



# Open Research Online

---

The Open University's repository of research publications and other research outputs

## Initiating organizational memories using ontology network analysis

### Conference or Workshop Item

#### How to cite:

Kalfoglou, Yannis; Alani, Harith; O'Hara, Kieron and Shadbolt, Nigel (2002). Initiating organizational memories using ontology network analysis. In: ECAI 2002 Workshop on Knowledge Management and Organizational Memories, 21-26 Jul 2002, Lyon, France.

For guidance on citations see [FAQs](#).

© 2002 The Authors

Version: Accepted Manuscript

Link(s) to article on publisher's website:  
<http://www.eccai.org/ecai.shtml>

---

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

---

[oro.open.ac.uk](http://oro.open.ac.uk)

# Initiating Organizational Memories using Ontology Network Analysis

Yannis Kalfoglou<sup>1</sup>, Harith Alani<sup>1</sup>, Kieron O'Hara<sup>1</sup>, Nigel Shadbolt<sup>1</sup>

**Abstract.** One of the important problems in organizational memories is their initial set-up. It is difficult to choose the right information to include in an organizational memory, and the right information is also a prerequisite for maximizing the uptake and relevance of the memory content. To tackle this problem, most developers adopt heavy-weight solutions and rely on a faithful continuous interaction with users to create and improve its content. In this paper, we explore the use of an automatic, light-weight solution, drawn from the underlying ingredients of an organizational memory: ontologies. We have developed an ontology-based network analysis method which we applied to tackle the problem of identifying communities of practice in an organization. We use ontology-based network analysis as a means to provide content automatically for the initial set-up of an organizational memory.

## 1 Introduction

Organizational memories (hereafter, OMs), have been studied as means for providing easy access and retrieval of relevant information to users. There are several technologies which support the implementation and deployment of OMs (some of them identified in [1]), however, there is relatively little support for the initial set-up of an OM. When implementing and deploying an OM, it is difficult to identify the right information to include. This task is, normally, a knowledge engineer's job, to identify relevant information and populate the OM accordingly. This process though, is time-consuming, manual and error-prone given the diversity and quantity of resources to be analyzed for relevance. Semi-automatic methods and techniques exist, but these are bound to individual technologies, as for example in [1] where the authors state that: "the knowledge engineer [then] integrates the information obtained from the thesaurus generator into the OM semi-automatically, scanning the similarity thesaurus and deciding which relations should be formalized and added to the knowledge base or ontology, which should be included in the thesaurus integrated with the ontology, and which should be ignored". On the other hand, it is always the user who has to "kick off" search in the OM. This however, requires the user to formulate a query, sometimes with the help of semi-automatic support, and then the OM system has to parse the query successfully, retrieve information deemed to be relevant according to some pre-defined notion of relevance, and present it to the user.

Another perception on OMs is in terms of knowledge delivery. There have been two, metaphorically-defined, ways of delivering knowledge reported in the literature: 'pull' and 'push' knowl-

edge [35]. The former refers to technologies which aim at pulling knowledge from vast repositories of data to people. Examples include the familiar search engines which, in some implementations, are facilitated by intelligent agents augmented with ontologies for semantically-enriched search (see, for example, the OntoSeek [21] and FindUR [30] systems). In these systems the user is expected to initiate the search by posing queries. On the other hand, 'push' systems aim at providing knowledge to their users without prior interaction. Means to achieve this ambitious goal in knowledge management (hereafter, KM) is the focus of semantically-described content, the identification of the user's task and task context.

In OM applications, both ways have been studied, though the 'pull' technologies seem to be dominant. The reason for the low uptake of 'push' technologies in knowledge delivery is probably the increased risk of 'bombarding' the user with irrelevant information which in turn could result in dissatisfaction with and discrediting the OM. To tackle this problem, OMs that used 'push' technologies made certain assumptions. For example, the KnowMore OM [2] assumes that an existing workflow engine will be in place; this in turn will be accessed and linked to the OM making it possible to reveal context-specific information regarding the user's task. Having such information available before initiating search, could (semi-)automate the task of filling-in queries with context-specific information. That way, knowledge deemed relevant to the process is proactively presented to its user.

Although we found this marriage of workflow processes and OMs an interesting one, we are skeptical about two, often unforeseen, obstacles in deploying such a system: (a) there might be situations where processes will not be easy to identify or codify in a workflow engine and (b) even when these are available and the OM is built around existing processes, it might not be desirable to restrict a user's search on those resources that are deemed to be relevant to the process the user is involved in. In addition, the technological challenges OM developers face when implementing this merger of workflow processes and OMs could be considerable [3].

To alleviate this situation, we are exploring the use of one of the core technologies for supporting OMs, that of ontologies. In particular, to cope with the problem of initially setting up an OM, we apply a method used in the Advanced Knowledge Technologies (AKT) project, Ontology Network Analysis (hereafter, ONA). We apply an algorithm to identify objects that are more important than others in the underlying ontology. We measure importance in terms of popularity. Those that have been identified are used as the initial seed to populate the OM, thus setting-up an OM containing some information readily available for use. Since our method is based on an ontology, we take advantage of the underlying ontological structures to draw inferences on the objects selected and reason about the rel-

<sup>1</sup> Advanced Knowledge Technologies (AKT), Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK, email: {y.kalfoglou,kmo,ha,nrs}@ecs.soton.ac.uk

evance of retrieved information. We applied this method to tackle an OM problem: how to identify communities of practice (hereafter, CoP).

This automation in initially setting up an OM does not eliminate the user from the picture. We are keen to explore the synergy between user-defined input and automatically-delivered content. To achieve this, we worked on ways to customize the ONA, allowing the user to customize the output of the automated content-delivery mechanism. We explored these issues in the context of our testbed application, CoP.

We give an overview of the work related to initiating OMs by emphasizing reported trade-offs between user-defined queries and (semi-)automatic query definition in section 2. We then continue with an objective analysis of the resources selection problem when setting-up an OM (section 3) which motivates our hypothesis in section 4. We test our hypothesis in section 5 with a comprehensive case study applying ONA to harvest information about a valuable OM resource: CoPs. We generalize the approach in section 6 and we discuss further implications of this approach to supporting OMs in 7 where we also point to future work.

## 2 Related work

In the KnowMore OM [2], means for semi-automatically constructing the underlying ontologies were investigated. The authors describe an interactive thesaurus-based methodology for ontology construction which is realized in a designated editor. Their focus is on extracting (semi-)automatically an ontology from domain-specific texts. In addition, the characterization of knowledge items to be used in the OM is supported by automatic tools which attach meta-data to the text. This will be used in later phases of an OM's lifecycle for guiding the retrieval and storage of related information. In our ONA-based approach we are not focusing on how to construct the underlying ontologies. As we will describe in section 4, we assume these have been constructed beforehand. Our focus is on how to provide as much information as possible to the OM user for initial set-up. However, there is an overlap of interests and methods with the (semi-)automatic ontology construction work done in AKT reported in [44].

The work described in [27] is the closest to the ONA approach. The authors describe the information retrieval process as a "select" operation on database query languages with appropriate search conditions formulated with respect to "(i) meta-data given in the information ontology (which information resources to consider or how old information to retrieve), (ii) specific-context information (employ sophisticated similarity measures for comparison of actual query situation and context factors of knowledge sources described in the OM), and (iii) the content searched for.". However, their retrieval techniques are based on annotations and their similarity measure algorithms explore only one dimension of the underlying ontology network: the subsumption dependence between nodes, i.e., class-subclass relationships. To allow users to customize their search, they provide application-specific heuristic search based on the notion of 'heuristic expression'. The users can formulate their own heuristic search formulae based on a standard template formula which takes as input a set of nodes of the underlying directed graph and for each node follows the links specified in the formula in a left to right order, delivering at each step an intermediary set of nodes as a new starting point for the next step. Although this option allows users to customize their search, the actual retrieval is based on the same subsumption mechanism. On the other hand, as we describe in section 5, ONA allows a multidimensional traversal of nodes in the ontology

network with thresholds, traversal paths, and starting nodes being user-defined, if desired.

In [6], Athloff and colleagues propose a method for OM Improvement (OMI). They argue for a method which supports user feedback as a way of improving OM over time. In their comprehensive analysis of factors that determine the usefulness of an OM they identified the selection of knowledge to be included in the OM as an important one:

"[conceptual knowledge] determines what and how experience stored in the OM plays a major role regarding the usefulness of a system."

They continue by arguing that "users often do not bother with too many questions, a problem which usually arises during the initial set-up of the OM". The conceptual knowledge Athloff and colleagues are referring to is the underlying ontology in our ONA-based approach. To tackle the problem of initial set-up, we use ONA to populate the OM automatically with the most important objects as identified from their popularity in the underlying ontology. As in [6], we also intend to use a characterization of the object to be displayed in the OM along with its popularity value as obtained from the ONA. This textual information is much appreciated by OM users [6], as it gives them explanations of the selected information. Since we base our method on an ontology, we could easily obtain these characterizations from standard 'documentation slots' which exist in most ontology development environments.

Cohen and colleagues [12], were among the first to investigate the use of metrics for ontologies. In the context of the HPKB US project [14], ontology metrics were defined to measure the level of reuse of ontological concepts in applications. For example, whenever a new axiom was added in the application's knowledge base, the metric calculated the ratio of reuse of existing ontological concepts in the newly added axiom. For ONA we use a spreading activation algorithm to all ontology constructs and do not define specific metrics.

## 3 The problem of resources selection

Despite the research reported above, a major problem when initially setting-up an OM remains unsolved: how to select the right resources to include in an OM? This problem has been identified in field surveys [17] as well as in implemented systems (e.g.: [2], [6]). This is a multi-faceted problem because it is not only concerned with the elicitation of resources that will be presented to the user or used for retrieving relevant information. These resources are also often:

- used by other systems within the organization, which incidentally also serve users in their quest for valuable information;
- 'unspecified', in that they are vaguely expressed, need to be composed by a number of related resources or are external to the organization;
- and once these resources are identified and put into use they act as a qualitative measure for the OM.

That is, if an OM's users are not satisfied with the quality of information presented to them, it is unlikely that they will return, especially when there are other conventional information-seeking systems in the organization that users used to use before confronted with an OM.

A way of tackling this resource-selection problem is by identifying the purpose of the OM: what are the users' needs and what will the OM be used for. This has been reported as one of the first phases in building an OM [17]. The techniques and methods for achieving this

rather ambitious goal are mostly taken from requirements analysis and elicitation research. They stem from Computer Supported Collaborative Work (hereafter, CSCW) research, from systems design research, and from the cognitive science literature.

However, we should be cautious when we are calling upon requirements engineering to elicit the needs when building an OM. As Zave and Jackson report in their survey [47], vague and imprecise requirements are always difficult to formalize and subsequently convert to specifications, in the early phases of software development<sup>2</sup>. This refinement is necessary, the authors continue, “to bridge the gap between requirements and specifications”, thus emerging with a specification that could satisfy users’ needs and meet the requirements. In the case of OMs, we should expect these requirements to be incomplete and vague. In addition, as Dieng and colleagues report in [17], building OMs presumes that we will re-use methods, approaches and techniques we have applied in the past in other domains:

“(1) corporate memories are not entirely new systems; they are adaptations, evolutions or integrations of existing systems; (2) before conceiving memories, the proponents or users of the solutions have taken part in the design of other types of systems (knowledge-based systems, CSCW systems, etc.), and they have transferred the solutions they already know. Most of the solutions can thus be considered as adaptations of existing solutions.”

The vagueness and incompleteness of requirements from prospective OM users led some designers to decide to build their OM around an existing workflow process engine, as for example in the Know-More OM. We discuss the adaptability of this approach and its advantages of achieving a ‘near perfect’ integration with existing IT organizational infrastructure and satisfying users’ (pre-defined) needs further in section 7, but for now we would like to focus on the importance of having a ‘comprehensive’ OM from its initial set-up. By comprehensive we mean an OM that includes a lot of resources that have been automatically extracted rather than waiting the user to initiate the extraction process. The side-effect of having this sort of OM in place is that we can tackle the ‘cold start’ syndrome identified in [19] in which the authors reported that they had relatively few knowledge assets in their OM during the first operational month which led to low access rates from its users as they couldn’t see the value-added of the OM. The problem was eventually solved, but at a cost: more systems and methods had to be used to chase users for contributions in order to enrich the content of the OM, thus leading to an increase in the OM’s knowledge assets and consequently in increased access figures.

In the following section we elaborate how our method sets up a comprehensive OM in an automated fashion.

## 4 Seeding the OM

The basis of our solution is ontologies. These consensual representations of the important concepts in some domain of interest have been studied, developed and deployed for over a decade now in various fields and applications in academia and industry. Their use in OMs has been advocated in field surveys [1] and in applied OMs (see, for example the KnowMore OM [2], the EULE2 system [40], or the integration of ontologies and Experience Factories, a form of OM, for improving maintenance [23]). Our hypothesis is that since we already use ontologies in OMs for the purposes of semantic interoperability

<sup>2</sup> In our case, the early phase of developing an OM.

and reuse, we could also use them in other ways. We could analyse their structure by taking into account relationships between their constructs, based on a tunable spread activation algorithm, yielding the nodes that are most “popular”. These are assumed, in the absence of contradicting evidence, to be the most important ones. The spreading activation algorithm also identifies nodes similar to a specific node. This is the premise underlying our hypothesis.

It could be argued that our analysis is not a qualitative one, but merely a quantitative one. However, as Cooper argues in [16], quality can be measured in two ways, in terms of popularity or importance. Our analysis yields concepts that are the most popular in the network, and since the network is about an ontology which by default represents important concepts, then these concepts are also important.

To operationalize our hypothesis, we assume that (a) ontologies will be available in the organization in which we want to deploy an OM, and (b) these will be populated. It is clear that these assumptions are strong and indeed are ongoing research issues in the knowledge engineering community, especially the latter. However, we should accept and anticipate that ontologies are popular in organizational settings nowadays, in the form of database systems, other knowledge sharing formalisms more common to the AI research community (e.g.: KIF) or indeed in emerging semantic web standard formats (e.g.: RDF(S)). As an open research issue, we are already in AKT investigating ways of (semi-)automatically constructing ontologies.

Using ontologies as the foundation for an OM is not a unique idea, but the use of ONA to provide initial information for populating the OM is novel. We should also mention that using an ontology at the start of an OM’s lifecycle allows us to provide support to users in formulating their queries from an early stage. Normally, users have to formulate initial queries unaided since there is no prior information available, as no retrievals have been made yet. In applying ONA, we support users in formulating queries by providing them with ontological information regarding the starting node for initiating an ONA-based search. This information is readily available in existing slots in the underlying ontology (such as the documentation slot).

## 5 ONA

In this section, we set out the principles underlying ONA, and then demonstrate an application of the method — gathering information on CoPs. In section 5.2, we then set out the opportunities and problems that characterize the study of CoPs. Finally, in section 5.3, we set out an application of ONA to the problem of kick-starting an OM for a particular CoP.

### 5.1 Principles of Ontology Network Analysis

ONA [5] is the technique of applying information network analysis methods to a populated ontology to uncover certain trends and object characteristics, such as shortest paths, object clusters, semantic similarity, object importance or popularity, etc. A variety of such methods have been explored in the past for different information retrieval purposes. ONA investigates the application of these methods to analyse the network of instances and relationships in a knowledge base, guided by the domain ontology. There are many methods of studying networks, and of course many types of networks that can be studied (cf. [33]). However, the advantage of studying ontologies is that the relations therein have semantics or types, and therefore that the semantics provide another source of information over and above connectivity or simple subsumption. This semantic information can be taken account of when performing a network analysis, allowing

“raw” results to be refined on a relatively principled basis. An ONA example application is described in section 5.3 and an example algorithm is detailed in [5].

ONA methods can be harnessed to address the resources selection problem in building OMs (section 3), by using populated ontologies already in place in organizations to select a set of important and interesting resources to feature in a new OM. The fact that the method is automatic takes some of the burden of OM development from its users or managers, and allows some quality content to be put in place prior to use, thereby increasing the likelihood of early take-up by its users.

Being automatic, ONA is not, of course, foolproof or infallible. Many points of interest in an organization’s ontology will not be spotted by the methods involved, especially if the ontology is in some way incomplete, and fails to cover the object domain fully in some important respect. Clearly, ONA cannot be the *only* principle used to populate an OM. However, by extracting some information from an ontology, ONA can be used to suggest an initial set of interesting concepts and relations. Certain assumptions must be made to support the use of ONA here, but as the OM develops, such assumptions can be relaxed, as the population of the OM begins to happen by its users. And user feedback as to the actual importance of the entities uncovered will always be essential.

The ONA technique of interest to this paper is the application of network measures to an ontology to determine popular entities in the domain. Such entities can be either classes or instances, where *popularity* is (a) defined in terms of the number of instances particular classes have (class popularity), and the number and type of relation paths between an entity and other entities (instance popularity), and (b) regarded as a proxy for importance. Clearly this latter claim is one that will not always be true. However, the working assumption is that the important objects will have a stronger presence in a representation of the domain, and will have a lot of key relationships with many other entities (they will act as “hubs” in the domain)<sup>3</sup>.

Given a first pass ONA of an ontology, giving the most popular entities, an OM developer can exploit user feedback to hone the analysis. Two particular ways of doing this can be envisaged.

1. Important instances can be selected — these instances may have been counted as ‘popular’ under the first pass analysis or not, as the case may be, and hence could be manually selected as important instances independently of the governing assumption that popularity = importance — and the ONA performed once more, this time measuring not the quantity of relations between all entities, but measuring the quantity of relations between the selected instances and other entities.
2. Relations can be weighted according to their importance, and the weights transferred from entity to entity along the relation-connection. Hence one relation (e.g. *co-author-with*) might be weighted more highly than another more common one (e.g. *shares-office-with*), whose relevance to the domain in question is not as high. In that case, the effect when performing an ONA is to privilege the entities that enter into the highly-weighted relations as against those that do not. There are two (classes of) ways of

<sup>3</sup> One doubtless common circumstance where this assumption will *not* be reliable would be where an ontology is pieced together from legacy datasets. In such a case, the most popular entities are likely to be those represented in detail elsewhere for other purposes, whose importance may not carry over into the current application. Another point to note is that quantitative information may be more prevalent than qualitative information, and that therefore entities that enter into many quantitative relations could be overvalued. We emphasise once more: user feedback is essential.

differentially weighting relations.

- (a) First, relations could be differentially weighted automatically, on similar lines to the selection of important entities, viz., the relations most often filled with values in the knowledge base will be weighted higher than others.
- (b) Alternatively, the weights can be fixed manually. This has the advantage of being sensitive to user understanding of the domain, and the disadvantage of being a complex and difficult process that could be time-consuming, especially if there are a lot of relations about. Of course, as with entity-selection, an initial cut using automatically-created weights could be run past a user, who might suggest adjustments; this might be the cheapest method of getting the best of both worlds.

In the next subsection, we discuss communities of practice, and then we go on to examine the use of a particular spreading activation algorithm to perform an ONA in order to extract information about communities that is latent in a domain ontology.

## 5.2 Communities of Practice

CoPs’ value in the construction and maintenance of OMs has been acknowledged by other OMs developers. To quote [4]:

“[...] employees solve knowledge-intensive tasks (KITs) cooperatively as a community of practice, embedded into the overall business workflows and supported and monitored by the OM system. Applications used and repositories filled and queried are the closely related basis of the OM environment, and value-added care about intelligent support for knowledge indexing, distribution, storage, search, retrieval, and integration.”

A CoP is an informal group of individuals with a common interest in a particular work practice. Their interest should take a particular form: the individuals concerned should wish to *improve* their practice, either for financial reasons (picking up bonuses, or securing promotion), or mere professional pride. The CoP then plays a number of roles. First, the individuals in it will meet informally to discuss particular problems and issues facing the practice; in this way the CoP fosters a common appreciation and characterisation of the practice. Second, particular solutions will be demonstrated and evaluated within the CoP; the CoP therefore fosters innovation, partly through the shared understanding of problems, and partly through the evaluation “process,” which is likely to be rigorous and competitive. Third, the informal nature of the contacts mean that, almost automatically, innovations will be built on by interested others who “tinker with” or improve them; informality means that restrictive practices such as patenting or licensing tend not to be invoked within the CoP, and therefore that innovation very naturally becomes a collaborative process. Fourth, new exponents of the practice can use the CoP as an important tool for situated learning of the practice. After training, most effective learning takes place “on the job,” as new practitioners discuss their problems with their fellows, or learn from their colleagues how to integrate the practice with the rest of their business workflow; in such a way, the CoP becomes a repository and dissemination mechanism combined for best practice [45].

A CoP contrasts with other, more formal structures that centre round a practice [15].

- *Functional groups* specialize in particular functions within an organization, for example, marketing, administration, security or finance. The agents form a homogeneous set, drawn together by

disciplinary specialization, and are organized in hierarchies; the purpose of the group is not to produce learning, though of course new recruits achieve situated learning. The hierarchical structure, and often a shared educational background, keeps the group together.

- *Teams* are also well-defined within organizations. They are made up of individuals brought together to carry out a given task, each chosen because of some specialist skill that is assumed to be required for the task's performance. Hence the members of a team are highly heterogeneous, and the team's management will be intended to integrate their functional knowledge. Learning, if it takes place, is unintended, and tends to be via the interactions across functional specialities — a specialist might come to understand the constraints on, and the requirements and responsibilities of, his colleagues. The team's life is normally not extended beyond the achievement of the task's goals.
- A *network* consists of individuals across organizations who have interests in working together, for example, in a rough system of producer interests, some of whom provide components, parts or expertise for a final manufacturer; the function of the network is to bring together suppliers and consumers of particular goods or services to facilitate negotiations, or to cut purchase overheads, e.g. information-gathering costs. Such a network is made up of heterogeneous agents, and focuses on the exchange of knowledge, perhaps encoded in price signals. The requirement for complementary knowledge keeps the community going, and a necessary condition of this is a high level of mutual trust.
- *Epistemic communities* are relatively formal groups of agents who produce knowledge, or codes for expressing knowledge, from some position of authority that may be formal (e.g. a professional association), or more informal (e.g. based on particular agents' positions of eminence). Different interests tend to insist on representation in such forums, and hence the makeup of such a community can be quite heterogeneous. Such communities often play a wider political role, and can be the "public face" of a discipline. Recruitment to such groups is founded on peer approval.

In contrast to these types of group, CoP's members — it has an informal, self-selecting, largely homogeneous membership — are interested in increasing their skills, and in accumulating and circulating best practice. As a result, a CoP is an excellent vehicle for situated learning of the practice [45].

When we consider which types of group are of interest for OMs, then the comparison is very instructive. Organizational learning has a dual aspect [8]. "Single-loop learning" is an organizational learning process whereby knowledge is obtained to solve problems based on an existing and well-understood model of the domain, in other words a routine process. "Double-loop learning" involves the establishment of a new set of paradigms, models, premises, representations or strategies to supersede the existing models, to improve the organization's response to existing problems, and to enable the organization to address new problems. These types of learning are called "Learning I" and "Learning II" by Bateson [9].

As Nonaka and Takeuchi point out ([31], p.45), one problem with the adoption of this approach to learning — useful as it has been in a number of respects — is that it sees organizational learning as a process of adaptation to external stimuli that involves the development and modification of existing routines supported by OM, not as a process where knowledge is created. Even when such a view is taken, it can be difficult for insiders to spot the right moment for attempting serious knowledge creation, except by making such a process routine

— in which case of course there is no guarantee that there will be no period when either (a) knowledge acquirable only by double-loop learning is required but not available, or (b) an expensive double-loop learning process is initiated for which there is no immediate requirement.

Part of the trouble is that much learning theory, as in epistemology generally, has as its focus the individual [10, 32]. The problem here is that when this focus is transferred to actual cases of organizational learning, the complexity of the collective learning process, which cannot straightforwardly be reduced to a simple addition of learning processes for the individuals in the organization, cannot be properly respected. The key to implementing effective organizational learning processes is to understand the organization in terms of the collectives that make it up, the overlapping groups that were listed above; learning across these organizations, then, is a complex process of interaction between these heterogeneous entities [10, 45, 15].

One important role for OM, therefore, is to act as the information storage buffer between these overlapping groups. In that event, a key factor from the point of view of creating or seeding an OM is the availability of various resources. *In general*, the more formal a group, the more likely it is that relatively tractable sources are available for populating an OM. There are two reasons for this: first, formal functions lend themselves to careful management that can track events and leave a highly visible audit trail, and second, their very formality places those events on the management radar. In contrast, informal groups, by their nature, are often undetected by management, and their "memory" may well boil down to the sum of the non-metaphorical psychological memories of their members, with all the potential problems that this implies.

In particular, a functional group, say, or a team, is barely likely to have a life outside of their working existences. For example, the former has a strict hierarchical structure, which regulates the permitted interactions between members — a division of labour intended to increase efficiency — to a series of delegations, as the task is understood at increasingly lower levels of abstraction as we move down the hierarchy. Each level of the hierarchy might well, by contrast, form a CoP, and may have links not only across equivalent nodes in the functional group hierarchy, but also with equivalent levels in hierarchies of orthogonal functional groups within the organization, or with similar levels in related functional groups in other organizations. Here the CoP, parasitic on the functional group, is formed by people wishing to understand the process of, in this case, receiving a task description at one level of abstraction, and decomposing it into subtasks which can then be delegated to available resources further down. The OM of the functional group will consist of the decompositions and delegations, together with the feedback that passes up the hierarchy; creating and maintaining such an OM is, of course, non-trivial. But the OM of the CoP is not something that will spontaneously appear, consisting as it does of informal chats and retellings of "war stories" around the photocopier or in the pub after work.

Similar considerations apply to teams and epistemic communities. Each consists of heterogeneous agents brought together to carry out a particular task, or open-ended series of tasks in the case of the epistemic community. In that event, the actual work of the team or epistemic community generally takes place in formal scheduled minuted meetings. Converting this relatively stable resource to an OM proper is, no doubt, problematic in various ways, but there is at least a fairly straightforward way to begin to populate the OM. On the other hand, the informal work done that pertains to the team goes on within related CoPs. Team members go back to their informal CoPs, transmitting new knowledge about the requirements of people who carry

out different functions, and tinkering with new ways to incorporate such exogenous requirements. The knowledge created by a team or epistemic community is analogous to the gears of an engine, whereas the knowledge of the CoP is analogous to the oil; the former is much more visible than the latter, but will eventually seize up and grind to a halt if the latter is not present.

As a result of such considerations, CoPs are seen as key elements in the efficient working of an organization, and as key agents in knowledge management [45, 18, 34]. Well-known companies that have nurtured CoPs include Hewlett-Packard Consulting, Arthur Andersen, Accenture, Ernst and Young, BP, Caltex, Chevron, Conoco, Marathon, Mobil, PDVSA, Shell, Statoil, TOTALFINAELF, Intel, Lucent, Siemens, Xerox, IBM, the World Bank and British Telecom [41, 26]. Smith and Farquhar give a detailed example of the use of CoPs in the oil industry consultants Schlumberger [41]. Schlumberger supports the development and maintenance of an OM for its oil engineering CoP by providing what is called a knowledge hub, consisting of a series of technologies designed to support worldwide connectivity between those engineers, and to foster a culture that encourages its use; such technologies are relatively straightforward — email, the web, bulletin boards, together with data management systems, project archives, expertise directories and so on. Maintenance of the different parts of the knowledge hub is detailed specifically to *knowledge champions*, people responsible for animating the community, encouraging participation, reporting successes etc. ([41], pp.22–27). Smith and Farquhar are clear about the importance of populating such resources.

“Just because an intranet portal has been built filled with world-class technology, it is not a given that community members will flock to it. Do not overwhelm them with all the features that computer scientists can think of that “clearly” would be beneficial. Instead, be cautious. Determine first what technology the community members actually use. . . .

An up-front investment is required to seed the initial knowledge repository. It is difficult, if not impossible, to convince community members to contribute to an empty shell. . . . Not only must there be content from the launch date, but it must be quality content as well.” ([41], p.28)

This vision of the creation of a CoP memory beginning with a seeding process is shared by Marshall and colleagues [28], where their concept of a *community memory*, the open-ended set of knowledge and shared understandings that acts as the CoP’s intellectual glue, maps pretty well onto the CoP OMs that we have been discussing. The daily activities of the CoP members are seen refracted through this community memory. The problem, as they see it, is that as the community develops, the memory grows so that the maintenance task becomes overwhelming; simultaneously, however, the memory is growing stale, with inconsistencies, redundancies and irrelevancies proliferating as the focus of the CoP changes, and as the CoP needs to maintain contact with exogenous sources of knowledge, such as the web or other large-scale information resources. In that case, there will have to be a process of purging, together with a restructuring of a trimmed down OM.

However, such seeding restructuring processes, as advocated by [41, 28], are rendered much more complex by the informal nature of the CoP itself. Too firm a smack of management will destroy the informal nature of the CoP — and therefore make it much more difficult for the CoP to support the invisible, informal parts of the work process [45]. CoP management is a delicate process, and various methods have been suggested for doing it [46, 29]. These methods

all begin with one of the most difficult aspects of managing informal communities — discovering the extent of the community itself.

### 5.3 ONTOCOPI

To this end, we have applied a particular instantiation of ONA to attempt to isolate CoPs within organizations described by ontologies [33]. The rough idea is to use an ontology-based spreading activation algorithm to search the knowledge base, moving from instance to instance along relationship connections as defined by the ontology. The system is called ONTOCOPI (ONTOlogy-based Community Of Practice Identifier), and is currently implemented as a Protege ([20]) plug-in as well as a standalone Web accessible program.

Spreading activation was first introduced by Quillian [38] to simulate human semantic processing in a machine subsequently it has formed the basis for many information retrieval methods such as semantic similarity measures, Web analysis algorithms, community identification, case-based reasoning, etc. ONTOCOPI’s algorithm combines and improves ideas from previous work on similarity measures, such as shortest path measures [39], multi-path traversal [36], and constrained spreading activation methods [13]. ONTOCOPI’s algorithm can make use of the ontology to make decisions about which relationships to select and how they should be valued. Ontological axioms can also be consulted in the relationship selection process.

Some caveats must be pointed out here. Relationships in ontologies are mostly of a formal nature. CoPs however, tend to have an informal nature, which is one of the major difficulties for CoP management (section 5.2). The traditional method used to identify CoPs most often<sup>4</sup> appears to be more or less structured interviewing ([46], pp.8–10) and recently Sol and Serra proposed a multiagent Web-based approach ([42]). The ONTOCOPI assumptions about CoP identification attempt to get around this time consuming activity.

A formal relationship can stand as proxy to an informal one. Hence we can infer that two people who co-author a paper are more likely to be members of the same CoP. If two CoP members actually share no formal relationships (at least, no formal relations captured by the ontology), then any vector addition of formal relations can also stand proxy for informal ones. Hence if A co-authored a paper with B, who works on a project with C, then it may be inferred that A and C, who have no formal connection, are more likely to be members of the same CoP. Total accuracy, of course, is impossible for an informal and rapidly-evolving social group like a CoP; furthermore, the aim of ONTOCOPI is only to *support* CoP identification, a very expensive operation in its own right [46]. A certain measure of indeterminacy is inevitable.

Another fact of importance is that ONTOCOPI can’t identify relationships that aren’t there: if two people in the same CoP simply have no formal relationship recorded in the ontology, and no chain of formal relations linking them, then their co-membership cannot be found. The information has to be in the ontology for ONA to tease it out. Finally, ONTOCOPI can’t distinguish *between* CoPs. If someone is a *broker*, i.e. a person who functions in two separate CoPs [45], then ONTOCOPI will tend to pick up the union of the two CoPs (although the settings can be modified somewhat to try to ameliorate this difficulty — see below).

It follows that ONTOCOPI cannot infallibly identify a CoP. But then a CoP is in many ways indeterminate anyway. ONTOCOPI, however, does support CoP identification, a resource-heavy task that

---

<sup>4</sup> Except in organizations *defined* around a CoP, which may include Schlumberger [41].



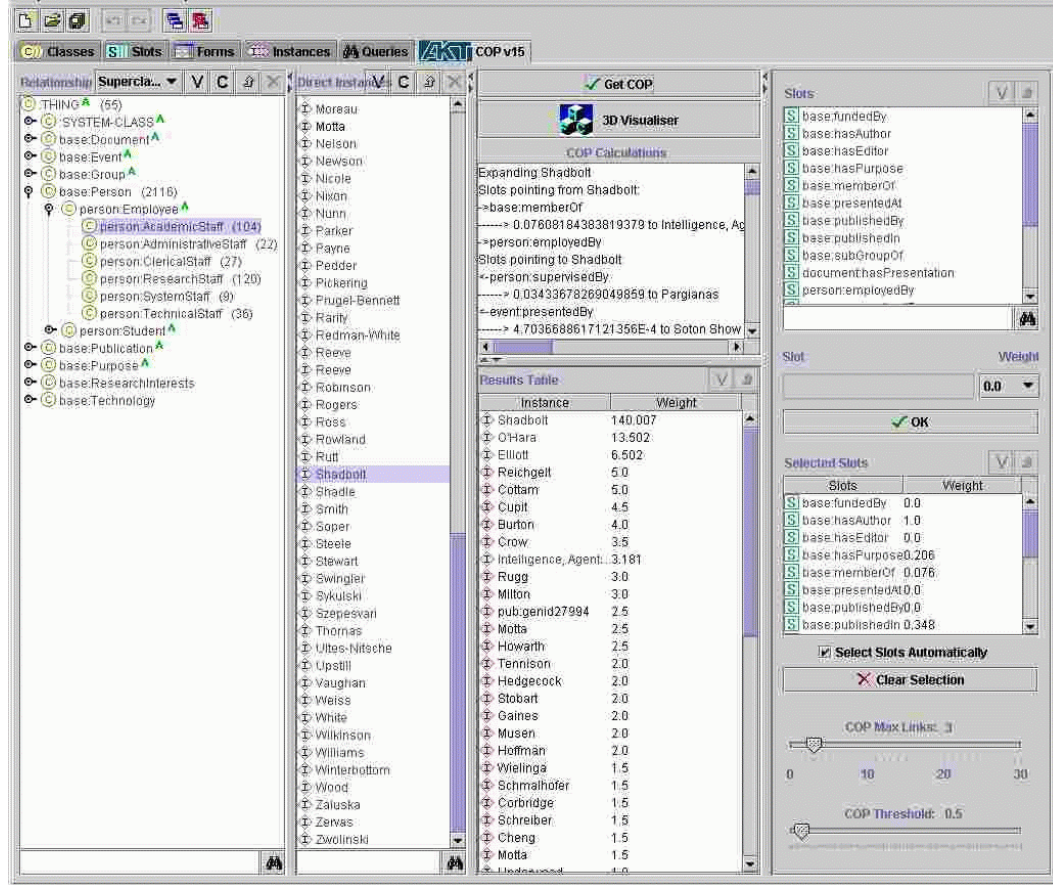


Figure 1. A screenshot of ONTOCOPI as a Protege plug-in.

may be alleviated to some extent by the not-so-subtle assumption that formal connections can approximate informal relationships.

The interface can be seen in figure 1. As a prototype, we do not claim that this is in any way optimal, but it indicates the information it can give. The panel on the far left shows the class hierarchy of the ontology. The panel next to it shows the instances of a selected class. From this panel, an instance can be selected to be the “centre” of the CoP investigation (i.e., the relations radiating out from this individual will be those used as the basis of the CoP identification). The panels on the right hand side set the relation weights and parameter values (e.g., the number of links the algorithm will spread to). Clicking the ‘Get COP’ button will set the algorithm going. The centre right top panel displays the current calculations, and centre right bottom displays the weights that have been transferred to other instances, in descending order of weight (i.e. a rough specification of the CoP, the main output of ONTOCOPI). In this diagram, the CoP of *Shadbolt* has been investigated, and ONTOCOPI has suggested, in descending order of preference, *O’Hara*, *Elliott*, *Reichgelt*, *Cottam*, *Cupit*, *Burton* and *Crow*, then the *Intelligence, Agents, Multimedia Group* of which *Shadbolt* is a member, then *Rugg* and so on.

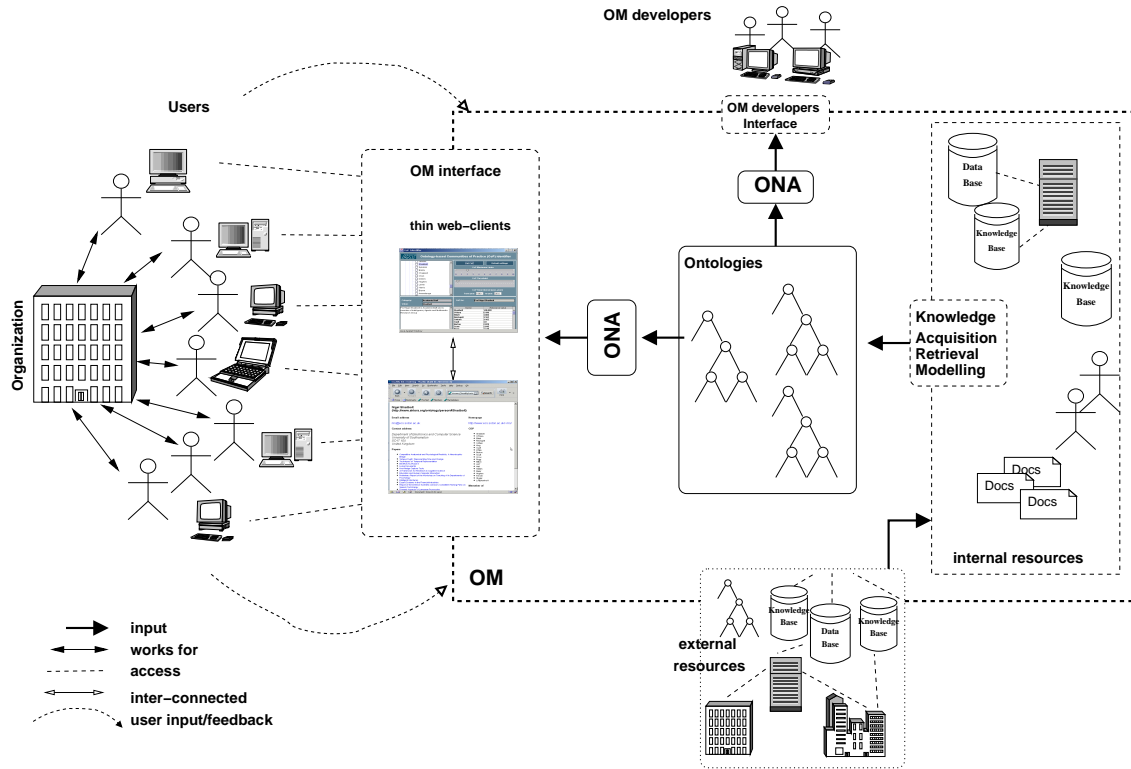
Order is important, so are the relative weights. *O’Hara* scores 13.5; this is meaningless except in the context of a particular search. Here, 13.5 is very good, twice the score of the next candidate. On the other hand, the user may be more suspicious of the ordering of, say, *Tennison*, who scores 2.0, and *Motta*, who scores 1.5. The figures themselves have no constant interpretation (except in terms of the al-

gorithm); it is for the users to take the suggestions and interpret them according to their own understanding of the structure of their CoP. Hence ONTOCOPI, to reiterate, only supports CoP identification.

The relation weights can be created automatically based on frequency, or created artificially. In this run, the weights were calculated automatically, with the most frequently used relation getting weight 1, those not used at all getting 0, and the others being allocated accordingly. This, then, might be a first run; a second run might adjust the weights manually, perhaps giving some less used but important relations higher weights.

The algorithm initializes instance weights to 1, and then applies a breadth-first spreading activation search, going through all the relations, and using the relation weight and the instance weight of the departure node, transfers more weight to the arrival node. It then continues the search, this time out from the arrival node. Instances then accumulate weight according to the numbers of relations (or chains of relations) they have with the initial instance chosen to start the process; the longer the chain, the smaller the weight transferred; the weightier the relation, the larger the weight transferred. Hence a short distance, or a significant connection, with the base instance will tend to push an instance up the batting order. In the example, *O’Hara* has written a lot of papers with *Shadbolt* — many individual relations of a highly significant kind in this context (indeed this paper by its very existence has already increased *O’Hara*’s score, as well as those of *Alani* and *Kalfoglou*). *Shadbolt* has few direct connections with *Gaines*, but their transitive links are many and varied, and hence





**Figure 2.** Applying ONA at different phases of OMs: to push knowledge to users as well as help developers tune their OMs.

Gaines appears on the radar.

The “raw” algorithm can be refined according to user feedback — recall that user feedback is essential with ONA. Manual setting of relation weights has already been mentioned. Other ways to control variables include:

- *Temporal considerations*, if they are modelled in the ontology, can be factored in. For example, the relations might only be considered if they were extant, say, in the last 5 years. [5] shows how, on this interpretation, *Shadbolt’s* CoP has altered over the last fifteen years, beginning in the mid 80s with a number of psychologists, who gradually fall out of the picture as we move towards the present, when AI and later knowledge engineering and KM concerns take over as *Shadbolt’s* academic career evolved; new people become colleagues, or become connected to *Shadbolt* by other more or less circuitous routes.
- *Filtering out “hubs”*. One problem, already implicitly mentioned, is that of “hubs”. A hub, in this context, is a highly-connected person with lots of relations with other people through work, publishing, or whatever. Such people carry a lot of relative weight — in more ways than one — and so can sometimes skew the CoP by transferring an inordinate amount of weight to the instances with which they are connected. The ONTOCOPI algorithm can constrain the weight transfer based on the level of connectivity

of such people. This allows the comparison of CoPs to see what contribution certain people made to them.

- *Privileging of classes*. Particular classes can be selected to identify the concepts of interest, and then the system will automatically select the relationships that interconnect these classes, and assigns relationship weights on the basis of their frequency.
- *Differential initial weighting of instances*. This is not implemented yet, but one could imagine altering the initial weights, either manually (selecting definite CoP members and ruling out definite non-members, and increasing the value/devaluing all their relationships accordingly), or automatically (e.g., increasing the weights of papers which contained certain key words in their titles or abstracts).

One could imagine many more adjustments to refine the basic picture. The appropriate refinements in a particular domain will depend on the features of the domain itself, and what is captured by the ontology.

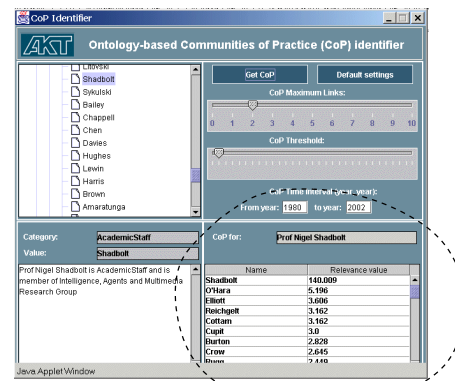
We have described one way to apply ONA to the problems of resource selection for OMs. In the next section, we move on to a generic account of the relation between ONA and OMs.

## 6 Generalising the method

In figure 2 we depict a high-level diagram of an OM. This is not meant to be a reference architecture for OMs, such as the one depicted in [25]. This figure emphasizes the dual role of ONA and the supportive role ontologies play in our scenario. On the left-hand side of the figure we have users of an organization performing their regular tasks. In the centre we have an OM which is composed, at this abstract level, by two interfaces to users and OM developers, a port to external resources, and internal resources existing in the organization's repositories. The latter could have several forms, ranging from tacit knowledge possessed by experts to explicit knowledge expressed formally in KBs or databases. In the centre of our abstract OM, lie the ontologies which underpin the entire OM. These are either existing resources or are constructed (semi-)automatically with the aid of knowledge acquisition, retrieval and modelling techniques. We do not refer to these in this paper as our focus is on the use of ONA: the two rectangular boxes denoting "ONA" are placed between the ontologies and OM interfaces to users and developers. The genericity of ONA makes it possible to use it for pushing knowledge to users but also as an aid for the OM's developers. They could apply ONA to the organization's ontologies in order to identify which concepts should be presented to certain types of users. For instance, assuming that there is a workflow engine in the organization, and developers are looking for ways of linking the OM to it, they could either engage in modelling techniques such as those used in linking the KnowMore OM with workflow processes [2], or they could use ONA to help them identify which concepts from the underlying ontologies are mapped onto the ones of the workflow's processes. This activity requires inspection and familiarization only with one end of the prospective link: that of the workflow processes. The developer then, uses the concepts found in the workflow processes as a starting node for his/her ONA. This could reveal whether further linking is feasible (or otherwise), thus saving development time and allowing developers to deal with ontologies that they are not familiar with. The approach taken by the KnowMore OM, requires a careful analysis and possibly, modelling of workflow processes and ontologies before a link between them could be implemented. ONA can ease the analysis on the ontology end of this prospective link.

We also include two curly dotted arcs in figure 2 linking users with the OM. These denote users' feedback and input. This is an important, probably the most important, element of any OM architecture. As Althoff and colleagues have shown in [7], an OM can be improved over time by user feedback and input. In our abstract architecture, we envisage light-weight feedback mechanisms, implemented as thin Web-clients, accessible through Web browsers, as a means for eliciting feedback on an OM's resources. An example of such technology from the AKT project is the Digital Document Discourse Environment [43] used as a digital discussion space.

Finally, the OM interface to its users is light-weight and accessible from distributed clients on the Web. We have developed several such interfaces for accessing our dedicated tools in AKT. An example, taken from the CoP application (section 5.3) is illustrated in figure 3. Two kinds of interfaces included here: a dedicated OM interface, where the user can state preferences in selecting the appropriate node to search for related information, or there could be a customized rendering of information into a user's Web browser. The latter is extracted automatically after applying ONA to the underlying ontology, whereas the former requires user input to tune the search criteria.



dedicated OM interface

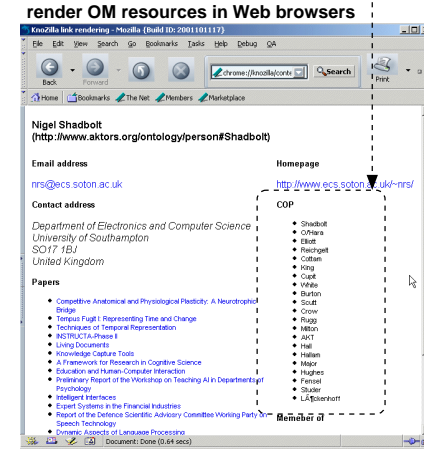


Figure 3. Different ways of accessing OM's resources: through dedicated Web-run interfaces or via standard Web browsers.

## 7 Discussion and further work

In this section we elaborate on some implications and potential caveats of our ONA. We categorize them in three broadly defined areas: information overload, context-awareness and domain-independence. We critically review the application of ONA when these areas are considered in deploying OMs:

- Information overload:** As Abecker and colleagues pointed out in their KnowMore OM, the progressive and query-based interaction with the OM from initial set-up acts as "a safeguard against unwanted information overload." [2]. Potential drawbacks include: progressive interaction means that the initial set-up will suffer from 'cold-start' syndrome, not enough information will be available; query-based interaction requires expertise and domain familiarization from the users to get the most out of an OM. The advantages are discussed below under the heading 'context-awareness'. There isn't a golden rule to follow when we, as developers, face this dilemma. It is worth pointing out though that users, amid the bulk of information ONA pushes to them, are still in control of it. They can change the search criteria (namely, the starting node in the ONA algorithm), to meet their preferences. Users can also choose which relations to traverse and their relative importance (weights). Further, we support this change as much as possible by ontologically-guiding the user in choosing the right starting node,

as nodes always carry some sort of semantic information drawn automatically from the underlying ontology. So, it could be argued, this task becomes a pedagogical experience for users apart from easing their query formulation.

- **Context-awareness:** this has been recognized as the Achilles' heel for OMs. One proposed remedy, advocated by proponents of marrying workflow processes and OMs (see, for example [3]), seems to work well in settings where workflow processes are either existing, or are relatively easy to identify and model. ONA takes a different approach in tackling context-awareness. We do not assume that workflow processes will exist, but we merely rely on ontological resources which we assume exist or could be constructed. Contextual relevance can be achieved in a number of ways thanks to the genericity of ONA. We could rely on ad-hoc technologies, such as profiling users' interests by using agents [37] or by embedding personalization facilities in thin Web clients [24], or rely on identification of users' tasks [11]. In addition, our reliance on organizational ontologies gives us the ability to exploit knowledge about users identity (obtained from system-entry logs), and thus help guess their information needs.
- **Domain-independence:** this is a desired feature for OMs. ONA is not specific to any kind of ontology, or indeed to any ontology at all! This makes it possible to apply ONA to more than one ontology as are likely to exist in large organizations. As we described in the previous section, we could use ONA as a tool to assist knowledge engineers in deciding which ontologies to consider for supporting the OM. This in turn, speeds-up the task of selecting appropriate organizational ontologies. However, ONA will not be the only tool to be used in this process: in the case of similar or conflicting ontologies there might be a need to integrate them or to resolve inconsistencies. In this case, ONA is only one of the many tools that knowledge engineers would like to have at their disposal to tackle these challenges.

A number of components described in this paper are not fully implemented yet. As this is ongoing work, we are in the process of integrating several tools developed in the context of the AKT project to realize the generic architecture described in section 6. We have already designed, developed and deployed the CoP exemplar application in various settings and are currently in the process of evaluating it. We have also developed much of the infrastructure needed to deploy such an OM: Web clients [24] and ontologies are ready for use. We are currently working on methods for maintaining these ontologies, constructing and populating them as automatically as possible [44]. Several application scenarios are currently under consideration one of which would use OMs to access heterogeneous resources and push information to dedicated members of a community. In these scenarios we plan to use the knowledge-sharing infrastructure developed in AKT [22].

## ACKNOWLEDGEMENTS

This work is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the EPSRC or any other member of the AKT IRC.

## REFERENCES

- [1] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, and M. Sintek, 'Toward a Technology for Organizational Memories', *IEEE Intelligent Systems*, **13**(3), 40–48, (June 1998).
- [2] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, and M. Sintek, 'Context-Aware, Proactive Delivery of Task-Specific Knowledge: The KnowMore Project', *International Journal on Information Systems Frontiers (ISF), Special Issue on Knowledge Management and Organizational Memory*, **2**(3/4), 139–162, (2000).
- [3] A. Abecker, A. Bernardi, S. Ntioudis, G. Mentzas, R. Herterich, C. Houy, S. Muller, and M. Legal, 'The DECOR toolbox for workflow-embedded organizational memory access', in *Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS'01), Setubal, Portugal*, (July 2001).
- [4] A. Abecker, A. Bernardi, and M. Sintek, 'Proactive knowledge delivery for enterprise knowledge management', in *Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering (SEKE'99), Kaiserslauten, Germany*, pp. 120–127, (June 1999).
- [5] H. Alani, K. O'Hara, and N. Shadbolt, 'ONTOCOPI: Methods and tools for identifying communities of practice', in *Proceedings of the 2002 IFIP World Computer Congress, Montreal, Canada*, (August 2002).
- [6] K.-D. Althoff, A. Birk, S. Hartkopf, W. Muller, M. Nick, D. Surmann, and C. Tautz, 'Managing Software Engineering Experience for Comprehensive Reuse', in *Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, SEKE'99, Kaiserslauten, Germany*, pp. 10–19, (June 1999).
- [7] K.-D. Althoff, M. Nick, and C. Tautz, 'Improving organizational memories through user feedback', in *Proceedings of the Learning Software Organizations (LSO'99) workshop, Kaiserslauten, Germany*, ed., F. Bommarius, pp. 27–44, (June 1999).
- [8] C. Argyris and D. Schon, *Organisational Learning: A Theory of Action*, Addison-Wesley, 1978.
- [9] G. Bateson, *Steps to an Ecology of Mind*, Paladin, 1973.
- [10] J.S. Brown and P. Duguid, 'Organizational Learning and Communities of Practice: Toward a Unified View of Working, Learning and Innovation', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. 99–121. Butterworth-Heinemann, (2000). first published 1991.
- [11] J. Budzik and K.J. Hammond, 'User interactions with Everyday Applications as Context for Just-In-Time Information Access', in *Proceedings of the 2000 International Conference on Intelligent User Interfaces, ACM*, (2000).
- [12] P. Cohen, V. Chaudhri, A. Pease, and R. Schrag, 'Does prior knowledge facilitate the development of knowledge-based systems?', in *Proceedings of the Sixteenth National Conference on Artificial Intelligence, AAAI'99, Orlando, FL, USA*, pp. 221–226, (July 1999).
- [13] P. Cohen and R. Kjeldsen, 'Information retrieval by constraint spreading activation in semantic networks', *Information processing and management*, **23**(4), 255–268, (1987).
- [14] P. Cohen, R. Schrag, E. Jones, A. Pease, A. Lin, B. Starr, D. Gunning, and M. Burke, 'The DARPA High Performance Knowledge Bases project', *AI Magazine*, **19**(4), 25–49, (1998).
- [15] P. Cohendet, F. Creplet, and O. Dupouet, 'Interactions Between Epistemic Communities and Communities of Practice as a Mechanism of Creation and Diffusion of Knowledge', in *Interaction and Market Structure: Essays on Heterogeneity in Economics*, eds., D. Delli Gatti, M. Gallegati, and A. Kirman. Springer-Verlag, (2000).
- [16] W.S. Cooper, 'On selecting a measure of retrieval effectiveness', in *Readings in Information Retrieval*, pp. 191–204, (1997).
- [17] R. Dieng, O. Corby, A. Giboin, and M. Ribiere, 'Methods and tools for corporate knowledge management', *International Journal of Human-Computer Studies (IJHCS)*, **51**, 567–598, (1999).
- [18] N.M. Dixon, *Common Knowledge: How Companies Thrive By Sharing What They Know*, Harvard Business School Press, 2000.
- [19] C. Gresse-vonWangenheim, D. Lichtnow, A. vonWangenheim, and E. Comunello, 'Supporting Knowledge Management in University Software R&D Groups', in *Proceedings of the 3rd International Workshop on Learning Software Organizations (LSO'01), Kaiserslautern, Germany*, eds., K.-D. Althoff, R. Feldmann, and Muller, volume 2176 of *Lecture Notes in Computer Science*, pp. 51–66. Springer-Verlag, (September 2001).
- [20] W. Grosso, H. Eriksson, R. Ferguson, J. Gennari, S. Tu, and M. Musen,

- 'Knowledge Modelling at the Millennium - The design and evolution of Protege2000', in *proceedings of the 12th Knowledge Acquisition, Modelling, and Management(KAW'99)*, Banff, Canada, (October 1999).
- [21] N. Guarino, C. Masolo, and G. Vetere, 'OntoSeek: Content-Based Access to the Web', *IEEE Intelligent Systems*, **14**(3), 70–80, (May 1999).
- [22] K. Hui and A. Preece, 'An Infrastructure for Knowledge Communication in AKT', White paper v.1.0, available from URL: <http://www.csd.abdn.ac.uk/research/akt/aktbus-1.0/index.html>, Department of Computer Science, University of Aberdeen, (July 2001).
- [23] Y. Kalfoglou, 'On the convergence of core technologies for knowledge management and organisational memories: ontologies and experience factories', in *Proceedings of the ECAI 2000 Workshop on Knowledge Management and Organisational Memories(W11-KMOM'00)*, Berlin, Germany, (August 2000).
- [24] Y. Kalfoglou, J. Domingue, E. Motta, M. Vargas-Vera, and S. Buckingham-Shum, 'MyPlanet: an ontology-driven Web-based personalised news service', in *Proceedings of the IJCAI'01 workshop on Ontologies and Information Sharing*, Seattle, USA, (August 2001).
- [25] O. Kuhn and A. Abecker, 'Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges', *Journal of Universal Computer Science*, **3**(8), 929–954, (1997).
- [26] E.L. Lesser, M.A. Fontaine, and J.A. Slusher, 'Preface', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. vii–xi. Butterworth-Heinemann, (2000).
- [27] M. Liao, A. Abecker, A. Bernardi, K. Hinkelmann, and M. Sintek, 'Ontologies for Knowledge Retrieval in Organizational Memories', in *Proceedings of the Learning Software Organizations(LSO'99) workshop*, Kaiserslautern, Germany, ed., F. Bomarius, pp. 19–26, (June 1999).
- [28] C.C. Marshall, F.M. Shipman III, and R.J. McCall, 'Making Large-Scale Information Resources Serve Communities of Practice', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. 225–247. Butterworth-Heinemann, (2000). first published 1995.
- [29] R. McDermott, 'Why Information Technology Inspired But Cannot Deliver Knowledge Management', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. 21–35. Butterworth-Heinemann, (2000). first published 1999.
- [30] L.D. McGuinness, 'Ontological Issues for Knowledge-Enhanced Search', in *Proceedings of the 1st International Conference on Formal Ontology in Information Systems(FOIS'98)*, Trento, Italy, ed., N. Guarino, pp. 302–316. IOS Press, (June 1998).
- [31] I. Nonaka and H. Takeuchi, *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press, 1995.
- [32] K. O'Hara, *Plato and the Internet*, Icon books, 2002. ISBN: 1840463465.
- [33] K. O'Hara, H. Alani, and N. Shadbolt, 'Identifying Communities of Practice: Analysing Ontologies as Networks to Support Community Recognition', in *Proceedings of the 2002 IFIP World Computer Congress*, Montreal, Canada, (August 2002).
- [34] K. O'Hara and N. Shadbolt, 'Issues for an Ontology for Knowledge Valuation', in *Proceedings of the IJCAI'01 workshop on E-Business and the Intelligent Web*, Seattle, WA, USA, (August 2001).
- [35] D. O'Leary, 'Using AI in Knowledge Management: Knowledge Bases and Ontologies', *IEEE Intelligent Systems*, **13**(3), 34–39, (June 1998).
- [36] C.D. Paice, 'A thesaural model of information retrieval', *Information processing and management*, **27**(5), 433–447, (1991).
- [37] Ph. Perez, 'Corporate Memory Management through Agents', in *Proceedings of the eBusiness and eWork Conference*, Madrid, Spain, (2000).
- [38] M.R. Quillian, 'Word concepts: A theory and simulation of some basic semantic capabilities', *Behavioral Science*, **12**, 410–430, (1967).
- [39] R. Rada, H. Mili, E. Bicknell, and M. Blettner, 'Development and application of a metric on semantic nets', *IEEE Transactions on systems, management, and cybernetics*, **19**(1), 17–30, (1989).
- [40] Reimer,U. and Margelisch,A. and Novotny,B. and Vetterli,Th., 'EULE2: A knowledge-based system for supporting office work', SIG-GROUP Bulletin 19(1), ACM, (1998).
- [41] R. Smith and A. Farquhar, 'The Road Ahead for Knowledge Management: An AI perspective', *AI Magazine*, **21**(4), 17–38, (2000).
- [42] R. Sol and J. Serra, 'Netexpert: A multiagent system for expertise location', in *Proceedings of the IJCAI'01 Workshop on Knowledge Management and Organizational Memories*, Seattle, USA, (August 2001).
- [43] T. Summer and S. Buckingham-Shum, 'From Documents to Discourse: Shifting Conceptions of Scholarly Publishing', in *proceedings of the CHI'98: Human Factors in Computing Systems*, Los Angeles, CA, USA, pp. 95–102. ACM Press, (1998).
- [44] M. Vargas-Vera, J. Domingue, Y. Kalfoglou, E. Motta, and S. Buckingham-Shum, 'Template-driven information extraction for populating ontologies', in *Proceedings of the IJCAI'01 Workshop on Ontology Learning*, Seattle, WA, USA, (August 2001).
- [45] E. Wenger, *Communities of Practice: The Key to Knowledge Strategy*, Cambridge University Press, 1998.
- [46] E. Wenger, 'Communities of Practice: The Key to Knowledge Strategy', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. 3–20. Butterworth-Heinemann, (2000). first published 1999.
- [47] P. Zave and M. Jackson, 'Four Dark Corners of Requirements Engineering', *ACM Transactions on Software Engineering and Methodology*, **6**(1), 1–30, (January 1997).