Journal of Universal Computer Science, vol. 8, no. 6 (2002), 652-661
submitted: 4/2/02, accepted: 3/5/02, appeared: 28/6/02 © J.UCS

Knowledge Nodes: the Building Blocks of a Distributed Approach to Knowledge Management

Matteo Bonifacio
(University of Trento, Italy
bonifacio@science.unitn.it)

Paolo Bouquet
(Department of Information and Communication Technologies
University of Trento, Italy
bouquet@dit.unitn.it)

Roberta Cuel
(Department of Computer and Management Sciences
University of Trento, Italy
rcuel@cs.unitn.it)

Abstract: In this paper, we criticise the objectivistic approach that underlies most current systems for Knowledge Management. We show that such an approach is incompatible with the very nature of what is to be managed (i.e., knowledge), and we argue that this may partially explain why most knowledge management systems are deserted by users. We propose a different approach - called distributed knowledge management - in which subjective and social (in a word, contextual) aspects of knowledge are seriously taken into account. Finally, we present a general technological architecture in which these ideas are implemented by introducing the concept of knowledge node.

Key Words: Distributed Knowledge Management, Knowledge Nodes, Context, Epistemology

Category: H

1 Introduction

Knowledge, in its different forms, is increasingly recognised as a crucial asset in modern organisations. Knowledge Management (KM) is referred to the process of creating, codifying and disseminating knowledge within complex organisations, such as large companies, universities, and world wide organisations.

Most KM projects aim at creating large, homogeneous knowledge repositories, in which corporate knowledge is made explicit, collected, represented and organised, according to a single - supposedly shared - conceptual schema. Such a schema is meant to represent a shared conceptualisation of corporate knowledge, and thus to enable communication and knowledge sharing across an entire organisation. The typical outcome of this kind of project is the creation of an Enterprise Knowledge Portal (EKP), a (web-based) interface which provides a unique access point to corporate knowledge.

In the paper, we argue that this approach reflects an objectivistic epistemology, as it presupposes that all contextual, subjective, and social aspects of knowledge can be eliminated in favour of an objective and general codification, and that this abstract and general knowledge can be shared and reused independently from the individuals or the organisational units (i.e. teams, work-groups, communities) in which it was created. If, on the one hand, this assumption is coherent with traditional organisational models and paradigms of control, on the other hand, it seems incompatible with many theories of knowledge, where subjective and social aspects of knowledge are viewed as essential features. We argue that this incoherence between the high level architecture of KM systems and the nature of knowledge may explain, at least partially, why many KM systems are deserted by users. We follow a different approach – called *Distributed Knowledge Management* (DKM) – in which subjectivity and sociality are viewed as a potential source of value, rather than as a problem to overcome [6]. Our proposal is to model an organisation as a "constellation" of *knowledge nodes*, namely autonomous and locally managed knowledge sources. In this approach, a KM system becomes a tool that must support two qualitatively different processes: the autonomous management of knowledge which is locally produced within a single knowledge node (*principle of autonomy*), and the coordination of the different knowledge nodes without a centrally defined semantics (*principle of coordination*). In the last part of the paper, we describe the high level architecture of a system which supports this distributed approach from a technological point of view.

2 Traditional approach to designing KM systems

If we abstract away some inessential differences, most KM projects share a pattern which involves the following steps (see e.g. [9]):

- the installation of corporate-wide Intranets to ensure physical accessibility to information;
- the design of a corporate language and of knowledge maps, which are used to represent corporate knowledge in a standard and common way, and to create semantically homogeneous and context-independent knowledge repositories (the corporate knowledge base, or KB);
- the creation/support of informal communities that represent the place where "raw" knowledge is produced through spontaneous and emerging social interaction of company peers (typically, these communities are materialised as "virtual communities" through the adoption of computer supported cooperative tools, such as groupware applications);
- the creation of a new role, the knowledge manager, whose goal is to support and facilitate the interaction within and across organisational units;
- the design of contribution processes which enable community members to explicit their tacit knowledge through the codification in the corporate language;
- the construction of an *Enterprise Knowledge Portal* (EKP), which provides a unique, simple interface through which people can contribute to the KB, socialise, and retrieve information.

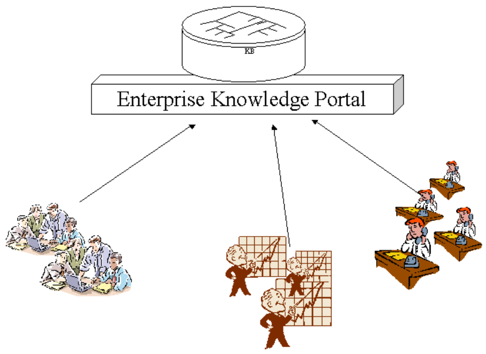

Figure 1: The traditional KM approach

The typical outcome is a system like that depicted in Figure 1.

Such an architecture is generally based on technologies like *content management tools* (text miners, search engines, and so on), which are used to produce a shared view (either implicit – e.g., clusters, neural nets – or explicit – e.g., ontologies, taxonomies) of the entire collection of corporate documents; common *formats* (such as HTML, XML, PDF), used to overcome the syntactic heterogeneity of documents from different knowledge sources; chats and discussion groups, used to satisfy the need of social interaction.

3 Problems with traditional KM approach

Despite the claim of business operators and software vendors that this approach is the right answer to the needs of managing corporate knowledge, KM systems are often deserted by users, who instead continue to produce and share knowledge as they did before, namely through structures of relations and processes that are quite different from those embedded within KM systems (many case studies are analysed in literature, in particular, the case of a worldwide consulting firm described in [5]).

In [6], it is argued that this situation does not originate from technological problems, but from an inadequate epistemological model, which is coherent with a traditional paradigm of managerial control, but is in contradiction with the deep nature of knowledge. The way most KM systems are designed embodies an objectivistic view of knowledge, a view according to which knowledge can be represented in an objective form, which is independent from all those subjective and contextual elements that are typical of "raw" knowledge (namely, knowledge in its original form). However, a large number of researchers, working in different disciplines, convincingly argued against this objectivistic view. The basic argument is that knowledge is not a simple "picture" of the world, as it always presupposes some degree of interpretation. This means that a fact is not a fact, unless we have a schema that allows us to give it an interpretation; and that different schemas produce different interpretations of the "same" fact. This aspect of knowledge was studied from different perspectives in different disciplines. Some authors stress the cognitive nature of interpretation schemas, where a schema is viewed as an individual's perspective on the world (see, for example, the notions of context [17, 7, 13], mental space [12], partitioned representation [10]); others stress their social nature, where a schema is thought of as the outcome of a special form of "agreement" within a community of knowing (see, for example, the notions of paradigm [15], frames [14]), thought worlds [11]). In general, interpretation schemas are only partially reducible to each others. Indeed, to get a complete reduction, one should have a perfect understanding of other agents' schemas, and many evidences seem to indicate that in general this is impossible for an agent with limited resources (see [19]).

In our opinion, this epistemological view has two important consequences for designers of KM systems:

1. any approach to designing KM systems which requires to organise corporate knowledge in a supposedly objective picture of the world is in fact trying to force a privileged schema (e.g., that of the Chief Knowledge Officer) onto people who may not share (and thus understand) that view;
2. any approach which disregards the plurality of interpretation schemas is bound to trouble, as the outcome will be perceived by users either as irrelevant (there is no deep understanding of the adopted schema) or as oppressive (there is no agreement on the unique schema, which is therefore rejected).

Therefore, the concept of *absolute knowledge*, which refers to an ideal, objective picture of the world, leaves the place to the concept of *local knowledge*, which refers to different, partial, approximate, perspectival interpretations of the world [2], generated by individuals and within groups of individuals (e.g. organisational units) through a process of meaning negotiation, namely a process of "distilling" a schema which makes sense for that unit. At an organisational level, each local knowledge appears as the synthesis between a collection of statements and the schemas that are used to make sense of them. Local knowledge is then a matter that was (and is continuously) socially negotiated by people that have an interest in building a common perspective (perspective making [3], or single loop learning [1]), but also in understanding how the world looks like from a different perspective (perspective taking [3] or double loop learning [1]). Therefore, rather than being a monolithic picture of the world as it is, knowledge appears as a heterogeneous and dynamic system of multiple "local knowledge systems" that live in the interplay between the need of sharing a perspective within an organisational unit (to incrementally improve performance) and of meeting different perspectives (to sustain innovation).

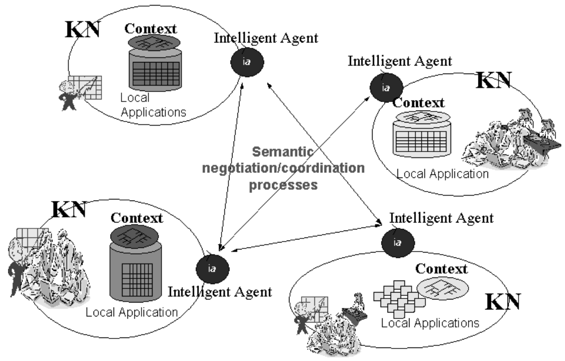

Figure 2: DKM architecture

4 Knowledge nodes and DKM

In this section, we extend the approach of DKM (as presented in [6]) with the concept of *knowledge node*, which provides a useful abstraction of organisational units from a designing perspective.

DKM is based on two very general principles:

1. **Principle of Autonomy**: each organisational unit should be granted 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*, that is the possibility of choosing the most appropriate conceptualisation of what is locally known (for example, through the creation of their own knowledge maps, which in [6] are called *contexts*);
2. **Principle of Coordination**: each unit must be enabled to exchange knowledge with other units not through the adoption of a single, common interpretation schema (this would be a violation of the first principle), but through a mechanism of mapping other units' contexts onto its context from its own perspective (that is, by projecting what other units know onto its own interpretation schema).

In this view, a DKM system must support two qualitatively different processes: the autonomous management of knowledge locally produced within a single unit, and the coordination of the different units without a centrally defined semantics. The resulting high level architecture of a system for DKM is depicted in Figure 2[1].

If a complex organisation can be thought as a constellation of units, an important issue is how this "socially distributed architecture" can be modelled to design an "ar-

[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.

chitecturally distributed" computer-based system for supporting KM processes. To this end, we introduce the concept of knowledge node (KN) as the building block of a model for designing DKM systems.

A KN can be viewed as the reification of organisational 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 local interpretation schemas (perspectives on the world). In other words, each KN represents a *knowledge owner* within the organisation, namely an entity (individual or collective) which has the capability of managing its own knowledge both from a conceptual and a technological point of view. Notice that most often knowledge owners within an organisation are not formally recognised, and thus their semantic autonomy emerges in the creation of "artifacts" (e.g. databases, web sites, collection of documents, archives, practices, and so on) which are not part of the official information system. In what follows, we describe how we applied the concept of KN to design the prototype of a document management application within an Italian national bank.

The back-end activity of the bank is organised in different offices (e.g. information technology, marketing, finance), each with different (but partially related) tasks. Currently, employees within each office share documents by publishing them on a locally shared directory, called "public", which is accessible only to people working in that office. In addition, there is a global public directory, which is used to share documents across the entire bank. Interestingly enough, these shared directories do not have a predefined structure, which means that each employee can add new folders at any depth in the file system in order to provide a sort of classification to the shared documents. In other words, local and global public directories are created, organised, and populated through the active participation of a large number of workers. From our perspective, each resulting directory structure can be viewed as a sort of local classification which represents its creator's viewpoint, and provides a distinctive perspective on the stored documents [4].

To identify the KNs (i.e. semantically autonomous organisational units), we investigated (mainly through interviews) the process through which the directory structures on the publicly accessible directories are created, maintained, and used. We discovered that many of these structures have a group of users who – having common problems, using a common language, and focusing on similar objectives – share an interpretation schema (indeed, we found some very well-defined directory structures, which were devised precisely as a more or less stable categorisation of information). More interestingly, these schemas do not reflect simply the office structure, but also some inter-office projects (e.g. a multi-channel project), namely projects that involve workers from different offices. The problem is that, using the current system, the only way to share documents among people who work for the same project but in different offices is to use the global public directory, which means that project documents are made accessible to everybody in the bank (alternatively, people exchange documents by e-mail, which is a very inefficient solution).

From a designing perspective, this situation led us to represent each office and project as a distinct KN, each of which needs to share documents within and across KNs. This is an ideal test bed for a DKM system, as it is a clear instance of a situation in which the principles of autonomy and coordination naturally apply. Indeed, for our document sharing prototype, we designed an architecture (depicted in Figure 2), which instantiates the architecture proposed for DKM in [4]. Here's a short description.

Each KN has the following high level architecture:

Local applications. An important assumption of DKM is that different organisational units tend to (autonomously) develop working tools that suit their internal needs, and that the choice aand usage of these tools is a manifestation of their semantic autonomy. This may be for historical reasons (for example people use old legacy systems that are still effective), but also because different tasks may require the use of different applications and formats data (i.e. text documents, audio/movies,) to work out effective procedures, and to adopt a specific and often technical language. In Figure 2, examples of local applications are software systems, procedures and artifacts (i.e. relational databases, groupware and content management tools, shared directories). Even if technologies and data formats are the same for two or more KN, the appropriation (i.e. the local understanding of specific uses in a given setting [18]) of each KN can be very different, depending – among other things – on the local interpretation schema.

In our case study, local applications are extremely simple: basically MS Office applications plus the local and global public directory structures on different shared file systems. In this case, the form of appropriation is reflected in two aspects: the different organisation of the directory structures (which partially represent a KN's semantic schemas), and the different processes of contribution to these directories that are implemented within each KN.

Contexts. In [6], a context is defined as an explicit representation of a community's perspective[2]. In simple situations, it can be the category system used to classify documents; in more complex scenarios, it can be an ontology, a collection of guidelines, or a business process. We can say that a context is the "reification" of a KN's perspective, and its continuous, autonomous management is a powerful way of keeping a unit's perspective alive and productive. From a designing perspective, contexts may be created from scratch, but more often can be "extracted" from semantic information embedded in the usage of local applications. These extraction

[2] The notion of context from which we started was formally defined and studied in a formal setting in [13] and in [2]. The basic intuition is that a context is a partial and approximate representation of the world from a given perspective. As such, a context can be formalized as a local theory which stands in some relationship (called a *compatibility relation*) with other local theories of the world. Such a relationship captures the fact that contexts, though autonomous, can "overlap", namely can represent the "same" portion of the world, and therefore there must be a degree of semantic coordination among them.

processes can be supported by tools like text miners, content management tools, and other similar technologies. Notice that, in a sense, the purpose of these technologies is partially re-interpreted: from instruments for the creation and management of global interpretation schemas, to instruments for the creation and management of local schemas.

In the bank, we found that many contexts could be extracted from sub-structures of the local and global public directories. As the prototype mainly aims at document sharing, we decided to use these structures as simple categorisations, which are used in each KN to provide a perspective on the classified documents. Technically, we represented contexts in a *Context Markup Language* (see [8] for more details), which allows to represent simple conceptualisations in an XML-based format. We also provided a *context manager*, namely a simple interface that allows authorized users to browse and edit the contexts of their KN, and to use local contexts to compose semantically enriched queries (see below for more details).

The extraction process can be made automatically, but we believe that it is strategic and necessary that knowledge owners take part of context extraction. Therefore we create a context editor that helps users to mange (i.e. add new item, delete, modify) them contexts.

Agents. In the proposed architecture, a software agent is associated to each KN (denoted as "ia" in Figure 2). Each agent "knows" (i.e., has direct access to) the context of its KN. Agents have two main functions: supporting the users of a KN to compose outgoing queries, and answering incoming queries from other KNs. The intuition is the following. Since each context represents a KN's perspective on some domain, it can be used not only to classify local documents, but also to "explain" to the agents of other KNs what is the semantic content of a query.

To give an idea of how documents search and sharing works, imagine that someone in the KN associated to the information technology (IT) office needs to retrieve documents about a software, say about anti-virus updating. Suppose that, in the context associated to the IT office, classifies documents about anti-virus software under the following semantic structure: "*/office-activities/software/antivirus/antivirus-up-date/manuals*". Through a context editor, the user can associate this (semantic) structure to its query, this way making clear, for example, that she's trying to find technical documents about anti-virus updates, and not about marketing issues. Now imagine that the agent of the KN associated to the marketing office gets the query. Its document repository contains documents about anti-virus, but they are classified under the category "*/products/software/-antivirus/market-reports/last*". Of course, we'd like the agents to be able to decide that the associated documents are unlikely to match the category of the IT context. On the contrary, if the agent of the multi-channel project gets the query, and the local context classifies documents under the structure "*/documents/security-system/anti-virus/antivirus-up-date/manuals*", then we'd like the agents to agree that those documents are potentially relevant. This "semantic matching" between contexts is performed through a protocol that "mimics" the process of "meaning negotiation" enacted by humans when trying to understand each others (for example, when we ask to an expert to give us references to relevant papers). Technically, agents use a matching algorithm between contexts whose preliminary description is provided in [16].

5 Conclusions

In this paper, we extended the framework of DKM (as presented in [6]) with the concept of KN. A KN is an abstraction from a designing perspective which provides the building blocks of a technological infrastructure for DKM. The idea of a KN is that it "reifies" an organisational unit which exhibits some degree of semantic autonomy, namely the capability of producing autonomous interpretation schemas. We believe that this capability is mostly disregarded in traditional KM systems, which tend to embody a "centralized" approach to knowledge representation and management, in other words an approach in which local perspectives are abstracted away and replaced by centrally designed semantic structures. As we argued elsewhere, we think this is one of the reasons why many KM systems look like cathedrals in the desert. Indeed, most often the problem is not in the technology, but in the epistemological and organisational assumptions which are implicitly made in the way a technology is implemented in a social system.

We suggested that KNs must be explicitly recognised, and granted some degree of autonomy at different levels: technologically (i.e. in the appropriation of local applications), syntactically (e.g. different information formats, and representation systems) and, most important, semantically (different organisational units must be allowed to generate and use different interpretation schemas). Indeed, autonomy – and thus heterogeneity – should no longer be seen as a potential threat for an organisation, but as a potential source of new insights and innovation. Indeed, most innovation processes are triggered by the encounter of different perspectives, as this generates a discontinuity in traditional and incremental organisational learning paths.

Defining the boundaries of semantically autonomous organisational units (and thus of KNs) can be hard work. Individuals that are part of an organisational unit are social interconnected with others to solve different objectives, often are part of two or more units, and use more than one contexts. Indeed each organisational unit differs from others for characteristics that are strictly dependent to the organisational strategy, organisational climate, and organisational competencies. It seems to us that a critical aspect of the DKM system is to define appropriate criteria that allows observers to analyse an organisation into KNs. In the paper, we showed how we did this analysis in a simple case, a national bank, where the objective was to design and implement a prototype of a document sharing application. However, we are aware that this analysis may prove to be much harder for more complex organisations, or for more complex KM applications.

Acknowledgements

This paper is part of the EDAMOK project, funded by the Provincia Autonoma di Trento for the years 2001-2003. We'd like to thank all the members of the project team, and in particular those who work on the application scenario, namely Alessandra Molani, Gianluca Mameli, and Elena Andretta.

References

1. C. Argyris. Teaching smart people how to learn. *Harvard business review*, 1999.
2. M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual Reasoning Distilled. *Journal of Theoretical and Experimental Artificial Intelligence*, 12(3):279–305, July 2000.
3. J.R. Boland and R.V.Tenkasi. Perspective making and perspective taking in communities of knowing. *Organization Science*, 6(4):350–372, 1995.
4. 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.
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. Submitted.
6. M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. managerial and technological implications. *Informatik – Informatique*, 1/2002.
7. P. Bouquet. *Contesti e ragionamento contestuale. Il ruolo del contesto in una teoria della rappresentazione della conoscenza*. Pantograph, Genova (Italy), 1998.
8. 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
9. T. H. Davenport, D. W. De Long, and M. C. Beers. Successful knowledge management projects. *Sloan Management Review*, 39(2), Winter 1998.
10. J. Dinsmore. *Partitioned Representations*. Kluwer Academic Publishers, 1991.
11. D. Dougherty. Interpretative barriers to successfull product innovation in large firms. *Organization Science*, 3(2), 1992.
12. G. Fauconnier. *Mental Spaces: aspects of meaning construction in natural language*. MIT Press, 1985.
13. C. Ghidini and F. Giunchiglia. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artificial Intelligence*, 127(2):221–259, April 2001.
14. I. Goffamn. *Frame Analisys*. Harper & Row, New York, 1974.
15. T. Kuhn. *The structure of Scientific Revolutions*. University of Chicago Press, 1979.
16. B. Magnini, L. Serafini, 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
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. Orlikowski and D. C. Gash. Technological Frames: Making Sense of Information Technology in Organization. *ACM Transactions on Information Systems*, 12(2):174–207, April 1994.
19. W. V. Quine. Propositions. In *Ontological Relativity and other Essays*. Columbia University Press, New York, 1969.