Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2007 Proceedings

European Conference on Information Systems (ECIS)

2007

Enterprise Engineering Using Semantic Technologies

Y. Chen-Burger
The University of Edinburgh, jessicac@inf.ed.ac.uk

Y. Kalfoglou
University of Southampton, ykalfoglou@ecs.soton.ac.uk

Follow this and additional works at: http://aisel.aisnet.org/ecis2007

Recommended Citation

Chen-Burger, Y. and Kalfoglou, Y., "Enterprise Engineering Using Semantic Technologies" (2007). *ECIS 2007 Proceedings*. 134.
http://aisel.aisnet.org/ecis2007/134

This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2007 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

ENTERPRISE ENGINEERING USING SEMANTIC TECHNOLOGIES

Yun-Heh Chen-Burger, Informatics, The University of Edinburgh, Appleton Tower, Crichton Street, Edinburgh EH8 9LE, UK, jessicac@inf.ed.ac.uk

Yannis Kalfoglou, Advanced Knowledge Technologies, School of Electronics and Computer Science, University of Southampton SO17 1BJ,UK, ykalfoglou@ecs.soton.ac.uk

Abstract

Modern Enterprises are facing unprecedented challenges in every aspect of their businesses: from marketing research, invention of products, prototyping, production, sales to billing. Innovation is the key to enhancing enterprise performances and knowledge is the main driving force in creating innovation. The identification and effective management of valuable knowledge, however, remains an illusive topic. Knowledge management (KM) techniques, such as enterprise process modelling, have long been recognised for their value and practiced as part of normal business. There are plentiful of KM techniques. However, what is still lacking is a holistic KM approach that enables one to fully connect KM efforts with existing business knowledge and practices already in IT systems, such as organisational memories. To address this problem, we present an integrated three-dimensional KM approach that supports innovative semantics technologies. Its automated formal methods allow us to tap into modern business practices and capitalise on existing knowledge. It closes the knowledge management cycle with user feedback loops. Since we are making use of reliable existing knowledge and methods, new knowledge can be extracted with less effort comparing with another method where new information has to be created from scratch.

Keywords: Role Modelling, Enterprise Process Modelling, Knowledge Management, Organisational Memories, Ontology Network Analysis, Semantic Web.

1 INTRODUCTION

The era we are living in is characterised by an unprecedented explosion of information that is digitized and available to large audiences through online open-ended environments. Organisations and enterprises had to quickly adopt to this new era: software applications, databases, and expert systems which were designed and constructed by a dedicated group of software and knowledge engineers who had control of the entire lifecycle of IT artefacts, seems to be old practice. Software engineering praxis is shifting from the custom-made, stand-alone systems to component-based software engineering (e.g. Commercial off-the-shelf, Enterprise Resources Planning systems); databases are gradually deployed in distributed architectures and subsequently federated; knowledge-based systems are built by reusing existing knowledge bases and inference engines. Moreover, the distributed nature of IT systems has experienced a dramatic explosion with the generalised use of the Internet. The World Wide Web (WWW), and its ambitious extension, the Semantic Web (SW), has brought an unprecedented global distribution of information, e.g. in the form of hypertext documents, online databases, open-source code, terminological repositories (e.g., WordNet, Miller 1990), semantically annotated documents (with mark-up formats like XML, RDF and OWL), Web services, which continually challenge the traditional roles of IT in our society.

One promise for IT Systems architects is to use Knowledge Management (KM) based methods and tools to cope with an ever expanding nature of distributed systems in a global scale. At the cornerstone of most of these tools lies the advocated buzzword of *semantics*. 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 SOA (Service Oriented Architecture) based systems. This broadness brings together the works of many communities and disciplines from the academic and industrial realm with a common goal: to inject descriptions of objects, theories, processes, and associations between components in a distributed system with semantics (meaning) that will enable automated interoperability between processes and services.

The successful blending of the newly emerged (or branded) semantic technologies with traditional KM systems starts from a fundamental principled part of any business: *process*. Starting from around ten years ago, the benefits of process-oriented approaches, such as BPR (Business Process Re-Engineering) and BPI (Business Process Improvement), for achieving radical enterprise changes and improvements was well recognised. Today it is seen as one of the fundamental steps that one goes through in order to improve enterprise performance. In this line of thinking, processes are seen as tangible entities that can be formally captured, analysed, and incrementally or radically modified in order to change enterprise behaviours so can achieve a certain set of enterprise objectives. In the context of KM, such approaches are also much appreciated (Schreiber et al., 1999).

Despite the abundance of KM supported organisational and enterprise engineering reported in the literature, we observe a dearth of approaches that tackle fundamental aspects of enterprise-wise contexts in: *data*, *actors* and *processes* in a holistic manner (PAKM, 2006)(Chen-Burger, 2005). These three aspects are arguably the cornerstones of any enterprise-wise context. Each represents a different dimension: *data* refers to the content within which the (virtual) enterprise operates; *actors* refer to the human or artificial entities (e.g. software) that operate in an enterprise; *processes* refer to the operations that the enterprise carries out. Having all these three key components inter-connected is important towards interoperability in modern enterprises. In this paper we propose a novel framework that makes use of (formal and semi-formal) modelling methods that represent and analyse the three dimensions of *data*, *actors* and *processes* of an enterprise in the context of organisational memories (OM). Moreover, it enables us to enable vital knowledge to be shared in different parts of an organization, thus deriving vital information that is previously not known to the enterprise.

What separates our approach from conventional BPR or BPI is the fact that we have employed modelling methods that are underpinned by formal methods that support automated inference. By doing so, it enables us to propagate knowledge already known in one area of the OM automatically to another area with only very little cost. It also allows us to combine knowledge learned from different areas of an organisation in order to generate new knowledge, thus assist KM activities. As such KM activities can also be described using business process modeling techniques, its results are then fed back to the sharing pool of knowledge, thus further enhancing our knowledge sharing and integration efforts.

Since most large OM would already have partial information assets described using some types of representations, e.g. ontologies, data models, or business process models; when this information is used in combination with the three dimensional approach we describe in the next section, rich reasoning knowledge is generated. Such vast and potentially distributed knowledge base can be described using SW languages, which could also be shared via the SW, and the processes implemented via a SOA (Service Oriented Architecture), when appropriate. This is also well suited to deploy an *Ontology Network Analysis (ONA)* algorithm, which we present in subsection 2.3, as it supports user-tailored queries. Over time, as users (that may be geographically distributed) use the knowledge bases, gaps in the knowledge are identified and KM processes may be created to collect such knowledge in order to assist future similar enquiries. This enables us to close the loop of knowledge acquisition and their uses.

2 A Brief description of the ADP approach

"The body of knowledge can be viewed as a piece of large diamond that can be cut in many different facets and can therefore be appreciated from many different angles." [1]

This is also the philosophy behind the Actor-Data-Process (ADP) support framework. The ADP framework consists of a set of Enterprise Modelling methods: *Role Activity and Communication Diagram (RACD)* (Actor aspect), *FBPML (Fundamental Business Process Modelling Language)* (Process aspect) and *Ontology Network Analysis (ONA)* (Data aspect). Its aim is to provide a rich, holistic KM support for an enterprise with minimised additional KM effort required. It uses these structural conceptual modelling methods to capture, describe, reason and make use of knowledge. It has three different cuttings: *Actor, Data* and *Process*. We have chosen these three aspects because they are fundamental to understand an enterprise's context. They are inter-related in an enterprise: enterprise *processes* operate (based) on *data*; *actors* (human and software agents) carry out *processes* to achieve objectives of their *roles* in the enterprise; *data* is formally defined in ontologies that give definitions and relationships between them. Data constraints are primary restrictions to an enterprise that processes and actors must obey. Such shared characteristics between the different aspects allow us to share information and detect inconsistencies existing in the different parts of an enterprise. This KM effort is augmented by applying ontology based analytical and querying methods, ONA, that allows us to apply knowledge in suitable business areas. This also identifies potential gaps in the knowledge and enables us to acquire and feed new knowledge back to the enterprise.

Example queries that the ADP framework may help answer are: *"Who has created these knowledge items?", "What process has created them?", "How are they being used?" and "Who are using them?", "Where are they being stored", "How are they being stored" and "What are the frequencies that those knowledge items are being used and in what context", "How critical are those knowledge items?" and ultimately "What are the impacts of those knowledge items to the enterprise?"*. A carefully combined ADP approach can provide good approximate answers to most of these questions. We describe how this may be achieved in the following sections. We first introduce the role-based modelling method, which is followed by the other "process" and data" based methods.

2.1 The Role-aware Enterprise Modelling

For any enterprise, people play the most central role. They are the ones that drive the enterprise. They actively take roles in enterprises, carry out work, create knowledge in their roles and work to accomplish goals. They also share their knowledge with others, thereby augmenting their abilities through cooperation and support thus realise their visions. To capture this important human factor of an enterprise, role modelling methods are used. *RACD Role Modelling* is one of such role modelling methods. It is part of the *Role Activity and Communication Diagram (RACD)* (Chen-Burger et. al., 2000) and was firstly introduced and used to capture US Air Force operations and the roles of their personnel in connection with air operations. RACD Role Models are ontologically based, which means the semantics of roles and the relationships between them are described in an underlying ontology. A role model depicts roles that a person plays in one or more organisations while interacting with other roles. Role capacities may therefore constraint the type of personnel that may play this role. Upon instantiation of the model, one could then associate actual personnel with roles. This will assist queries such as "Who plays the role X?"

The role model indicates the formal, informal and operational relationships between the different types of roles. Figure 1a is a screen capture of KBST-EM (Knowledge Based Support Tool for Enterprise

[1] This statement was firstly given by a senior IBM software engineer. The exact origin is unknown.

Modelling) (Chen-Burger, et al, 2003)[2]. It illustrates an example role model that depicts personnel's roles in US Air Force Operations. Typically, such roles span across different enterprises. This role model enables one to describe the typical enterprise-wise *hierarchical relationships* 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 enterprise-wise context, such as "has authority over", "audit" and "give advice to". *Informal influences*, on the other hand, are not explicitly described - as some roles support other roles in their tasks, they have implicit influences over them. For example, the "supportive" relationships between secretaries and their bosses and colleagues are informal influences.

Hierarchical relationships (denoted in black dash links) normally have a direct correspondence to an enterprise's organisational charts. Functional relationships describe the functional roles that each role plays while interacting with others. They give detailed insights into how the different roles relate to, support, command, monitor and/or constraint each other. This is invaluable to 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 directional role-relationships, one can discover how this knowledge may be used by other knowledge users. If the task also requires *contextual information about where* this knowledge item is used, the following process section gives this information.

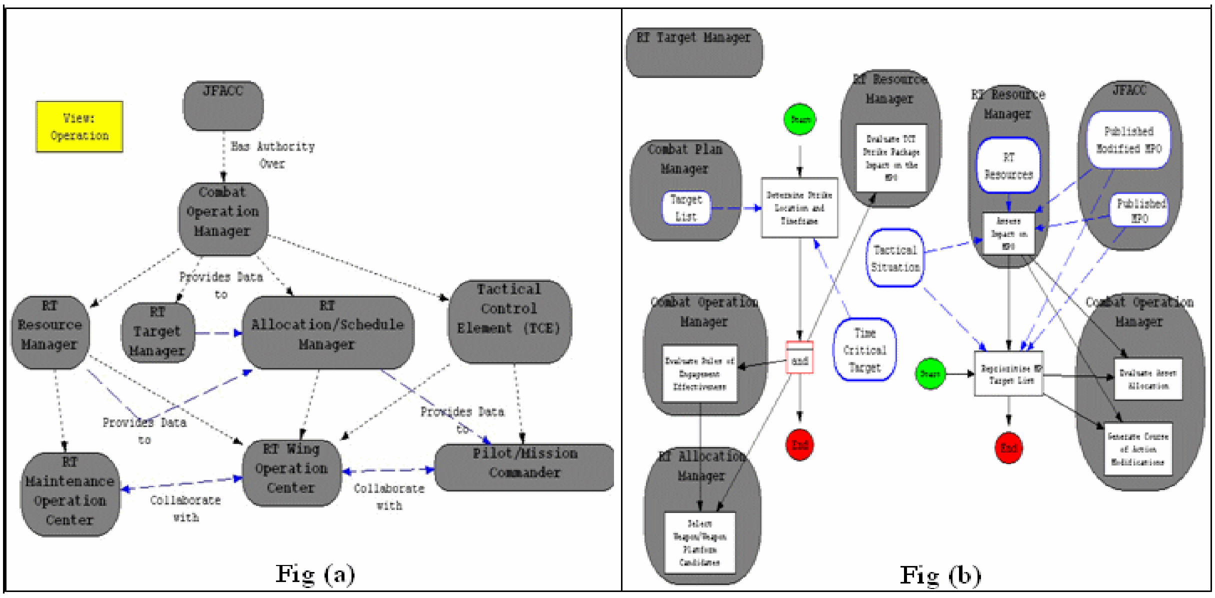

Figure 1: (a) A *RACD Role Model* that depicts the roles personnel play in US Air Operations that span across different enterprises; (b) A role-aware FBPML process model for US Air Force operations that is across different enterprises. (Fig 1a and 1b are screen captures of KBST-EM.)

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 enterprise or its subdivisions; whereas concrete roles can be mapped to an individual actor (that may be a human or a software program). An abstract role can be decomposed to more detailed ones. For instance, Figure 1a provides a higher-level view on personnel roles and their relations. However, these abstract roles may consist of smaller ones: "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

[2] Currently, KBST-EM houses 30 different modelling methods where common ontologies and knowledge can be shared and reasoned through an underlying partitioned knowledge base and inference engine.

concise view at different levels of an enterprise's structure – which is very useful, especially in the context of a virtual enterprise where roles and their functions and interactions are complex. It also allows one to gain an understanding of detailed functions of individual actors and how they interact with each other. By doing so, one gains in-depth comprehension of an enterprise and thus may improve enterprise-wise efficiency. In addition, such role modelling methods may be used to provide a direct input when capturing enterprise processes.

2.2 Rich Process Support

Process models describe an enterprise's operations. They can also be used as a basis for analysing and commanding of enterprise operations. Together with a close integration and good understanding of the *actor* and *data* aspects of an enterprise, a process model acts as an integrated part within an enterprise's IT front end to achieve enterprise goals – one such IT front end is, e.g., an OM which we will examine in the next section. *FBPML (Fundamental Business Process Modelling Language)* is one such process language that meets business requirements (Chen-Burger & Stader, 2003). It is described in a rich three-layered objectives-process-application modelling framework that is fully aware of an enterprise's context. FBPML is goal-directed. That is to say, that those corresponding long- and short-term business objectives are already explicitly encoded in their processes and business rules are closely linked to these processes. It can be exported to SW languages: BPEL4WS (Guo, 2005) and OWL-S (Nadarajan, 2006), thus is SOA compatible and suits modern distributed virtual enterprises.

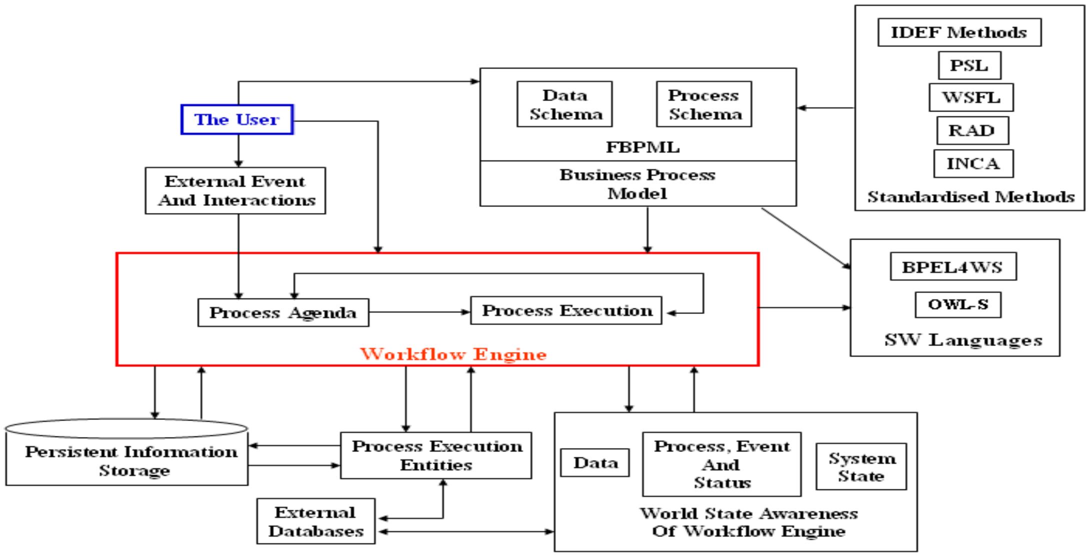

Figure 2: A conceptual overview of the FBPML Workflow Engine

In our proposed ADP-based approach, the process modelling method acts as a glue to interact with the *actor* and *data* aspects within an enterprise context. FBPML is ontology based, which means that each data item that a process manipulates is defined in ontologies – this is supportive to the concept of distributed ontologies within the Semantic Web. 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 2 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 behaviours by providing initial process descriptions. It also shows how users can create workflow system behaviours in real-time by dynamically interact with the workflow engine. This ability consequently enables us to carry out more flexible and adaptive KM processes.

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*, i.e. if all of the corresponding triggers are found to be true.

The workflow engine also looks for discrepancies in the process instances in the *Process Agenda*. When a discrepancy is found, it will be reported to the user together with advice of source of conflicts and possible resolutions for conflicts. 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 hold in the agenda, thus also preventing the agenda to store expired/irrelevant old process instances indefinitely. Once a list of clear process instances is ready to be executed, they are added to the *Process Execution* queue and executed instantly.

From the simplified overview depicted in Figure 2, one gets an insight into how the FBPML workflow engine works and also the fact that it takes at least two elements as main input: the data and process descriptions. We mentioned previously that it is important to know about the data that processes are manipulating, as they often impose constraints to processes. FBPML is embedded with a formal data language. When assisting user queries, such integrated data and process information provide a convenient basis of information sources. In our example user enquiries in Section 2, the interested knowledge items may be formulated using FBPML-DL. These will have been identified with the method we describe in the following section for ONA, and thus will already be in a formal representation format. 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*, etc, 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, as it is common practice in process modelling methods that relevant business analysis are carried out - such as identification of *critical processes* in an enterprise, 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. refer to) by very few and low-frequency processes, it is straightforward to derive that this knowledge item/information is not used frequently.

This way, we can infer new knowledge relatively easy and reliably based upon existing knowledge on processes, which does not require much additional effort. In addition, FBPML allows its users to define new process constructs. To identify knowledge items within 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 processes, we will need to look into the description and definitions of its underlying computational module.

We have so far answered the above proposed data and process-related queries. Some of the above queries, however, are relevant to the "who" questions and their answers 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 described in terms of *actor roles* – these are mapped to roles in the Role Model. Each process is labelled with the corresponding "actor" that carries out the task. 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. Upon linking actors with roles in the role model where actual personnel are provided, one can then identify relevant personnel (sometimes more than one) that are related to a particular query.

Figure 1b shows an example FBPML process model for the same domain of US Air Force operations. This figure shows the two operations of the RT (real-time) Target Manager. These are the two operations (indicated in black squared boxes) that are outside of any grey grounded square. However, in the same diagram it also encompasses different roles that other personnel play (indicated in grey rounded 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 blue rounded boxes). In conjunction with process knowledge, we could now answer most of the above "who" questions.

By seeking out the relevant processing components in a process model, we can now identify the actors who carried out these tasks. For example, if it is a "creation" type of tasks that the actor performs, then this actor is the one who has created the knowledge item/information in the data store. Similarly, if it is a "reference" type of task, we may say that the corresponding actor is using that 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 and maintains that piece of information or knowledge. In this way, one can get good quality initial answers.

As illustrated, using of the combined ADP formal approach requires less additional effort and because it is built upon existing tested methods, 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 knowledge and processes are often not recorded in a formal ontology or (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, even in this case, this approach still helps to identify the first person to talk to in order to find out who is the original knowledge creator or source documents[3].

2.3 Operate the ADP model in OM

Once we have the actor and process dimensions formalised and represented as described in the previous sections, we deal with the data. As mentioned previously, most enterprises would have collected some ADP-related information in one form or another. As data is fundamental to any business, we exemplify its uses in the ADP approach through a favourable KM technology that has been used in modern enterprises: OMs. They represent a breed of technologies that reflect a change in enterprise engineering, that is, the shift from data-oriented processing systems to more integrated with the human intellect and enterprise processes systems (Carlsson & Turban, 2002). OMs have been studied as a means for providing easy access and retrieval of relevant information to users. There are several technologies that support the implementation and deployment of OMs (Abecker et. al., 1998). Having an ideal OM in place could assist in decision making, which means that -- crudely speaking -- any information regarding the enterprise could be made easily accessible.

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: 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 analysed for relevance. Semi-automatic methods and techniques do exist, but these are

3 Information that may be indicated in the electronic or paper-based record that is not part of the formal system.

bound to individual technologies. Importantly, it is always the user who has to initiate search in the OM. But this requires the user to formulate a query, sometimes with the help of semi-automatic support, and then the OM system must parse the query successfully in order to retrieve information deemed to be relevant according to some pre-defined notion of relevance, and present it back to the user. This is time consuming and prone to error, as field surveys (Dieng et. al., 1999) as well as implemented systems (Abecker et. al., 2000) reveal. We are not only concerned with the quality and elicitation of resources that will be presented to OM users or the difficulties in engaging them in the technical task of formulating the right query, but we are also concerned with the nature of what these resources could be: (a) used by other systems within the enterprise, which incidentally also serve users in their quest for valuable information, (b) `unspecified', in that they are vaguely expressed, and need to be composed by a number of related resources or are external to the enterprise. More importantly, once these resources have been 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 enterprise that users used to use before confronted with an OM. This has been identified as the “cold start” syndrome (Gresse et.al. 2001) where the author reports that there were 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 an 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, which led to an increase in the OM's knowledge assets and consequently to increased access figures.

A way of tackling the 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 (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. (Zave & Jackson, 1997) report in their survey, vague and imprecise requirements are always difficult to formalise and subsequently convert to specifications, in the early phases of software development[4]. 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 their requirements. The vagueness and incompleteness of requirements from prospective OM users led some designers to build their OM around an existing workflow process engine, as for example in the *KnowMore* OM (Abecker et al, 2000). But this requires familiarisation and existence of a robust workflow process in the first place, and intensive modelling to link the two systems together.

Our approach to this problem is to use ontologies. These are formalised conceptualisations of the most prominent objects and entities in a business environment that shared across a variety of stakeholders. They are various forms and formats of ontologies (Chandrasekaran et.al., 1999) and their uses in KM have been around for few years (O’Leary, 1998). We assume that (a) ontologies will be available in the enterprise 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 enterprise settings nowadays, in the form of database repositories, SW data formatted in RDF/RDFS, and OWL ontologies.

Our hypothesis is that since we already have ontologies in the work place and some OMs are also based on ontologies, we could use them in other ways. ONA (Alani et. al., 2002) is the technique of applying information network analysis methods to a populated ontology in order to uncover certain

4 In our case, the early phase of developing an OM.

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 a domain ontology. There are many methods of studying networks, and of course many types of networks that can be studied (see, for example, O'Hara et. al., 2002). However, the advantage of studying ontologies is that the relations therein have semantics or types, and therefore 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 (Alani et. al., 2002).

ONA methods can be harnessed to address the resources selection problem in building OMs by using populated ontologies 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 semantically annotated 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 enterprise's ontology will not be spotted by the methods involved, especially if the ontology is in some way incomplete, and fails to cover the subject domain fully in some important aspect. 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 we used applies 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 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)[5]. 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:

ü 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.

ü 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.

5 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..

- 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.

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).

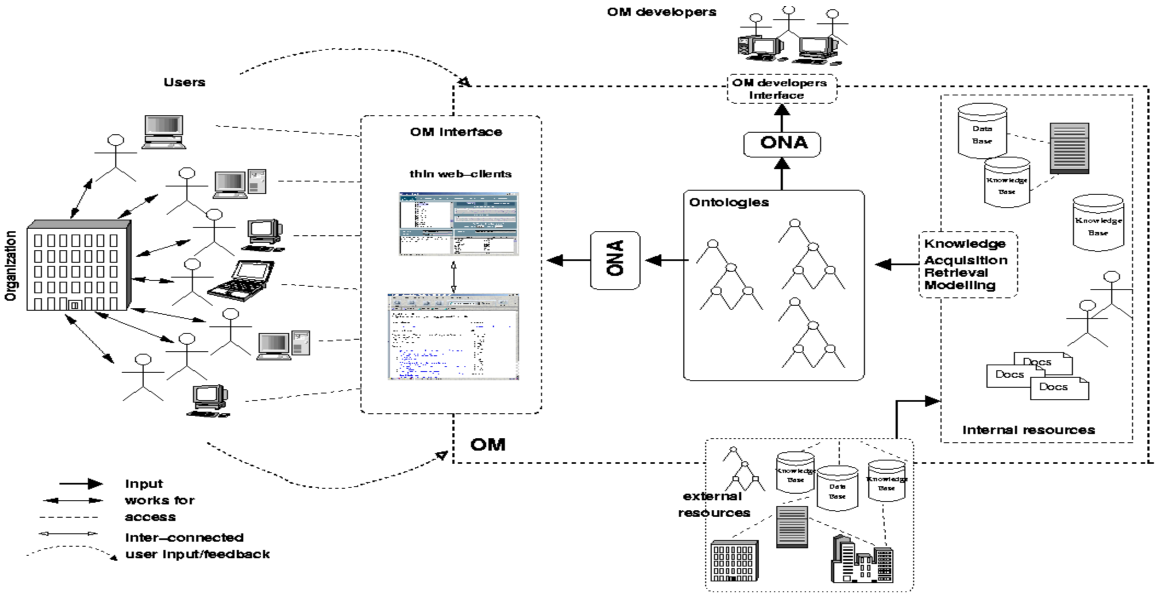

Figure 3: Supporting initial seeding of an OM: pushing knowledge to the OM as well as pulling it out - using ONA techniques.

In Figure 3 we depict a high-level diagram of an OM. This is not meant to be a reference architecture for OMs. 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 enterprise 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 enterprise'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 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 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 enterprise's ontologies in order to identify which concepts should be presented to certain types of users. This is where the ADP approach with the use of the FBPML model comes together.

The method described above is neither infallible nor adaptable to any existing OM setting. We identified potential caveats on using ONA to bootstrap OMs and categorise them in three broadly defined 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 many ontologies as are likely to exist in large enterprises.

The ONA-based solution we presented above targets the problem of setting up a comprehensive OM in a bid to attract high rates of access from its potential users. Our approach is based on the idea of an enterprise ontology underpinning the OM; however, it is likely that there will be more than one ontology in place and sometimes we need to resort to ontology mapping (Kalfoglou & Schorlemmer, 2003a) to provide solutions in this space (Kalfoglou & Schorlemmer, 2003b).

3 DISCUSSION

This is Knowledge Era, an enterprise's economical growth depends upon the wealth of its knowledge and how well it taps onto it and act upon. The task of capitalising on knowledge and get in-depth understanding to ripe benefits, however, is not trivial -- especially when most modern organisations are heterogeneous, physically distributed *virtual enterprises* that consist of many independent organisations. It is therefore paramount that KM tasks are carried out efficiently and effectively, especially when a large OM is present.

Our combined ADP analytical and inference approach provides rich support for KM tasks in the context of OM for enterprise (re-) engineering. Its main advantages are to make use of existing reliable enterprise modelling methods and their known properties, thereby less additional effort is required to elicit maximum benefits for KM tasks and 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 analyse and propose content for the initial seeding of an OM. This is a powerful incentive and tool for enterprise engineers for OM as they can effectively tackle the cold-start syndrome that haunts most of these systems in their initial set up. Combine with FBPML, where template and user-defined business processes are supported, the use of ONA allows identification of knowledge gaps that will inspire new knowledge-incentive business processes. The ONA-based approach coupled with the modelling flexibility of an ADP approach provides an interesting and holistic ontology-based business process support geared towards comprehensive OM for distributed enterprises.

However, our approaches are not entirely infallible, as not all enterprise aspects can be captured explicitly. This is a common challenge when trying to provide a complete set of KM and organisational memory support. When facing trade-offs between utilising knowledge for gains and the cost of capturing and maintaining it, a balance is often stroked. To compensate the information gap caused by informality, one must employ common sense and domain specific knowledge when searching for true answers to queries. Another useful approach of combating missing information is to employ iterative and adaptive business processes as a part of KM life cycle -- thus can improve the underlying three ADP models based on business demands. The quality of KM and answers to queries, therefore, can be improved incrementally over time.

Acknowledgements: This work is supported by the Advanced Knowledge Technologies, Interdisciplinary Research Collaboration project, UK ESPRC GR/N15764/01. The views and conclusions contained herein are those of the authors and should not be interpreted as official policies or endorsements of the EPSRC or any other member of the AKT IRC.

References

Alani, H., O'Hara, K. & Shadbolt, N. (2002). *ONTOCOPI: Methods and tools for identifying communities of practice.* In Proceedings of the 2002 IFIP World Computer Congress, Montreal, Canada.

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

Abecker, A., Bernardi, A., Hinkelmann, K., Kuhn, O. & Sintek, M. (2000). 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.

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

Chandrasekaran,B., Josephson, R. & Benjamins,R. (1999) *What Are Ontologies, and Why Do We Need Them?*, IEEE Intelligent Systems (14)1:20-26.

Chen-Burger, Y-H., Robertson, D. & Stader, Y. (2000) *A Modelling 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.

Chen-Burger, Y-H & Robertson D. (2005) *Automating Business Modelling: A Guide to Using Logic to Represent Informal Methods and Support Reasoning.* Book Series of Advanced Information and Knowledge Processing, Springer Ver-Lag, 2005. ISSN: 1-85233-835-0. DOI: 10.1007/b94136.

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.

Gresse-von Wangenheim,C., Lichtnow, D., von Wangenheim, A. & Comunello, E. (2001). Supporting Knowledge Management in University Software R D Groups, In Proceedings of the 3rd International Workshop on Learning Software Organizations (LSO'01), p51-60, LNCS 2176.

Guo, L., Robertson, D., Chen-Burger, Y-H. (2005) *Enacting the Distributed Business Workflows Using BPEL4WS on the Multi-Agent Platform.* In Lecture Notes in Computer Science, Springer, Ed. Torsten Eymann, Franziska Klugl, Winfried Lamersdorf, Matthias Klusch, Michael N. Huhns. Vol: 3550/2005. ISSN: 0302-9743, pp. 35-46.

Kalfoglou, Y. & Schorlemmer,M. (2003a) *Ontology mapping: the state of the art*, The Knowledge Engineering Review 18(1):1-31.

Kalfoglou, Y. & Schorlemmer, M. (2003b) *IF-Map: an ontology mapping method based on information flow theory*, Journal on Data Semantics (1)1:98-127.

Miller, G.A. (1990). WordNet: an online lexical database. International Journal of Lexicography (3)4:235-312.

Nadarajan, G., Chen-Burger, Y-H. (2006) *Ontological based mapping between a business process model to OWL-S*. In Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid (CCGrid), Singapore.

O'Hara, K., Alani, H. & Shadbolt, N. (2002). *Identifying Communities of Practice: Analysing Ontologies as Networks to Support Community Recognition.* Paper presented at the Proceedings of the 2002 IFIP World Computer Congress, Montreal, Canada.

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

PAKM (2006). Practical Aspects of Knowledge Management. Proceedings of the 6th International PAKM Conference. Karagiannis, D., Reimer,U. (eds), Vienna, Austria.

Schreiber, G., de Hoog, R., Akkermans, H., Anjewierden, A., Shadbolt, N., Van de Velde, W. & Weilinga, B. (Eds.) (1999) *Knowledge Engineering and Management: The CommonKADS Methodology.* MIT Press

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