Open Research Online

Initiating organizational memories using ontology-based network analysis as a bootstrapping tool

Journal Item

For guidance on citations see FAQs.

Version: Accepted Manuscript

oro.open.ac.uk

Initiating Organizational Memories using Ontology-based Network Analysis as a Bootstrapping Tool

Harith Alani, Yannis Kalfoglou, Kieron O'Hara, Nigel Shadbolt

Intelligence, Agents, Multimedia Group
Electronics and Computer Science Department
University of Southampton
Highfield, SO17 1BJ. Southampton. Hampshire. UK.
Tel: (44) 2380594492, Fax: (44) 2380592865
Email: {ha, y.kalfoglou,kmo,nrs}@ecs.soton.ac.uk

Abstract

An important problem for many kinds of knowledge systems is their initial set-up. It is difficult to choose the right information to include in such systems, and the right information is also a prerequisite for maximizing the uptake and relevance. To tackle this problem, most developers adopt heavyweight solutions and rely on a faithful continuous interaction with users to create and improve content. In this paper, we explore the use of an automatic, lightweight ontology-based solution to the bootstrapping problem, in which domain-describing ontologies are analysed to uncover significant yet implicit relationships between instances. We illustrate the approach by using such an analysis to provide content automatically for the initial set-up of an organizational memory.

1 INTRODUCTION

In this paper we discuss an approach to the problem of *bootstrapping* in knowledge system management. In general, we refer to any system designed to manage, transfer, acquire, or filter knowledge within an organization as a knowledge system; hence a knowledge system is indexed to an organization which will have its own priorities and idiosyncrasies. Knowledge systems have a number of properties, including formal functional properties, describing what they do to the knowledge, and it is on these properties that system developers generally focus. However, of course, because knowledge systems are overlaid onto organizational contexts, they have organizational constraints too.

One important constraint is that they fit into existing organizational patterns of behaviour, which they integrate with current working practice more or less smoothly. There are two reasons for this. First, obviously, if they integrate it is easier to implement them; you don't have to dig up the entire organization root and branch for a new system with all the uncertainty that implies. But secondly, such systems often attempt to glean or skim knowledge from working practices, to gain knowledge as a by-product of the normal functions of the organization. As knowledge-based tasks and practices take place, the knowledge system functions as a recording or documentation device, and may then use the information thus gathered to generate still more.

There are many examples of such systems. For example, anyone seeking knowledge from the World Wide Web is likely to do some browsing. But if that browsing is monitored by an intelligent user profiler, then the browsing task is now yielding information to a knowledge system where it did not before, viz., information about what pages are browsed, and depending on what is downloaded, information about what the user thinks is significant. Such information, which was being lost as soon as it was generated before the system was in place, once gathered can be used to generate further knowledge: for example, the browsed websites might be used as raw material for developing an ontology of the domain; or perhaps it can be used with an

1

ontology of the domain to recommend further websites that share characteristics with the browsed ones, and therefore may also be of interest.

Such systems typically need to be used if they are to generate interesting information – they take standard organizational actions as their input. If they are not used, then they remain either content-free, or stuck only with the boring skeletal knowledge with which they are primed or seeded. Furthermore, in the early stages, when the system contains little knowledge about the organization, its use may be difficult to integrate, in that there may need to be a lot of form-filling and formal information extraction for its users before the system contains sufficient knowledge about the domain to work smoothly, which of course is off-putting for the early users. This then can lead to the bootstrapping problem: the system must be used before it has any interesting content, but that precludes its having any interesting content in its initial conditions. Therefore, in its initial state, it will have no takers. An obvious solution is to order employees to use it, but then the knowledge system can hardly be said to integrate smoothly with current organizational practices (Smith and Farquhar 2000).

In this paper, we want to suggest an ontology-based method of generating information for bootstrapping purposes which can alleviate this problem to some extent. In the next section, we introduce the principles of ontology-based network analysis, a method for extracting information that lies latent in ontologies, and ONTOCOPI, a tool for doing this, illustrating the ideas with a couple of examples of using ONTOCOPI for bootstrapping. In the following section we will develop a case study of the use of ontology-based network analysis to bootstrap on type of knowledge system, organizational memories.

2   ONTOLOGY-BASED NETWORK ANALYSIS

2.1   Principles of Ontology-Based Network Analysis

Ontology-based Network Analysis (ONA, Alani et al 2002a, O'Hara et al 2002) 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. O'Hara et al 2002). However, the advantage of studying ontologies is that the concepts and relations therein have explicit semantics or types, and therefore providing 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 2.2 and an example algorithm is detailed in (Alani et al 2002a).

ONA methods can be harnessed to address the bootstrapping problem of knowledge systems by using populated ontologies already in place in organizations to uncover the knowledge required to initialise such systems. The fact that the method is automatic takes some of the burden of system start-up from its users or developers, 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 knowledge systems. 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 in such applications, but as the knowledge systems develop, such assumptions can be relaxed, as the population of the systems begin to happen by their users. And user feedback to the initial setup will always be essential.

The following section discusses an application that uses ONA for the purpose of identifying communities of practice within an organization, and how this system is used to bootstrap knowledge systems.

2.2 ONTOCOPI

ONA was originally conceived as a method of getting to grips with the problem of identifying Communities of Practice (CoP, Kalfoglou et al 2002, O'Hara et al 2002) within organizations. To this end, a particular instantiation of ONA was created, using an ontology-based spreading activation algorithm to search the knowledge base, moving from instance to instance along a weighted graph defined by the relationship connections in the ontology, together with a representation of their importance for the particular community. Enough relationship connections of sufficient weight implied that two instances were in the same community. The system is called ONTOCOPI (ONTOlogy-based Community Of Practice Identifier – Alani et al 2002a), and is currently implemented as a Protégé (Musen et al 2000) plug-in as well as a standalone Web-accessible program.

ONTOCOPI's algorithm combines and improves ideas from previous work on similarity measures, such as shortest path measures (Rada et al 1989), multi-path traversal (Paice 1991), and constrained spreading activation methods (Cohen and Kjeldsen 1987). 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.
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 identification. The assumption of ONTOCOPI is that a formal relationship can stand as proxy to an informal one. Hence ONTOCOPI infers that two people who co-author a paper are more likely to be members of the same CoP, for example. 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. This is not an exact science; the aim of ONTOCOPI is only to *support* CoP identification, a very expensive operation in its own right (Wenger 1999, O'Hara et al 2002). But on this basis, what ONTOCOPI is intended to do is to discover vector additions of relationships within an ontology, in which the relations can be weighted according to their importance for the particular task in hand. It is this property that is important for bootstrapping.

It is important to note that ONTOCOPI can not identify relationships that are not there: if two people in the same CoP simply have no formal relationships 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.

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 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 any instances selected to be ignored, and centre right bottom displays the weights that have been transferred to other instances, in descending order (i.e. a rough specification of the CoP, the main output of ONTOCOPI). In Figure 1, the CoP of *Shadbolt* has been investigated, and ONTOCOPI has suggested, in descending order of preference, *O'Hara*, *Elliott*, *Reichgelt*, *Cottam*, *Cupit*, *Burton*, *Crow*, *Rugg* and so on.
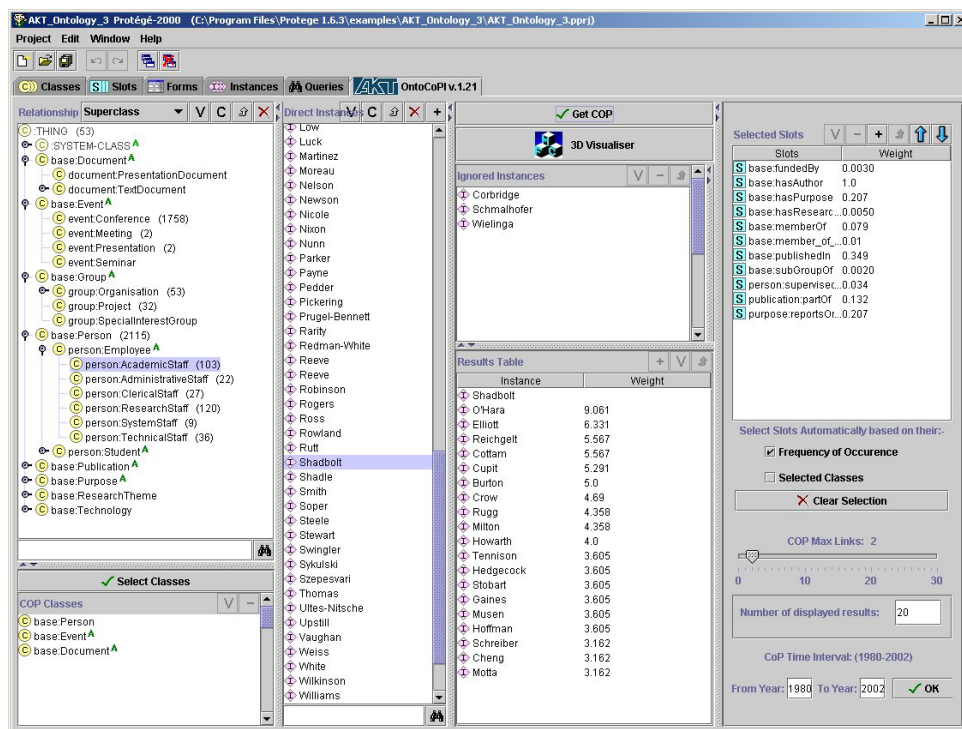

Figure 1. A screenshot of ONTOCOPI as a Protégé plug-in.

Order is important, so are the relative weights. *O'Hara* scores 9; this is meaningless except in the context of a particular search. Here, 9 is very good, 50% higher than the score of the next candidate. On the other hand, the user may be more suspicious of the ordering of, say, *Tennison*, who scores 3.6, and *Motta*, who scores 3.1. The figures themselves have no constant interpretation (except in terms of the algorithm); it is for the users to take the suggestions and interpret them according to their own understanding of the structure of their CoP.

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 in the analysed knowledge base getting a weight value of 1, those not used at all getting 0, and the others being allocated in proportion 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.

ONTOCOPI's algorithm initialises 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 *Gaines* appears on the radar.

The *raw* algorithm can be refined according to user feedback. Manual setting of relation weights has already been mentioned. Other ways to control variables include privileging certain classes, or differential weighting of instances. The appropriate refinements in a particular domain will depend on the features of the domain itself, and what is captured by the ontology.

2.3   Applications of ONTOCOPI

As noted above, ONTOCOPI was originally developed in order to address the problem of CoP identification. However, because its primary purpose is to extract information implicit in ontologies, it can be used in the context of the bootstrapping problem in the following manner.

If there is a domain in which a knowledge system is about to be introduced, and if there exist representations of that domain in the form of ontologies, ONA can be used to extract information about connectivity within those ontologies. This implicit information, made explicit, can then be used to populate the knowledge system. In this way, ONA makes it possible to circumvent the bootstrapping problem, partially at least, by providing extra domain information, which may be used as an entry point into the use-acquisition cycle. Of course, it will not be sufficient in every case, sometimes because the knowledge system requires a lot of knowledge content for bootstrapping to begin, sometimes because the connectivity knowledge implicit in the ontology will not be the right sort of knowledge. But there are cases where ONTOCOPI has a role to play in bootstrapping.

In the next section, we will discuss the case of initialisation of organizational memories in some detail. But as a pointer to ONTOCOPI's value in such circumstances, we will first briefly sketch some other bootstrapping applications.

2.3.1   Recommender Systems

One type of knowledge system that suffers from the bootstrapping problem is the recommender system, i.e. a system which learns about user preferences for, say, web pages, over time, and automatically finds new pages that are similar to the user's historical preferences. Again, a new system, or a new user of an established system, has to begin afresh, and the result is likely to be poor performance initially. This can be self-destructive as users are put off and the recommender never harvests enough information to achieve good performance.

The Quickstep system (Middleton et al 2002) is intended to recommend online research papers to researchers; user browsing is monitored via a proxy server, and a nearest neighbour algorithm classifies browsed URLs based on a training set of labelled sample papers. Explicit feedback and browsed URLs form the basis of the interest profile of the user, and a daily set of recommendations is computed; the user can offer feedback to improve the training set and classification accuracy as the system is used.

Integration between Quickstep and ONTOCOPI has been investigated (Middleton et al 2002), using ONTOCOPI to help with the bootstrapping problem for new users. Upon start-up, an ontology gives Quickstep an initial set of publications for each user. Each user's known publications are correlated with Quickstep's paper database, and a set of interests is then generated for the users. This provides one way of overcoming the initialisation problem for the new system. Then when a new user is added the ontology provides the publication list of the new user (i.e. the new user's papers that have already made it into the ontology), and ONTOCOPI provides the CoP of the new user, i.e. performing an ONA using the new user as a starting node. The CoP of the user, in this case a list of the most similar users, can then feed into a correlation between the user's history of publication and similar user profiles to form an initial profile of the new user.

2.3.2   *Referential Integrity*

A second example application area for ONTOCOPI is within ontology development itself. Since ontologies will be a central technology for the development of the semantic web, we can expect to see more being developed, and, in particular, more being created for hybrid domains by merging legacy ontologies, databases, and other information stores. In that case, it is inevitable that there will be problems preserving referential integrity across the components of the merger; the same object/concept may be referred to with different names (e.g. Alani, Harith and H. Alani), or alternatively different objects may be referred to using the same name (e.g. two John Smiths). Although it is often possible to live with such referential confusion, clearly the use of an ontology and any knowledge service relying on it will be less efficient if it does not have referential integrity. Even something as minor as a typographical error could impair effectiveness.

The field of ontology-merging tools and techniques has shown that even relatively unsophisticated heuristics for uncovering conflicts can be quite effective, and this fact has been exploited in the development of a stepwise approach (Alani et al 2002b). To begin with, an ontology is populated, using semi-automatic methods, from existing resources; then soft string similarity measures and a set of generic heuristics (e.g. if two different names have equivalent personal attributes; telephone number, address, email, etc., the entities could be identical) are used to cluster potential instance duplicates.

This leads onto a second step, in which ONA is used to analyse the connections between the instances thus clustered. Using the clustered instances as the starting nodes of the analysis, the CoPs of the instances can be calculated, and the degree of overlap between the CoPs can be calculated as a similarity value; if the value is over some threshold, then the two instances can be assumed to be identical. Naturally, such an approach is bound to fail on some occasions, particularly for names that are not well connected within the ontology. The more connected the individuals, the more likely it is that the similarity of their CoPs will be correlated with their identity or distinctness.

This approach is currently under test within the Advanced Knowledge Technologies (AKT 2001), using a reference ontology created from legacy resources such as personnel databases, publications databases, and so on, harvested from the websites initially of the AKT partners, and intended ultimately to serve the UK computer science community with a repository of knowledge about, for example, declared areas of research or currently active projects. The scale of such an application will inevitably involve referential promiscuity, and will give plenty of scope for testing the use of ONA for preserving referential modesty.

3   ORGANIZATIONAL MEMORIES

For our major case study, we will look at organisational memories (OM). 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 (e.g. Abecker et al 1998), 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 and populate the OM accordingly. The process 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 with (Abecker et al 1998). 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 view on OMs is in terms of knowledge delivery. There have been two ways of delivering knowledge reported in the literature: *pull* and *push* knowledge (O'Leary 1998). The former refers to technologies that aim at pulling knowledge from vast repositories of data to people, where 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.

Push technologies in knowledge delivery have been much less prominent, probably because of 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 (Abecker et al 2000) 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 sceptical about two 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 to only those resources that are considered 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 (Abecker et al 2001).

3.1   The problem of resource selection

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 (Dieng et al 1999) as well as in implemented systems (e.g. Abecker et al 2000, Althoff et al 1999). 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, but also concerned with the nature of these resources which could be *unspecified*, in that they are vaguely expressed, need to be composed by a number of related resources or are external to the organization, and also used by other systems within the organization, which incidentally also serve users in their quest for valuable information. 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 exploit 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 (Dieng et al 1999). 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 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 (1997), vague and imprecise requirements are always difficult to formalize and subsequently convert to specifications in the early phases of software development. In the case of OMs, we should expect these requirements to be incomplete and vague.

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 KnowMore OM. Our own approach is to aim for 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 for 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 (Gresse-vonWangenheim et al 2001) 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. As we have seen, this problem can be solved, but at a cost: more systems and methods have 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 on how our method sets up a comprehensive OM in an automated fashion.

3.2   Seeding the OM

Our working hypothesis is that since we already use ontologies in OMs for the purposes of semantic interoperability and reuse, we could also use them in other ways. We could analyse their structure by taking into account relationships between their constructs.

To investigate this hypothesis, we are looking at the application of ONA 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.

The popular entities that have been identified are used as the initial seed to populate the OM, thus setting it up to contain 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 relevance of retrieved information. 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 control the output of the automated content-delivery mechanism. Initially though, we elaborate on the crux of the problem on how to select the right resources for seeding the OM.

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).

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

3.3 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 (Kuhn and Abecker 1997). This figure emphasises 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. The focus in this paper is on the use of ONA; the two rectangular boxes placed between the ontologies and OM interfaces. The generic nature of ONA makes it possible to use it for pushing knowledge to users but also as an aid for the OM developers. They could apply ONA to the organization's ontologies in order to identify which concepts should be presented to certain type of users. For instance, assuming that there is a workflow

9

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 (Abecker et al 2000), 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 familiarisation 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 (Althoff et al 1999), 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 is the Digital Document Discourse Environment (Sumner and Buckingham Shum 1998) 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 derived from the Web version of ONTOCOPI is illustrated in Kalfoglou et al (2002).

4   DISCUSSION

The method described above is not infallible or easily adaptable to any existing OM setting. We identified potential caveats on using ONA to bootstrap OMs and categorize them in three broadly defined areas: information overload, context-awareness and domain-independence:

- *Information overload*: progressive and query-based interaction with the OM from initial set-up acts as a safeguard against information overload. However, progressive interaction means that the initial set-up may suffer from 'cold-start' syndrome. And query-based interaction requires expertise and domain familiarization from the users to get the most out of an OM. There is not a golden rule to follow when we 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 ontology 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.

- *Context-awareness*: this has been recognized as the Achilles' heel for OMs. Marrying workflow processes and OMs seems to work well in tackling context-awareness in settings where workflow processes either exist, or are relatively easy to identify and model. ONA takes a different approach. We do not assume that workflow processes will exist, but we rely on ontological resources. Contextual relevance can be achieved in a number of ways. We could rely on ad-hoc technologies, such as profiling users' interests by using agents (Perez 2000) embedding personalization facilities in thin Web clients (Kalfoglou et al 2001), or identifying users' tasks (Budzik and Hammond 2000). 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*: ONA is generic. This makes it possible to apply ONA to more than one ontology as are likely to exist in large organizations. 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.

5   ACKNOWLEDGEMENT

6   REFERENCES

1. Abecker A., Bernardi, A., Hinkelmann, K., Kuhn, O., and Sintek, M. Toward a Technology for Organizational Memories, IEEE Intelligent Systems, 13(3), 40-48, June 1998.

2. Abecker, A., Bernardi, A., Hinkelmann, K., Kuhn, O., and Sintek, M. 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. Abecker, A., Bernardi, A., Ntioudis, S., Mentzas, G., Herterich, R., Houy, C., Muller, S., and Legal, M. The DECOR Toolbox for Workflow-Embedded Organizational Memory Access. Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS'01), Setubal, Portugal, 2001.

4. AKT Project, *The AKT Manifesto*, http://www.aktors.org/publications/Manifesto.doc, 2001

5. Alani, H., O'Hara, K., and Shadbolt, N. ONTOCOPI: Methods and Tools for Identifying Communities of Practice. Proceedings of the 2002 IFIP World Computer Congress, Montreal, Canada, 2002a.

6. Alani, H., S. Dasmahapatra, N. Gibbins, H. Glaser, S. Harris, Y. Kalfoglou, K. O'Hara and N. Shadbolt. Managing Reference: Ensuring Referential Integrity of Ontologies for the Semantic Web. 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02, Siguenza, Spain, 2002b.

7. Althoff, K-D., Nick, M., and Tautz, C. Improving Organizational Memories through User Feedback. Proceedings of the Learning Software Organizations (LSO'99) Workshop, Kaiserslauten, Germany, ed., F. Bomarius, pp. 27--44, 1999.

8. Budzik, J. and Hammond, K.J. User Interactions with Everyday Applications as Context for Just-In-Time Information Access. Proceedings of the 2000 International Conference on Intelligent User Interfaces, ACM, 2000.

9.   Cohen, P. and Kjeldsen, R. Information Retrieval by Constraint Spreading Activation in Semantic Networks. Information Processing and Management, 23(4), 255-268, 1987.

10.   Dieng, R., Corby, O., Giboin, A., and Ribiere, M. Methods and Tools for Corporate Knowledge Management. International Journal of Human-Computer Studies (IJHCS), 51, pp. 567-598, 1999.

11.   Gresse-vonWangenheim, C., Lichtnow, D., vonWangenheim, A. and Comunello, E. Supporting Knowledge Management in University Software R&D Groups. 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, 2001.

12.   Kalfoglou, Y., Domingue, J., Motta, E., Vargas-Vera, M., and Buckingham-Shum, S. My Planet: an Ontology-Driven Web-based Personalised News Service. Proceedings of the IJCAI'01 Workshop on Ontologies and Information Sharing, Seattle, USA, 2001.

13.   Kalfoglou, Y., H. Alani, K. O'Hara and N. Shadbolt. Initiating Organizational Memories using Ontology Network Analysis. Knowledge Management and Organizational Memories workshop, 15th European Conference on Artificial Intelligence, Lyon, France, 2002.

14.   Kuhn, O., and Abecker, A. Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges. Journal of Universal Computer Science, 3(8), pp.929-954, 1997.

15.   Middleton, S., H. Alani, N. Shadbolt, and D.C. De Roure, "Exploiting Synergy Between Ontologies and Recommender Systems," *The Semantic Web Workshop, World Wide Web Conf.*, (WWW'02), Hawaii, USA, pp. 41-50, 2002.

16.   Musen, M. A., R. W. Fergerson, W. E. Grosso, N. F. Noy, M. Y. Grubezy and J. H. Gennari. Component-based support for building knowledge-acquisition systems. Proceedings of the Intelligent Information Processing (IIP) Conference of the International Federation for Processing (IFIP), World Computer Congress (WCC'2000), Beijing, China, 2000.

17.   O'Hara, K., Alani, H., and Shadbolt, N. Identifying Communities of Practice: Analysing Ontologies as Networks to Support Community Recognition. Proceedings of the 2002 IFIP World Computer Congress, Montreal, Canada, 2002.

18.   O'Leary, D. Using AI in Knowledge Management: Knowledge Bases and Ontologies. IEEE Intelligent Systems, 13(3), pp. 34-39, 1998.

19.   Paice, C.D. A Thesaural Model of Information Retrieval. Information Processing and Management, 27(5), pp. 433-447, 1991.

20.   Perez, Ph. Corporate Memory Management through Agents. Proceedings of the eBusiness and eWork Conference, Madrid, Spain, 2000.

21.   Rada, R., Mili, H., Bicknell, E., and Blettner, M. Development and Application of a Metric on Semantic Nets. IEEE Transactions on Systems, Management, and Cybernetics, 19(1), pp. 17-30, 1989.

22.   Smith, R. and Farquhar, A. The Road Ahead for Knowledge Management: An AI Perspective, AI Magazine, 21(4), pp. 17-38, 2000.

23.   Sumner, T. and Buckingham-Shum, S. From Documents to Discourse: Shifting Conceptions of Scholarly Publishing. Proceedings of the CHI'98: Human Factors in Computing Systems, Los Angeles, CA, USA, pp. 95-102. ACM Press, 1998.

24.   Wenger, E. Communities of Practice: The Key to Knowledge Strategy. Knowledge and Communities, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. 3-20. Butterworth-Heinemann, 2000. first published 1999.

25. Zave, P. and Jackson, M. Four Dark Corners of Requirements Engineering. ACM Transactions on Software Engineering and Methodology, 6(1), pp. 1-30, 1997.