Chapter XVII
Knowledge Management Support for Enterprise Distributed Systems

Yun-Heh Chen-Burger
The University of Edinburgh, UK

Yannis Kalfoglou
University of Southampton, UK

ABSTRACT

Explosion of information and increasing demands on semantic processing Web applications have pushed software systems to their limits. To address this problem, we propose a semantic-based formal framework (ADP) that makes use of promising technologies to enable knowledge generation and retrieval. We argue that this approach is cost-effective, as it reuses and builds on existing knowledge and structure. It is also a good starting point for creating an Organizational Memory and providing Knowledge Management functions.

BACKGROUND

The era we are living in is characterized by an unprecedented explosion of information that is digitized and available to large audiences through online, distributed, and open-ended environments. Presented with it are also opportunities to exploit and benefit from it. Organizations have to quickly adapt to this new phenomenon. Software applications, database and expert systems designed and run by a closed group of software and knowledge engineers who had centralized control over the lifecycle of IT artefacts seem to be outdated. Moreover, the distributed nature of IT systems has experienced a dramatic explosion with the arrival and revolutionary use of the Internet and its associated technologies—hypertext and XML-based documents, online databases, terminological repositories, Web services, and blogs—which continually challenge the traditional roles of IT in our society.

One promising approach for IT system architects is to use intelligent knowledge management (KM) methods to cope with this expanding nature of distributed systems in a global scale. At the cornerstone of most of these tools lies the buzzwords of *semantics technologies* that are deployed in the Semantic Web (SW). Semantic technology is a broad term coined recently in the business domain to refer to technologies ranging from ontologies and information extraction on the SW to ebXML schemata and service-oriented architecture based systems. This term enables synergies in distributed systems that automate semantic (meaning) interoperability between processes and services.

The successful blending of semantic technologies with the traditional KM systems starts from a fundamental part of any business: *process.* For more than 10 years, the values of process-oriented approaches, such as BPR (business process reengineering) and BPI (business process improvement) are well-recognized. Today, it is one of the fundamental steps to radically improving organizational performance. Processes are treated as tangible entities that can be formally captured, analyzed, incrementally, and radically modified to change organizational behaviors and achieve goals. Recent KM that have taken process-oriented approaches are exemplified in Schreiber, de Hoog, Akkermans, Anjewierden, Shadbolt, and Van de Velde (1999) and Abecker, Bernardi, Hinkelmann, Kuhn, and Sintek (1998).

KM is no longer just about identifying and storing knowledge, but also about providing efficient ways to retrieve, disseminates, and use knowledge to achieve goals. It embeds "KM processes" as a part of normal business practices, so no more than necessary efforts need to be spent to benefit from KM. Furthermore, KM as a discipline can benefit from process-oriented approaches. KM tasks can be described in learnable processes that can be compared with business processes, analyzed and improved upon. In addition, in KM, the human is the central issue—they are the key knowledge creators, holders, and users; organizational memories (OM) are often the main tool to hold and provide information central to an organization. In this chapter, we therefore examine the roles played by human, OM, and business processes in an organization and how they relate to each other. We also speculate that formal logical methods, such as the proposed semantic-based Actor, Data and Process-oriented (ADP) framework, can interface these fundamental organizational components to help improve the utilization of an OM, thus leading to organizational performance enhancements. We start our exploration of KM support for enterprise distributed systems by focussing on a core component of many enterprises: the OM.

ORGANIZATIONAL MEMORIES

We witness a shift in the decision support literature from data-oriented processing systems to ones integrated with human intellect and organizational processes (Carlsson & Turban, 2002). These have been studied in the context of KM and OM to provide means for easy access and retrieval of information for users. In parallel, we see recognition that the goals of KM will be most effectively realized through actions connected to normal day-to-day business processes (Breuker & Van de Velde, 1994). This makes it easier to demonstrate value-added contributions to an organization, which is better than isolated KM efforts (Abecker et al., 1998). An ideal OM could assist in effective decision-making, which means information regarding the organization could be made easily accessible.

However, there is little support to help create an OM. It is difficult to identify the right information to include. This process is time-consuming, manual, and error-prone, given the diversity, quality, and quantity of resources to be analyzed for reliability and relevance. Semi-automatic methods *do* exist, but these are bound to individual technologies. It is always the user who has to initiate search in the OM. But this requires the user to be able to formulate a query, with or without automated help; the OM system must be able to correctly parse this query, retrieve relevant information according to predefined mechanisms, and present it back to the user.

Several issues are identified in field surveys (Dieng, Corby, Goboin, & Ribiere, 1999) and systems (Abecker et al., 1998). It is a multifaceted problem because it is not only concerned with the quality and elicitation of resources, or the difficulties in engaging the user in technical tasks, but is also related to the usage of these resources. They may be (a) used by other systems for different purposes, (b) "unspecified" or "ambiguous" and need to be interpreted or composed by other (external) resources, and (c) once these resources are identified and used, they act as a qualitative measure for the OM. That is, if OM users are not satisfied with the quality of information presented to them, it is unlikely that they will return.

One way to tackle this problem is to identify the purpose of an OM project early on (Dieng et al., 1999): what are the users' needs and what will the OM be used for. Most techniques and methods are taken from requirements analysis and elicitation research. However, one should be cautious when using requirement engineering techniques. Zave and Jackson (1997) reported that vague and imprecise requirements are difficult to formalize and convert to specifications and further refinements are necessary.

This problem has led some OM designers to build their systems around existing workflow process engines, for example, the *KnowMore* OM (Abecker et al., 1998). We are sceptical of this approach, as it requires familiarization and the existence of robust workflow processes, supported by intensive modeling to link the two systems. We therefore propose an ontology based approach for seeding OMs.

SEEDING ORGANIZATIONAL MEMORIES USING ONTOLOGY NETWORK ANALYSIS

Since it is common that ontologies are used in organizations—for semantic interoperability and reuse—one could also use them for other purposes. Ontology network analysis (ONA) (Alani, Kalfoglou, O'Hara, & Shadbolt, 2002) applies information network analysis methods to a populated ontology to uncover trends and object characteristics, such as shortest paths, object clusters, semantic similarity, object importance, and/or popularity. Similar methods have also been explored for information retrieval purposes. ONA uses these methods to analyze the network of instances and relationships in a knowledge base, guided by ontology. There are many types of networks that can be studied (O'Hara, Alani, & Shadbolt, 2002). The advantage of studying ontologies is that the relations therein have semantics that provide rich sources of information over and above connectivity or simple subsumption. This semantic information can be used to enable "raw'" results to be refined on a relatively principled basis. An example ONA application is described in another section of this chapter.

ONA methods can be harnessed by selecting a set of focused resources to feature in a new OM based on existing populated ontologies. The fact that this method is automatic takes some of the burden of OM development from its users and managers and allows 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 entirely foolproof. Points of interests may not be spotted, especially if the ontology is incomplete or fails to cover some important aspects in the domain; however, by extracting information from ontologies currently in use, ONA suggests an initial reliable set of interesting concepts and relations. Certain assumptions must be made to support the use of ONA, but as the OM develops, such assumptions are relaxed, and the OM begins to be populated by its users.

During the seeding exercise, the ONA technique is used to carry out network measures to an ontology to determine popular entities in the domain. Such entities can be either classes or instances, and *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. The working assumption is that 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 (i.e., they will act as "hubs" in the domain).[1]

ONTOLOGY NETWORK ANALYSIS ALGORITHM

Given a first pass ONA of an ontology, given the most popular entities, an OM developer can exploit user feedback to hone the analysis. Two ways of doing this are:

- 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.
- 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 differentially weighting relations.
  - 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.
  - 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.

The spreading activation algorithm underpinning ONA 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 a quantitative one. However, Cooper (1997) argues that 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 that *by default* represents important concepts, then these concepts are also important.

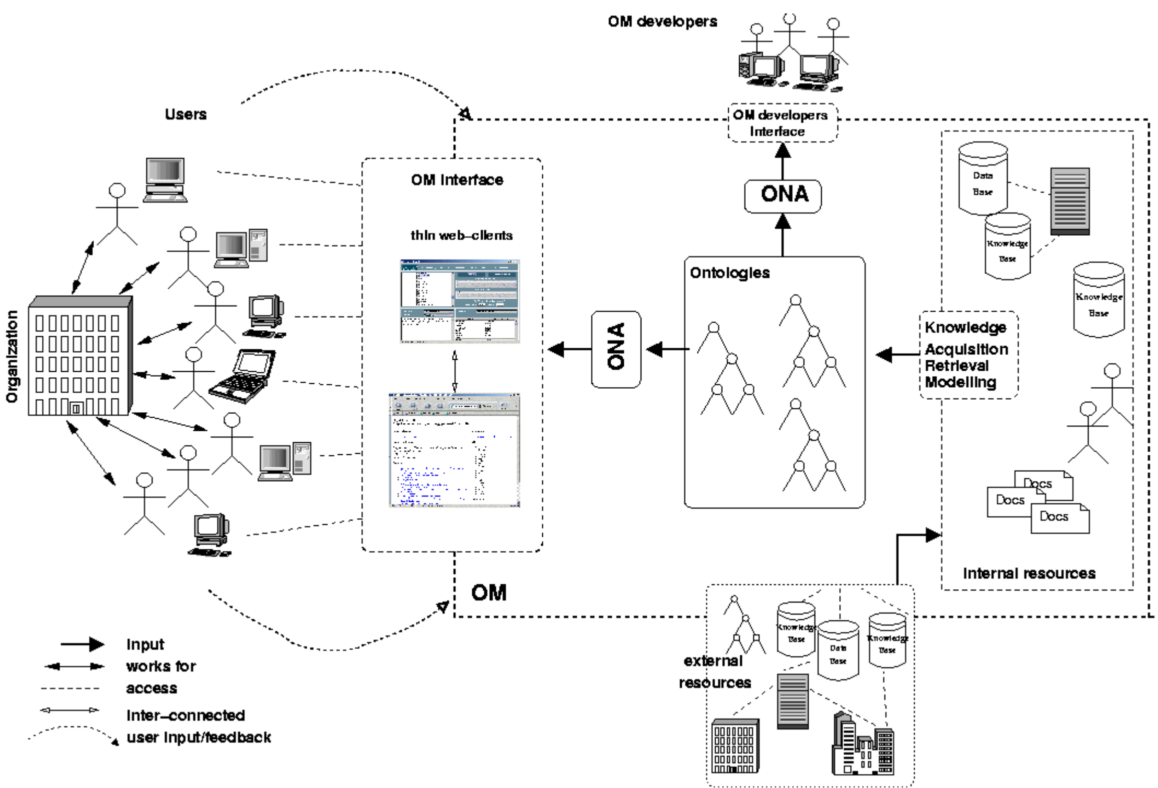
To operate our hypothesis, we assume that (a) ontologies will be available in the organization in which we want to deploy an OM, and (b) these will be populated. These assumptions are strong and indeed are ongoing research issues in the knowledge engineering community. However, we should accept and anticipate that ontologies are popular in organizational settings nowadays: in the form of database repositories, SW data formatted in RDF/RDFS, and OWL ontologies.

USING ONTOLOGY NETWORK ANALYSIS IN ORGANIZATIONAL MEMORIES

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 ontology (e.g., documentation slots).

In Figure 1 we depict a high-level diagram of an OM. This is not meant to be a reference architecture for OMs. 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 center 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 knowledge bases or digital discussion spaces. In the center of our abstract OM lie the ontologies that underpin the entire OM. These are either existing resources or are constructed (semi-) automatically with the aid of knowledge acquisition, retrieval, and modeling techniques. The focus, though, is on the use of ONA: the two rectangular boxes denoting "ONA" are placed between the ontologies and OM interfaces to users and developers.

Figure 1. Supporting initial seeding of an OM: Pushing knowledge into the OM and pulling it out—using ontology network analysis and business process analysis techniques



The generality 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 modeling exercises such as those reported in (Abecker et al., 1998), or they could use ONA to help them identify concepts from the underlying ontologies and map them onto workflow processes. The developers can then use these concepts found used in the workflow processes as a starting node for his/her next round of ONA analysis. This could reveal further node linkages, thus saving development time and allowing developers to deal with ontologies that they are not familiar with.

We also include two curly dotted arcs in Figure 1 linking users with the OM. These denote users' feedback and input. This is a very important element of an OM architecture, as OMs can be improved over time by user feedback and inputs. In our abstract architecture, we implemented light-weight feedback mechanisms, like thin Web-clients, accessible through Web browsers, as a means for eliciting feedback on an OM's resources (see Figure 2). Finally, the OM interface to its users is light-weight and accessible from distributed clients on the Web. We have developed two kinds of interfaces: 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.

LIMITATIONS

We identified potential caveats to using ONA to bootstrap OMs and categorize them in three areas:

a. **Information overload:** A progressive and query-based interaction with the OM from initial set-up acts as a safeguard against unwanted information overload. However, progressive interaction means that the initial set-up suffers 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.
b. **Context-awareness:** This has been recognized as the Achilles' heel for OMs. One proposed remedy, advocated by proponents of marrying workflow processes and OMs, seems to work well only in settings where workflow processes are either existing, or are relatively easy to identify and model.
c. **Domain-independence:** This is a desired feature for OMs. But, the proposed ONA approach 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.

The ONA-based solution we proposed above addresses the problem of setting up a comprehensive OM in a bid to attract high usage rates. However, in a dynamic and ever-changing organizational context, we are faced with a number of challenges related to the capturing of the right types of (business) requirements. In the next section we elaborate on how we assist the appropriate capturing of organizational requirements with the use of a novel business process approach that is geared towards supporting an OM, thus extending and complementing our ONA based method for seeding the initial OM.

A CASE STUDY: ONTOCOPI

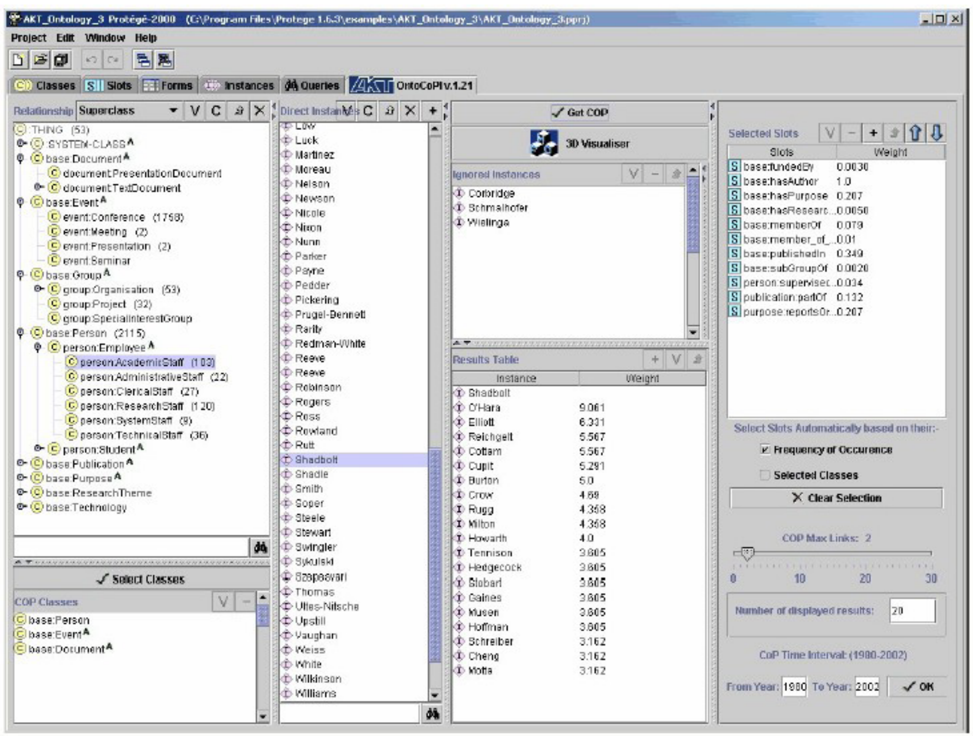
An example application is to use ONA to identify communities of practice (CoP) within organizations. One such tool is ONTOlogy-based community of practice identifier (ONTOCOPI) (O'Hara et al., 2002).

ONTOCOPI's algorithm combines and improves ideas from previous work on similarity measures, such as shortest path measures (Rada, Mili, Becknell, & Blettner, 1989), multi-path traversal (Paice, 1991), and constrained spreading activation methods (Cohen & Kjeldsen, 1987). Its algorithm makes 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 described formally. They stand as proxies to informal ones—the types of relationships found in CoPs. One may infer that two people who co-author a paper are likely to be members of the same CoP. If two CoP members share no formal relationships, then any vector addition of formal relations can also stand proxy for informal ones. For instance, if A co-authored a paper with B, who works on a project with C, then it may be inferred that A and C, are likely to be members of the same CoP. Total accuracy, however, is impossible for an informal and rapidly-evolving social group like a CoP. Furthermore, the aim of ONTOCOPI is to *support* CoP identification, an expensive operation in its own right (Wenger, 1999). A certain measure of indeterminacy is inevitable.

ONTOCOPI cannot identify relationships that are not represented: if two people in the same CoP have no formal relationship recorded in the ontology, and no chain of formal relations link them, then their comembership cannot be found. ONTOCOPI also can't distinguish *between* CoPs. If someone is a *broker*, that

Figure 2. ONTOCOPI: Ontology based communities of practice identifier



is, a person who functions in two separate CoPs, then ONTOCOPI will tend to select the union of the two CoPs. ONTOCOPI, however, does support CoP identification, a resource-heavy task that may be alleviated to some extent by assumptions that formal connections can approximate informal relationships.

Figure 2 shows an ONTOCOPI'S interface. 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 "center" of the CoP (the relations radiating out from this individual are used for CoP identification). The panels on the right set the relation weights and parameter values (e.g., the number of links the algorithm will spread to). Clicking the "Get COP" button will run the algorithm. The center right top panel displays the current calculations and center right bottom displays the weights that have been transferred to other instances, in descending order of weight (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*.

Ordering and relative weights are important. *O'Hara* scores 13.5; this is meaningless except in the context of a search. Here, 13.5 is good, twice the score of the next candidate. However, the user may be suspicious of the ordering of *Tennison*, who scores 2.0, and *Motta*, who scores 1.5. These figures have no absolute interpretation (except in terms of the algorithm); it is therefore for the users to interpret them according to their own understanding of the structure of their CoP.

Weights can be created based on frequency or manually assigned. In this example, the weights were calculated automatically, with the most frequently used relation getting weight 1; those not used getting 0; anything in between is allocated accordingly. 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. It then applies a breadth-first search, following the relations, transferring the weights of the relation and instance to the next nodes. It continues until time out from the start node. Instances accumulate weights according to the numbers of relations from the starting node; the longer the path, the smaller the weight transferred. The weightier the relation, the larger the weight transferred. Therefore a short distance, or a significant connection, with the base instance will tend to push an instance up the batting order. In this example, since *O'Hara* has written many papers with *Shadbolt*—many individual relations with a heavy weighted node—it has increased *O'Hara's* score, and other nodes connected to it.

THE ACTOR, DATA, AND PROCESS-ORIENTED (ADP) APPROACH

To support today's knowledge economy, an appropriately designed OM must closely support organizational operations and its business aims. To address these needs, we combine the use of the ONA method with methods that capture and analyze two other important aspects of an organization—the human and operational aspects.

In our approach, we examine an organizational context in three different dimensions: *the data* that the (virtual) organization operates upon, *the actors* that operate within the organization, and *the processes* that the organization carries out. These three dimensions are the cornerstones of any organization and are closely interconnected with each other. We show how these important aspects of an organization can be used seamlessly in different modeling and analytical methods to help support an OM.

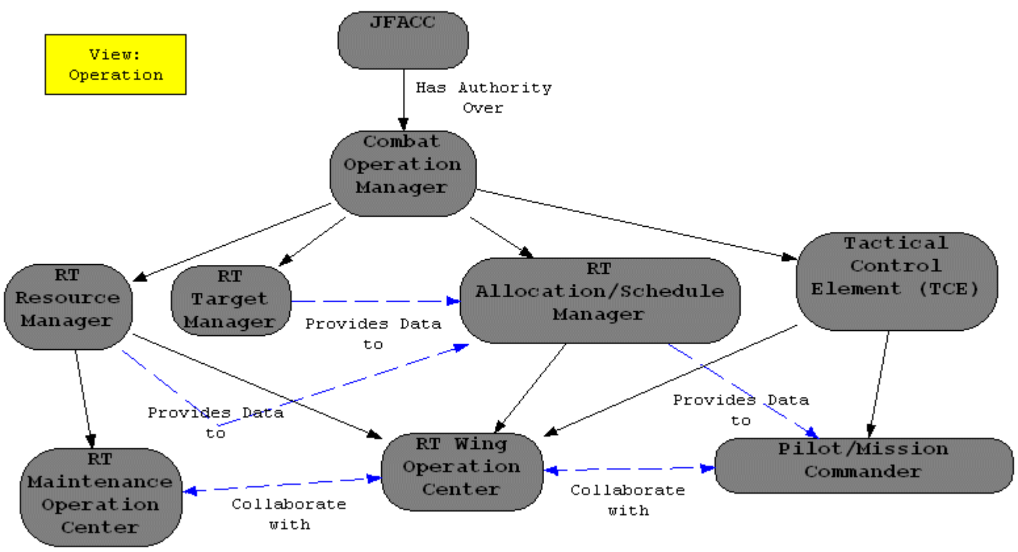
The ONA approach we presented in the previous section dealt mostly with *the data* aspect. In the following two sections we will introduce the other two aspects: *actor* and *process.* First, we describe a role-odeling method that suitably captures the *actor* aspect of a (virtual) organization and then a rich business process modeling method that captures the *process* aspect of an organization.

A ROLE-AWARE SUPPORT FOR OM

RACD Role Modeling as part of role activity and communication diagram (RACD) (Chen-Burger et al., 2000) was firstly introduced and used to capture U.S. Air Force operations, and roles of their personnel in connection with their operations in U.S. DARPA-funded Air Operations Enterprise Modeling (AOEM) project. Similar role modeling methods are organizational charts that are commonly used to illustrate organizational structures. Such methods, however, are typically informal that does not support formal reasoning tasks. They also do not capture sufficient information for KM tasks. RACD Role Models are formal descriptions that describe KM-related data: the different types of roles and relationships between them are formally defined in an underlying ontology. A Role Model depicts roles that different personnel may play within one or more organizations while interacting with other roles. It also indicates the formal, informal, and operational relationships between the different types of roles. Figure 3 illustrates an example role model that depicts personnel's roles in U.S. Air Force Operations. Typically, such roles span across different organizations.

This role model enables one to describe the typical organizational *hierarchical relationship* between roles, such as "has authority over." It also enables one to capture *functional relationships* such as "provides data to" and "collaborates with." Broadly speaking, there are two types of influence relationships between different roles: formal and informal (Schreiber et al., 1999). Formal influences are explicitly described in an organizational context, such as "has authority over,"

Figure 3. A high-level RACD role model that depicts the roles personnel play in U.S. Air operations that span across different organizations (a screen capture of KBST-EM)



"audit" or "give advice to." Informal influences, on the other hand, are not explicitly described, as some roles support other roles in their tasks that they have implicit influence over them. For example, the "supports" relationships between secretaries and their bosses and colleagues are informal influences.

Hierarchical relationships (denoted in solid links) normally have a direct correspondence with organizational charts. *Functional relationships* (denoted in dashed links) describe the functional roles that a role plays while interacting with others. Functional relationships also give detailed insights into how the different roles relate with, support, command, monitor, and/or constraint each other. This is invaluable for KM tasks, as it captures knowledge flows and the functions of these flows. For instance, if a KM task is to assess how a certain knowledge item was used, one can relate this knowledge item to its provider and then by following the different role-relationships, one can discover how the knowledge item may be used by the different knowledge users.

In RACD models, two types of roles are described: *abstract* and *concrete* roles. *Abstract roles* are performed by a collective group of actors such as an organization or its subdivisions. *Concrete roles* can be mapped to an individual actor (whether that is a human or a piece of software). An abstract role can be decomposed to more detailed ones. For instance, Figure 3 provides a higher-level view on personnel roles and their relations. However, these abstract roles may consist of smaller ones: for example, "RT (Real-Time) Wing Operation Center" may consist of several smaller and more detailed roles that support each other. The ability of being able to compose and decompose roles enables one to gain a concise view of organizational structures—which is invaluable, especially in the context of a virtual organization where roles, their functions and interactions between them are complex. It also allows one to gain a detailed understanding of responsibilities of individual actors and how they support each other given certain tasks. By doing so, one gains in-depth comprehension of an organization and may thus improve organizational efficiency. In addition, such organizational role modeling methods may be used to provide a direct input when capturing organizational processes, which will be discussed in the following session.

A RICH PROCESS SUPPORT FOR OM

Process models are commonly used to describe and analyse an organization's operations. Popular process models are IDEF3, UML's activity diagram and Petri Net. When a process model is developed with an organization's context in mind, a process model can be used instrumentally to achieve organizational goals—an aim for methods such as BPR and BPI. When used with a close integration and good understanding of the actor and data aspects of an organization, a process model can act as an integrated part of an OM life cycle. Fundamental business process modeling language (FBPML) is equipped to meet with such requirements (Chen-Burger & Stader, 2003). It is described in a rich three-layered objective-process-application modeling framework that is fully aware of an organization's environment. It is suitable to be used in business contexts, but is also applicable to other more generic process modeling needs. FBPML is goal-directed. That is to say that those corresponding long- and short-term business objectives are explicitly encoded in their processes and business rules are closely linked to these processes.

In our proposed ADP-based approach, the process modeling method acts as a glue to interact with the *actor* and *data* aspects (the ONA method described before deals with the data aspect) within an organizational context. FBPML is ontology based, which means that each data item that a process manipulates is defined in an ontology. It also supplies a formal data language, FBPML-DL, which describes the domain concepts (including instances, classes, and axioms) that processes operate upon. The formal process representation of FBPML, FBPML-PL (process language) takes in FBPML-DL constructs as part of its description and provides them to the *Workflow Engine* for interpretation and execution. Figure 4 provides a conceptual overview of how a FBPML workflow engine works in practice. This figure shows how a user can directly conduct the workflow engine's behaviors by providing initial process descriptions. It also shows how a user can create workflow system behaviors in real-time and in a flexible manner by dynamically interact with the workflow engine. This ability consequently enables us to carry out more flexible and adaptive KM processes later on.

THE WORKFLOW ENGINE

The workflow engine has two components: a *process manager* for handling the execution of the workflow and a *meta-interpreter* for reading and understanding the descriptions of processes and data. Equipped with an appropriate workflow algorithm, the workflow engine periodically retrieves new events that occur dynamically and identifies processes that have been specified in the process model which are relevant to these events. It examines the truth value of the triggers of each of those retrieved processes. It then creates a process instance for each of those processes and put it in the *Process Agenda,* that is, if all of the corresponding triggers are found to be true.

The workflow engine also looks for discrepancies between process instances in the Process Agenda. One example conflict is when one process wishes to delete data while another needs it (as its preconditions) for its execution. In this case, individually, each process will have its triggers and preconditions satisfied prior to execution. However, when examined together, their execution goals conflict with each other. Once culprit processes are found, the conflict is explained and resolution suggestions are given to the user (Chen-Burger & Robertson, 2005).

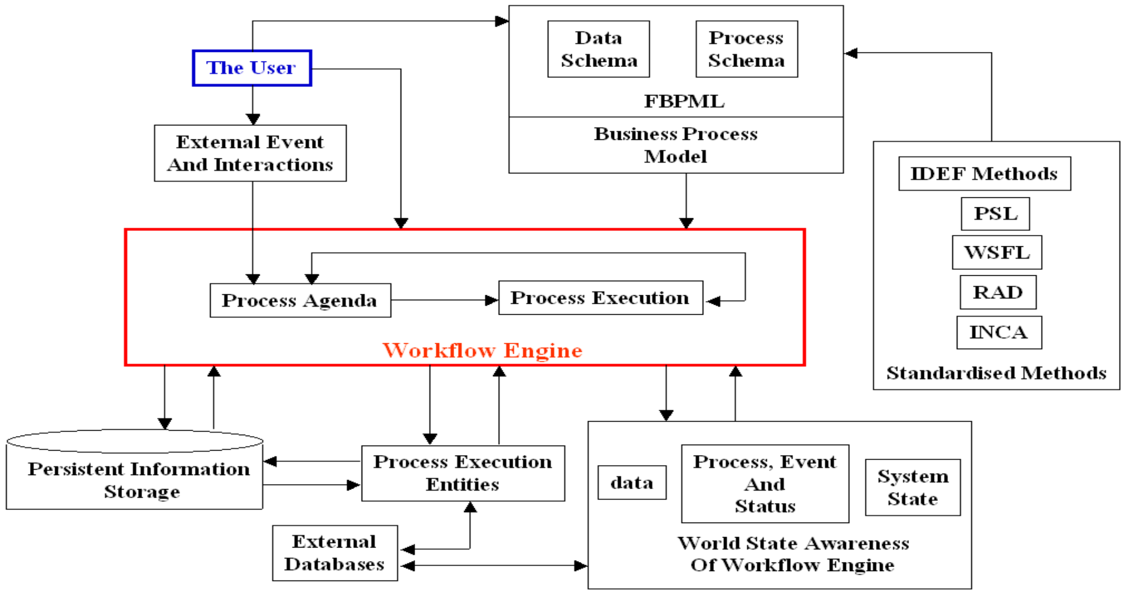
The Process Agenda stores a list of all process instances that are waiting to be executed. However, process instances that are in conflict with other instances are reported to the user and left in the agenda until the conflicts are resolved. For this, a time-out mechanism has been put in place to prevent indefinite waits in the agenda, thus also preventing the agenda to store expired/irrelevant old process instances indefinitely. Once a list of "clear" process instances are ready to be executed, they are added to the Process Execution queue and are executed instantly.

USING PROCESS MODEL FOR ADP-BASED KM ANALYSIS

From the simplified overview depicted in Figure 4, one gets an insight into how the FBPML workflow engine works and also the fact that it takes at least two elements as main inputs: the data and process descriptions. This is where an ontological method such as ONA can tap into a process modeling method and make a direct influence into how processes may be carried out—which produces significant impact in KM tasks. For example, in the scenario mentioned before, where interesting knowledge items, for example, certain instances in a populated ontology, have been identified via the ONA powered algorithm. The user is interested to find out information about these knowledge items, in particular, regarding their relations with organizational processes. Example queries in this scenario therefore are: "Who has created these knowledge items?," "What process has created them?," "When are they being modified?," "How are they being used?," "Who are using them?," "Where are they being stored," "How are they being stored," "What are the frequencies that those knowledge items are being used and in what context," "How critical are those knowledge items—for example, to which task and to whom?" and, ultimately, "What are the impacts of those knowledge items to the organization?" A carefully combined actor, data/ontological, and process based approach can provide good approximate answers to most of these questions with minimum effort required.

For example, our proposed ADP approach will work as follows: as FBPML is embedded with a formal description of a data language, interesting knowledge items may be formulated using FBPML-DL. These will have been identified with ONA, and thus will already be in a formal representation format.

Figure 4. A conceptual overview of the FBPML workflow engine



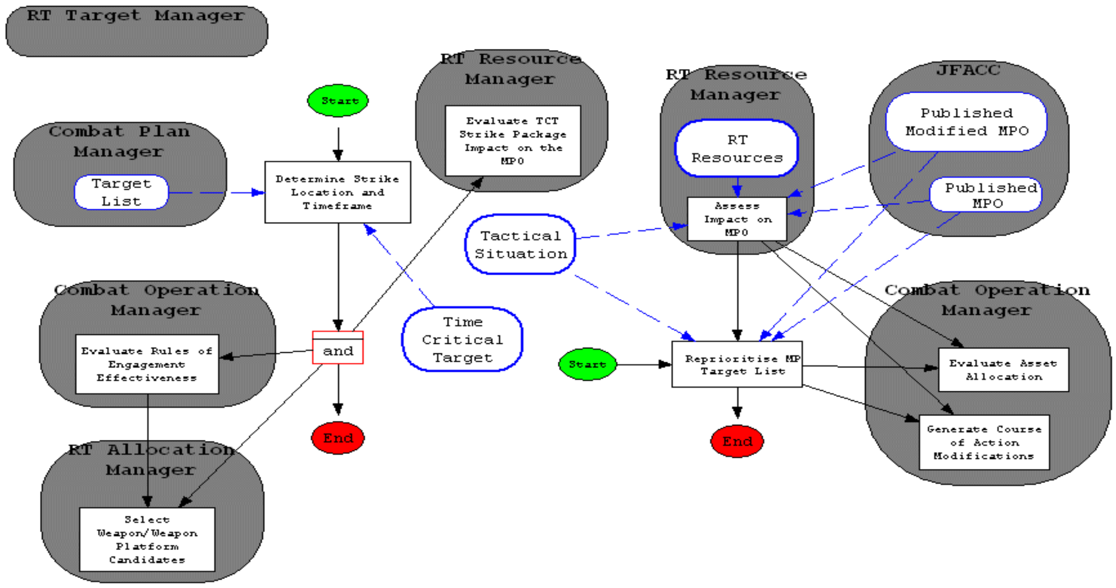
A FBPML model will therefore take such FBPML-DL constructions as part of its process description that is used as a basis for searching. For instance, based on FBPML-DL constructs, typical automated actions, such as *Create*, *Update*, *Monitor*, *Query*, and so forth, are formulated. Therefore, one can perform a relatively easy pattern-matching algorithm on the different process descriptions to work out the processes that generate, use, refer to, and audit those knowledge items. In addition, it is common practice in process modeling methods that relevant business analysis are carried out, such as identification of *critical processes* in an organization and the frequencies of a process. One may therefore derive approximate answers for such knowledge items based on information that he or she already knows about the processes that operate upon them. For instance, for a knowledge item/piece of information that is the main or only input for a critical process, he or she may derive that this piece of knowledge or information is also of critical importance. Another example is when a knowledge item or a piece of information is only used (e.g., referred to) by very few and low-frequency processes, it is straightforward to derive that this knowledge item/information is not used frequently.

In this way, we can now infer new knowledge about the data base upon existing knowledge about processes that is of minimum effort. In addition, as FBPML allows its users to define new process constructs. To identify such novel processes, we need to search for the relevant FBPML-DL constructs within all FBPML process descriptions. However, to understand the semantics of such process components, we will need to look into the description and definitions of its underlying computational module.

ACTOR-RELATED QUERIES

We have so far answered the above proposed process-related queries. Some of the above queries, however, are relevant to the "who" questions and their answers

Figure 5. An FBPML process model for U.S. Air operations that is across organizations (a screen capture of KBST-EM)

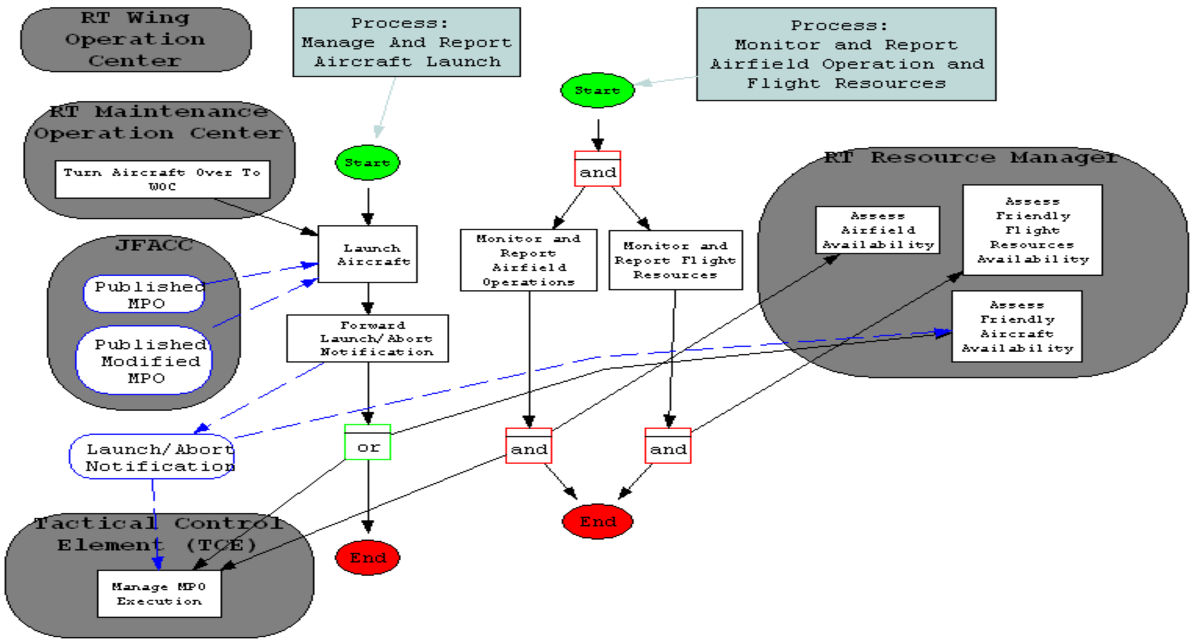


are not provided yet. To answer these "who" questions, we need to ask how the RACD role models fit with the FBPML model, so that we can provide suitable answers with it. FBPML processes are grouped and describes in terms of actor roles. Each process is labelled with the corresponding "actor" that carries out the task. In this way, it is possible to see all of the processes that an actor carries out. It is also easy to see how the different actors collaborate with each other through sharing a larger process model.

Figure 5 shows an example FBPML process model (that is a screen capture of KBST-EM) for the same domain of U.S. Air Force operations. This figure shows the two operations of the RT (real-time) Target Manager. These are the two operations (indicated in squared boxes) that are outside of any rounded outer squares. However, in the same diagram it also encompasses different roles that other personnel play (indicated in the rounded outer squares) and their corresponding processes (indicated in squares) that they perform. The links between the different processes indicate the directional control and data flows between them. Note that this diagram also indicates the data types that a role stores (denoted in small rounded boxes). Figure 6 gives another example model that describes the two main processes of RT Wing Operation Center: launch and monitor aircrafts (indicated by their headings "Process:"). While interacting with other roles, note the document MPO (published and modified) are used in operations in the two provided process models and by several personnel as input of their activities. In conjunction with process knowledge, we can now answer most of the above "who" questions.

By seeking out the relevant processing components in a process model, we could now identify the actors who carried out these tasks. For example, if it

Figure 6. A FBPML process model for RT wing operation center (a screen capture of KBST-EM)



is a "creation" type of task that the actor performs, then this actor is the one who has created the knowledge/information item in the data store. Similarly, if it is a "reference" type of task, we may say that the corresponding actor is using that piece of information or knowledge as a part of their work. If it is the same actor who creates, updates, uses, and monitors the same information, one may say that this is the main actor that creates, maintains, and uses that piece of information or knowledge. In this way, one can get good quality initial answers.

Use of the combined ADP formal approach requires minimum additional effort and it is reliable for as long as the domain knowledge captured is as accurate and complete as possible. However, this approach is not entirely infallible. One possible problem resides in the fact that informal processes are often not recorded in a formal (business) process model. In the example of the creation type of processes above, it is possible that they may be performed by separate key-in personnel and not by the knowledge creators themselves. However, in this case, one still gets the first line of defence—it helps to identify the first person to talk to in order to find out who is the original knowledge creator (a piece of information that may be indicated in the electronic or paper-based record that is not part of the formal system).

SOCIAL AND MANAGERIAL IMPLICATIONS

The KM approach proposed is an incremental one. This separates it from other more radical approaches where much existing practices, including organizational structures, are subject to changes. Such methods are therefore more likely to attract staff resistances. Our approach, instead, re-uses existing knowledge and tools to derive new knowledge. In this way, less resistance should be met. Due to less disruption in existing work, it is also easier for the staff to contribute towards KM projects.

In addition, due to selective reuse, benefits for the KM project can be quicker realized when compared with another project that needs to create information from ground level. This will be an attractive feature for the management. When business value creation is a direct result of deploying KM processes and appropriate rewards are offered, it provides strong incentives for all personnel involved, which in turn boosts the success rate of the project.

One issue of KM is human resources management—to make sure that knowledge is distributed appropriately and stable within the organization. One way to do this is to map the actual personnel to the RACD role model for knowledge distribution analysis. When it is identified that a person performs several roles in an organization, the information may become saturated. It is therefore important to identify such information overload and capture critical knowledge through KM processes, thereby reserve/distribute important knowledge within the organization.

FUTURE TRENDS AND CONCLUSION

In this Knowledge Era we are living in, an organization's economical growth heavily depends upon the wealth of its knowledge and how well it taps into it. It is therefore critical that KM tasks are carried out efficiently and effectively to its advantages, especially when a large OM is present. Perfect KM solutions that entirely unlock knowledge from information, however, are not available yet. The increasingly high demand for immediate usable knowledge stemmed from information explosion will continue to inspire new and specialized KM related technologies to be included in systems of a variety types of applications for many years to come.

To address these demands, our combined ADP analytical and inference framework provides rich support for KM tasks in the context of OM. Its main advantages are to make use of existing reliable methods and their known properties, thereby minimizing additional effort for KM tasks so to elicit maximum benefits for OM queries. Based on the ADP method, good quality approximate answers can be derived with minimal effort when compared with another approach where brand new answers must be sought and compiled from raw data.

Furthermore, the ONA-based OM architecture we proposed makes it possible to analyze and propose content for the initial seeding of an OM. This is a powerful incentive and tool for OM engineers, as they can effectively tackle the cold-start syndrome that haunts most of these systems in their initial set up. The ONA-based approach coupled with the modeling flexibility of an ADP approach provides an interesting and holistic ontology-based business process support geared towards comprehensive OM for distributed enterprises.

However, these approaches are not entirely infallible, as not all organizational aspects can be captured explicitly. This is a common challenge when trying to provide a complete set of KM and OM support. When facing the trade-offs between utilizing knowledge for gains and bearing the cost of capturing and maintaining it at the first place, a balance is often struck. To compensate for the information gap caused by informality, one must employ common sense and domain specific knowledge when searching for the true answers to queries. Another useful approach to combat missing information is to employ iterative and adaptive KM life cycles, thus improve the underlying three ADP models based on query demands. Hence, the quality of KM and answers to queries can be improved incrementally through time.

ACKNOWLEDGMENT

This work is supported under the Advanced Knowledge Technologies (AKT IRC) project, UK EPSRC

GR/N15764/01. 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

Abecker, A., Bernardi, A., Hinkelmann, K., Kuhn, O., & Sintek, M. (1998). Toward a technology for organizational memories. *IEEE Intelligent Systems, 13*(3), 40-48.

Alani, H., Kalfoglou, Y., O'Hara, K., & Shadbolt, N. (2002). Initiating organizational memories using ontology-based network analysis as a bootstrapping tool. *BCS-SGAI Expert Update, 5*(3), 43-46.

Breuker, J., & Van de Velde,W. (1994). *The commonKADS library for expertise modeling.* Netherlands: IOS Press.

Carlsson, C., & Turban, E. (2002). DSS: Directions for the next decade. *Decision Support Systems, 33,* 105-110.

Chen-Burger, Y-H., & Robertson., D. (2005). *Automating business modeling: A guide to using logic to represent informal methods and support reasoning.* Book Series of Advanced Information and Knowledge Processing. Springer.

Chen-Burger, Y-H., Robertson, D., & Stader, Y. (2000). *A modeling support framework for enterprise knowledge capturing, sharing and reusing.* INAP2000 book proceedings.

Chen-Burger, Y-H., & Stader, Y. (2003). *Formal support for adaptive workflow systems in a distributed environment.* In L. Fischer (Ed.), *Workflow handbook.* Future Strategies Inc.

Cohen, P., & Kjeldsen, R. (1987). Information retrieval by constraint spreading activation in semantic networks. *Information processing and management, 23*(4), 255-268.

Cooper, W.S. (1997). *On selecting a measure of retrieval effectiveness.* In Readings in Information Retrieval.

Dieng, R., Corby, O., Giboin, A., & Ribiere, M. (1999). Methods and tools for corporate knowledge management. *International Journal of Human-Computer Studies (IJHCS), 51,* 567-598.

Grosso, W., Eriksson, H., Fergeson, R., Gennari, J., Tu, S., & Musen, M. (1999). Knowledge modeling at the millennium - the design and evolution of Protege2000. In *Proceedings of the 12th Knowledge Acquisition, Modeling, and Management (KAW'99*). Banff, Canada.

O'Hara, K., Alani, H., & Shadbolt, N. (2002). Identifying communities of practice: Analyzing ontologies as networks to support community recognition. In *Proceedings of the 2002 IFIP World Computer Congress,* Montreal, Canada.

Paice, C.D. (1991). A thesaural model of information retrieval. *Information processing and management, 27*(5), 433-447.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on systems, management, and cybernetics, 19*(1), 17-30.

Schreiber, G., de Hoog, R., Akkermans, H., Anjewierden, A., Shadbolt, N., Van de Velde, W. et al. (Eds.). (1999). *Knowledge engineering and management: The commonKADS methodology.* MIT Press.

Wenger, E. (1999). Communities of practice: The key to knowledge strategy. In E.L. Lesser, M.A. Fontaine & J.A. Slusher (Eds.), *Knowledge and Communities* (pp. 3-20). Butterworth-Heinemann.

Zave, P., & Jackson, M. (1997). Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology, 6*(1), 1-30.

ENDNOTE

[1] 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.