make public its perspective on the information that KN can provide. Therefore, as we will see, contexts are an essential tool for semantic coordination among different KN. It is important to observe that contexts provide only a common syntax for classification structures. Indeed, we could see them as a language for wrapping any classification structure (e.g., like directory systems, databases schemas, web directories). This means that in principle people can continue to work with their preferred document management system, provided it can be wrapped using CTXML.

Context management module. The context management module allows users to create, manipulate, and use contexts in KEEx. The module has two main components:

– **Context editor**: provides users with a simple interface to create and edit contexts, and to classify information with respect to a context. This happens by allowing users to create links from a resource (identified by a URI) to a node in a context. Examples of resources are: documents in local directories, the address of a database access services, addresses of other peers that provide information that a KN wants to explicitly classify in its own context.

– **Context browser**: is part of Seeker component and allows users to navigate contexts in the context repository. The main reasons for navigating a context in KEEx are two. The first is obviously to find documents in the local knowledge repository by navigating the semantic structure. The second, and more important reason, is to build queries. The intuitive idea is that users can make context dependent queries (namely, from their perspective) by selecting a category in one of the available contexts. Once a category is selected, the context browser builds a contextual interpretation of the user's query – by automatically extracting the relevant portion of the context to which the category belongs. The category is then used as a basis for meaning coordination and negotiation with other peers during the search.

2.1.2 Roles of PKM peers in KEEx

Each PKM peer can play two main roles: seeker and provider. Their interactions are described in detail in the following two sections.

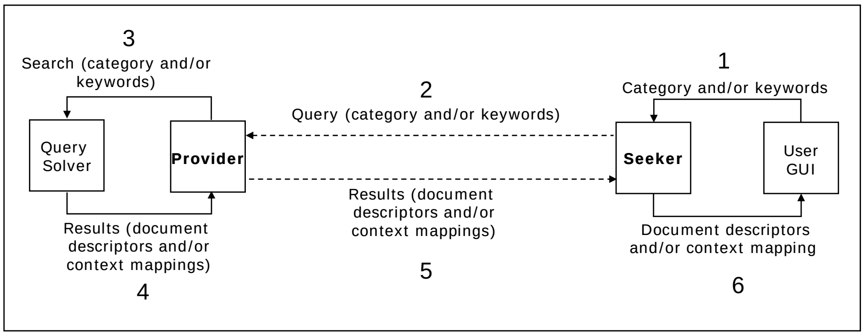

Figure 1: Query

Seeker As a seeker, a PKM peer allows users to search for documents (and other information) from other PKM peers and Communities (see below). The seeker supports the user in the definition of context-dependent queries through the context browser. A query is composed by a query expression and a category. A query expression is a list (possibly empty) of one or more keywords provided by a user; a category is a portion of a context determined by the user's selection. Moreover, the seeker provides the discovery mechanism, used to find resources to which the query has to be sent. The user decides to send the query to some of the available PKM peers and Communities. When the user submits the query, the seeker activates a session associated to that query (there can be only one active session for each seeker). In a session, a seeker can receive several asynchronous replies from the providers which resolved the query (through the meaning negotiation protocol, see below). The results returned to the user are composed by the aggregation of all the results received from the providers; each result is made up of a list of document descriptors (i.e., name of the document, short description, and so on). Each result is presented together with the part of context that the provider has matched against the current query. This relationship between contexts can be used as an opportunity for learning relationships across contexts of different KNs that the seeker can store and reuse for future queries. Finally, if one or more interesting documents are found, the seeker can contact the peers that have the documents and, if possible, download them.

Figure 1 describe the main phases of query process and the interaction between seeker and provider. The number on the arrows represents the sequence of the operations.

The role of seeker is supported by the mechanism of **discovery** that allows the user to discover resources in the P2P network. The user needs to discover PKM peers or Source peers and communities available in the network to contact and query them. A peer advertises the existence of resources by publishing an XML document (called *Advertisement*). In the KEEx system, two types of resources are advertised:

– **PKM peers and Source peers** (see Section 2.1.3) that have a provider service to solve queries. The main elements of the advertisement are a description of the peers contexts, and the peer address to contact it, to send it queries, and to retrieve documents;

– **Communities** (see Section 2.1.4 and 2.2), namely sets of peers that have a community service to solve queries. The community assures that a query sent to a community is propagated to all active peers that are member of the community. In this case the main elements of the advertisement are the community topic, its address and information for joining the community. To discover resources, a peer sends a discovery request to another known peer, or sends a multi-cast request over the network, and receives responses (a list of advertisements) that describe the available services and resources. It is possible to specify search criteria (currently only keywords or textual expression) that are matched against the contents provided by the advertisement related to each peer or community description.

Provider. The provider is the second main role in the KEEx system. It contains the functionalities required to take and resolve a query, and to identify the results that must to be returned to the seeker. When a PKM peer or a Source peer (see Section 2.1.3) receives a query (keywords and/or category), it instantiates a provider (which is configured to use a set of contexts and to provide documents in a given portion of the knowledge repository), in order to solve the query. Seeker can ask to a provider to solve a query in a combination of three ways:

– **Semantic resolution**: using a context matching algorithm [16] (see Section 3), the provider searches for relations between the locally available contexts and the query's category. More specifically, the matching algorithm searches categories whose associated contextual information in the providers contexts matches (in a sense defined in [16]) with the category in the query. If a match is found, the URIs of the resources associated to the provider's context are returned to the seeker, together with a short information on the reason why a semantic match was found.

– **Lexical resolution**: using a keyword-based indexer, the provider searches for the occurrence of specific keywords, combined with logical operator (AND/OR), into the set of documents of the local repository. This type of search is performed into the body of the document (text) and in the name of document (file name).

– **Conceptual resolution**: based on keyword-based search, where keywords are combined with logical operator as lexical resolution, conceptual resolution searches keywords in the contexts labels. The results are documents classified in concepts that satisfy the search parameter.

2.1.3 Source Peers

Source peer is a peer that integrate documental data sources available in an organization classified with taxonomy (intranet site, content management tools, search engine, etc). It is different from a PKM peer, managed from a user that share is own local knowledge with other users, because the Source peer is the way which an organization share the "institutional knowledge", with the users (PKM peers). In term of roles, Source peer in the KEEx P2P network plays only the provider role and can be part of a community.

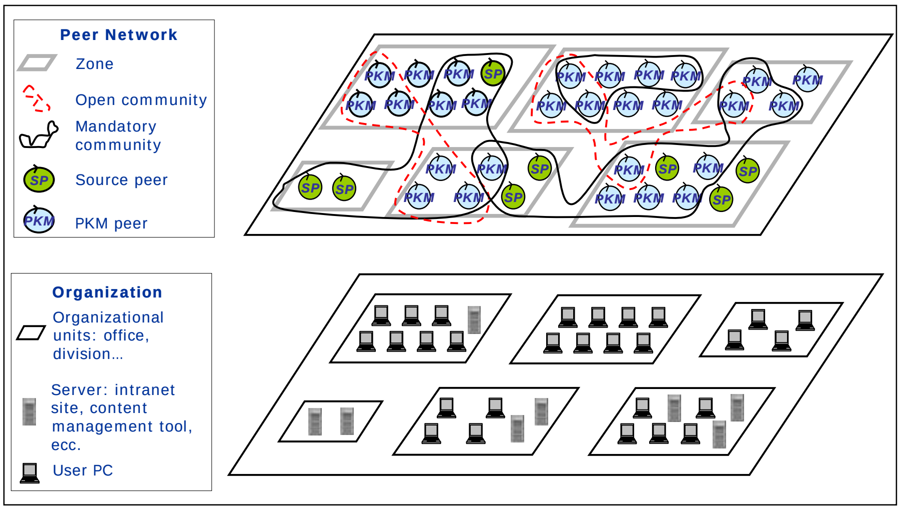

Figure 2: Organization vs Peer Network

2.1.4 Community

In the KEEx system a **community** is a group of peers that agree to appear as a unique entity to PKM peers that perform a search. Each community can be though as a "social" aggregation of PKM peers and Source peers that display some synergy in terms of content (e.g., as they provide topic-related content, or decided to use the same linguistic resource to create a common "vocabulary", thus providing more homogeneous and specific answers), quality (certify content) or access policies (certify members). A peer take part of a community sharing a sub set of local knowledge (sub set of contexts) that is coherent with the community themes.

PKM peers can send queries directly to communities, and the query is managed internally at the community. The query is distributed to all the members of the community. The result is not the same as if the query was sent directly to each member of the community, the difference being that PKM peers explicitly answer as members of the community using a sub set of local knowledge coherent with the whole community knowledge.

A community in KEEx could be **open**, a PKM peer can propose a open community to the other PKM peers and each PKM peer can decide to join or leave this community. Otherwise there are **zone** or **mandatory community** that are created by organization (see Section 2.2) using Super peer, one of the Service Peer provides by KEEx.

2.2 Service Peers

KEEx provides three service peers which have respectively important roles in supporting knowledge exchange (Normalization peer), supporting organization with a set o administration functionalities of KEEx P2P network (Super peer), and supporting communication (Rendez-Vous peer). The service peers are described in the following section.

Normalization peer: this peer allows to KEEx P2P network (PKM peers and Source peers) to use the contexts normalization service, This service allows to perform a linguistic normalization (e.g., deleting stop words, tokenizing, part-of-speech tagging, etc.) on user defined contexts, to use knowledge from an external linguistic resource (e.g., WordNet) to add semantic information to the categories in a context, and to discover semantic relations across different contexts.

Super peer: this peer provides some configuration and administration functionalities in order to support organization to deploy KEEx system, in the best way, to obtain the P2P network that better represent the physical and functional network organization.

PKM peers and Source peers are instantiated from the Super peer inside to **zones**, partitions of the network, from a logical point of view, that maps in the best way the organization (See Figure 2). If a peer does not have a zone assigned, or there are no zones created, peers are part of a default group, called **main group**. For instance a zone can correspond to an organization unit (an office, a division, etc), so the main group represents the whole organization. A PKM peer or a Source peer are part of a unique group (main group or zone) and depending configuration can discover and interact with other peers, zone or communities.

Another network partition can be done from communities, that in KEEx can be open (see Section 2.1.4) or mandatory. A **mandatory community** has the same philosophy of an open community, but it is created from a Super peer, that decides which PKM peer or Source peer have to be member, and forces them to join mandatory community created. As in open communities a PKM peer or Source peer can be part to one or more mandatory communities and the community can have, as members, peers from different zones.

Mandatory communities can be used from an organization to group users (PKM peers) and to give access, through Source peer, to the institutional knowledge contextually to a group. This could be the way that allows the organization to map in the P2P network dynamic or temporary work groups (as project team).

Mandatory communities can be derived from open communities, created and participated from PKM peers. The organization, monitoring P2P network, can become aware that an open community is useful from an organization point of view. It could think to institutionalize, creating a mandatory community that replaces the open community.

Rendez-Vous peer:in the KEEx system this type of peer does not interact directly with other peers, but supports the communication between them in some particular configurations of physical networks. Figure 3 depicts an example of how the KEEx network can be mapped on the physical one.

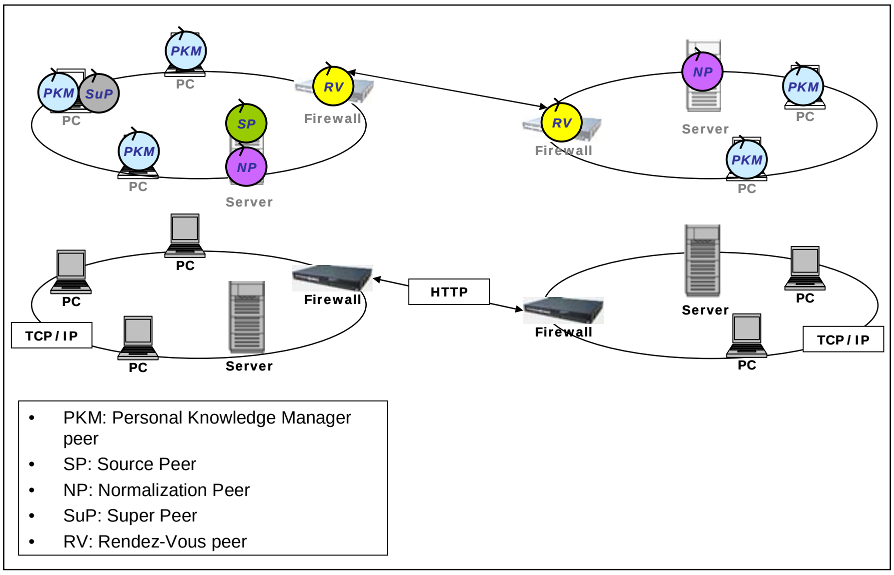

Figure 3: Network topology

3 The Semantic Matching Algorithm

A system like KEEx critically depends on the availability of semantic mappings across autonomously developed schemas (here called contexts) that are used to organize and retrieve locally available data (e.g., classification schemas, database schemas, directory structures). Today, mappings are still largely done by hand, in a labour intensive and error-prone process. As a consequence, semantic integration issues have now become a key bottleneck in the deployment of a wide variety of information and knowledge management applications. The high cost of this bottleneck has motivated numerous research activities on methods for describing mappings, manipulating them, and generating them automatically (or semi automatically).

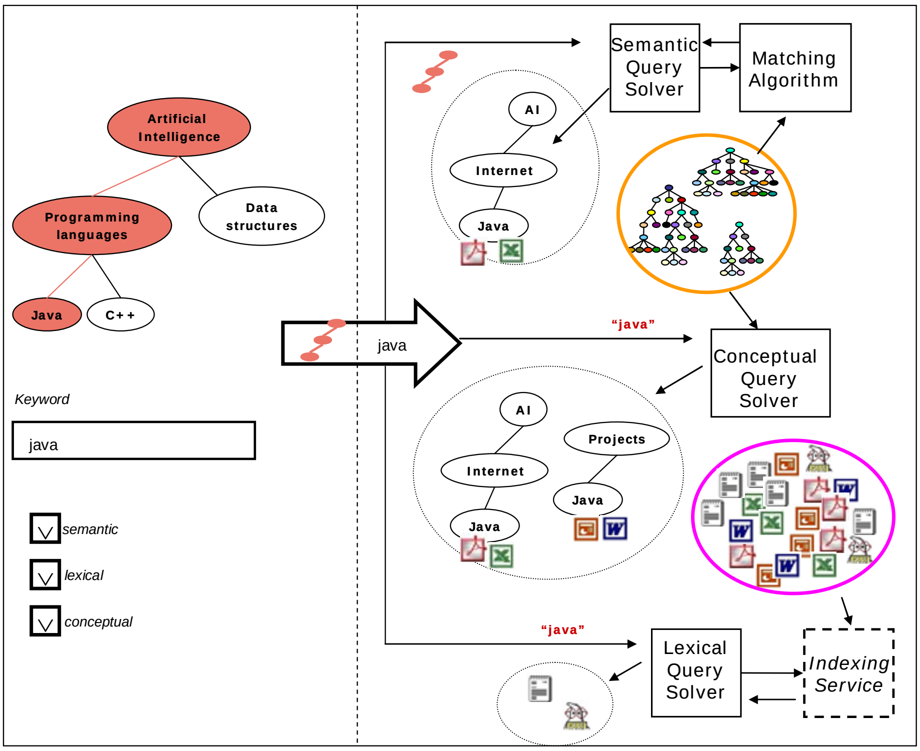

Figure 4: Query execution example

In P2P systems, different techniques for generating mappings have been developed (see [17] for a survey of the main techniques), based on graph matching algorithms, instance based matching, ontology-based mediation. A key feature of KEEx is that it embodies a state of the art algorithm for automatically discovering semantic relations across autonomously developed schemas. The algorithm, called CTXMatch [18], takes two schemas (in practice, a seeker's schema and a provider's schema) in input and returns a collection of mappings across pairs of nodes belonging to different schemas; mappings are then used by KEEx to return a collection of links to potentially relevant documents across heterogeneous schemas The aim of this section is to provide an high-level description of how the algorithm works.

CTXMatch produces the required mapping in two main steps: *semantic explicitation* and *semantic comparison*. See [18] for details. Intuitively, the first phase uses external resources (in the current version, WordNet) to make explicit the concept associated to each node of a schema (for example, if the schema contains a path like

IMAGES/TUSCANY/FLORENCE, then the output of this phase would be a logical formula that approximates the meaning “Images of Florence, the Tuscan city”); the second phase starts from the logical formulae associated to nodes of different schemas and – via logical reasoning – checks whether there is a model-theoretic relation between them (this, in turn, depends on a body of background knowledge about the concepts occurring in the two formulae). If a relation is deduced (e.g., that one of the two concepts is more general than or equivalent to or disjoint from the other concept), then the relation is returned as a mapping between the two nodes. For example, if the concept associated to the path IMAGES/TUSCANY/FLORENCE is semantically compared with the concepts associated to the path IMAGES/ITALY/FLORENCE, the semantic relation returned is logical equivalence, as background knowledge on the fact that Tuscany is in Italy and Florence is in Tuscany allows the system to conclude that the concept “Images of Florence, the Tuscan city” is equivalent to the concept of “Images of Tuscany, the Italian city” (in other words, the documents that a user would classify under the first concept are the same as those that the same user would classify under the second, and *vice versa*).

Figure 4 gives an example of query composed combining all the 3 types of query available in KEEx. The seeker (on the left) select a category on a context (the red path) and specify a keyword (java) to be used both for lexical and conceptual search. The right side of the picture represent the provider: the three solvers are instanciated:. The lexical one use the repository of documents and the indexer to find documents containing the “java” keyword. The semantic and the conceptual one use the repository of context.

It is important to realize that different PKM peers might use different linguistic and ontological knowledge to derive mappings between their local contexts and other PKM peers’ contexts. Therefore, the notion of mapping we use in KEEx is intrinsically directional, as it depends on the knowledge locally available to the PKM peer that derives a mapping. In other words, also mapping are perspectival, which seems coherent with the vision of DKM adopted in this paper.

The algorithm has been successfully tested on real examples, like web directories (Google vs. Yahoo) and commercial catalogs of goods and services.

4 Conclusions and Research Issues

In this paper, we argued that technological architectures, when dealing with processes in which human communication is strongly involved, 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. Here, we suggest that P2P infrastructures are especially suitable for KM applications, as they naturally implement meaning distribution and autonomy. It is perhaps worth noting at this point that other research areas are moving toward P2P architectures.

In particular, we can mention the work on P2P approaches to the semantic web [19], to databases [20], to web services [21]. We believe this is a general trend, and that in the near future P2P infrastructure will become more and more interesting for all areas where we can't assume a centralized control.

A number of research issues need to be addressed to map aspects of distributed cognition into technological requirements. Here we propose three of them:

– **social discovery and propagation**: in order to find knowledge, people need to discover who is reachable and available to answer a request. On the one hand, broadcasting messages generates communication overflow, on the other hand talking just to physically available neighbour reduces the potential of a distributed network. A third option could be for a seeker to ask his neighbours who they trust on a topic and, among them, who is currently available. Here the question is about social mechanisms through which people find – based on 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;

– **learning:** when the matching algorithm finds a semantic correspondence between concepts of different contexts, the provider can store this information for future reuse. This information is represented as a semantic "mapping" between concepts (see [14]), and can be used in three ways:

1. when the Peer receives a query from a seeker, it can reuse stored mappings to facilitate (and possibly to avoid executing) the matching algorithm;
2. a provider can use the existing mapping to forward a query to other peers that have a semantic relation with the category in the query;
3. the seeker can search into the available mappings to suggest the user a set of providers with which it already had previous interactions and are considered qualified with respect to the semantic meaning of the category selected in a query. Using this mechanism, the PKM peer network defines and increases the number and quality of the semantic relations among its members, and becomes a dynamic web of knowledge links;

– **building communities**: if we consider communities as networks of people that, to some extent, tend to share a common perspective [10], mechanisms are needed to support the bottom-up emergence of semantic similarities across interacting KNs.

Through this process, which are based on meaning negotiation protocols, people can 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 neighbourhoods to support social discovery and propagation.

References

[1] M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management: managerial and technological implications. *Novatica and Informatik/Informatique*, III(1), 2002.

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

[3] P. Bouquet. *Contesti e ragionamento contestuale. Il ruolo del contesto in una teoria della rappresentazione della conoscenza*. Pantograph, Genova (Italy), 1998.

[4] C. Ghidini and F. Giunchiglia. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artificial Intelligence*, 127(2):221–259, April 2001.

[5] G. Fauconnier. *Mental Spaces: aspects of meaning construction in natural language*. MIT Press, 1985.

[6] J. Dinsmore. *Partitioned Representations*. Kluwer Academic Publishers, 1991.

[7] T. Kuhn. *The structure of Scientific Revolutions*. University of Chicago Press, 1979.

[8] I. Goffaman. *Frame Analysis*. Harper & Row, New York, 1974.

[9] D. Dougherty. Interpretative barriers to successful product innovation in large firms. *Organization Science*, 3(2), 1992.

[10] J.R. Boland and R.V.Tenkasi. Perspective making and perspective taking in communities ofknowing. *Organization Science*, 6(4):350–372, 1995.

[11] M. Bonifacio, P. Bouquet, and R. Cuel. Knowledge Nodes: the Building Blocks of a Distributed Approach to Knowledge Management. *Journal of Universal Computer Science*, 8(6):652–661, 2002. Springer Pub. & Co.11

[12] M. Bonifacio, P. Bouquet, and A. Manzardo. A distributed intelligence paradigm for knowledge management. In*Working Notes of the AAAI Spring Symposium Series 2000 on Bringing Knowledge to Business Processes*. AAAI, March 18-20 2000.

[13] G. C. Bowker and S. L. Star. *Sorting things out: classification and its consequences*. MIT Press., 1999.

[14] P. Bouquet, A. Donà, and L. Serafini. ConTeXtualized local ontology specification via ctxml. In *MeaN-02 – AAAI workshop on Meaning Negotiation*, Edmonton, Alberta, Canada, 2002.

[15] M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual Reasoning Distilled. *Journal ofTheoretical and Experimental Artificial Intelligence*, 12(3):279–305, July 2000.

[16] B. Magnini, L. Serafini, and M. Speranza. Linguistic based matching of local ontologies. In P. Bouquet, editor, *Working Notes of the AAAI-02 workshop on Meaning Negotiation. Edmonton (Canada)*. AAAI, AAAI Press, 2002.

[17] E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. VLDB Journal, 10(4), 2001.

[18] P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: a new approach and an application. In K. Sycara, editor, Second International Semantic Web Conference (ISWC-03), Lecture Notes in Computer Science (LNCS), Sanibel Island (Florida, USA), October 2003.

[19] M. Arumugam, A. Sheth, and I. Budak Arpinar. The peer-to-peer semantic web: A distributed environment for sharing semantic knowledge on the web. In *WWW2002 Workshopon Real World RDF and Semantic Web Applications. Honolulu, Hawaii (USA)*, 2002.

[20] F. Giunchiglia and I. Zaihrayeu. Making peer databases interact – a vision for an architecture supporting data coordination. In *6th International Workshop on Cooperative Information Agents (CIA-2002), Universidad Rey Juan Carlos, Madrid, Spain, September 18 - 20, 2002*, 2002. Invited talk.

[21] M.P. Papazoglou, J. Jang, and B.J.Kraemer. Leveraging web-services and peer-to-peer networks. October 2002.