Knowledge Modelling Techniques for Developing Knowledge Management Systems

Mohd Syazwan Abdullah, Ian Benest, Andy Evans, Chris Kimble
Department of Computer Science, University of York, Heslington, YORK YO10 5DD, UK
Email: {syazwan, ian.benest, andy.evans, chris.kimble}@cs.york.ac.uk

Knowledge management is fast becoming a commercial necessity for many organizations, in order that they manage their intellectual assets and gain competitive advantage. To maximise that advantage, knowledge management needs to be available across the whole of the enterprise. Before a knowledge management system can be built, the knowledge that pervades the organization must be identified and modelled. This paper reviews four important modelling techniques that are used to develop knowledge management systems.

Knowledge modelling, knowledge management, knowledge management system, knowledge, knowledge engineering.

1. Introduction

The lifetime's accumulation of facts, events, procedures and so on, stored in our memories, enable us to exist in this world. With the ending of the single-job-for-life culture, businesses lose much of that knowledge when an individual leaves the organization. Knowledge management (KM) is emerging as the new discipline that provides the mechanisms for systematically managing the knowledge that evolves with the enterprise. Most large organizations have been experimenting with knowledge management with a view to improving profits, being competitively innovative, or simply to survive (Nonaka and Takeuchi 1995, Prusak 1997, Wigg 1997, Hendriks and Virens 1999, Loucopoulos and Kavakli 1999, Davenport and Prusak 2000, Gao *et al.* 2002). Furthermore, exploiting technology enables organizations to derive knowledge from data and information collected as the business proceeds. It then may be exploited in decision-making, product development, human resourcing, customer relationships, the supply chain and so on. Clearly, knowledge management needs to infiltrate every aspect of the enterprise to improve business efficiency.

> *"Knowledge is a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and information. It originates and is applied in the minds of knowers. In organizations, it often becomes embedded not only in documents or repositories but also in organizational routines, processes, practices, and norms".*
>
> Davenport and Prusak (2000)

In the literature on KM, there is much debate about what constitutes knowledge, what is data and what is information. Most literature on KM classifies knowledge into two main categories: explicit knowledge and tacit knowledge. Explicit knowledge can be defined as things that are clearly stated or defined, while tacit knowledge can be defined as things that are not expressed openly, but implied (Choo 2000, Bloodgood and Salisbury 2001, Carvalho and Ferreira 2001, Herschel *et al.*, 2001).

In this paper, we define explicit knowledge as knowledge that can be seen, shared and communicated with others. For example, a business's strategic planning report can be circulated within the organization in any appropriate form and the employees can read and execute the plan. Explicit knowledge can be the business plan, the level of sales, product design, marketing reports and so on. They can be in electronic or paper form, they can be

M.S. ABDULLAH, I. BENEST, A. EVANS, and C. KIMBLE, Knowledge Modelling Techniques For Developing Knowledge Management Systems, 3rd European Conference on Knowledge Management, Dublin, Ireland, September 2002, ISBN:0-9540488-6-5, pp. 15-25.

mathematical formulae or they can exist as diagrams. Tacit knowledge on the other hand is that which is embedded in a person's memory and which is difficult to extract and share with others. For example, how a senior manager uses a particular decision theory to solve certain problems. The knowledge of "how-to solve the problem" is actually the manager's ability, knowledge and skill. While the techniques for problem solving can be learnt in the classroom, how they are used in a particular solution is a personal skill and the solution created by one employee will differ from that of another. These variations are attributable to the differences in their tacit knowledge.

Like data, information and knowledge, the term knowledge management itself is ill defined. It is seen as the systematic means of managing individual, group and organizational knowledge using the appropriate means and technology (Sallis and Jones 2002). It is to do with managing people, what they know, their social interactions in performing tasks, their decision making, the way information flows and the enterprise's work culture. It utilises the power of information communication technologies such as the Internet or the business's local intranet. Nevertheless, it is not a technology-based solution. Most of the knowledge resides in human memories rather than in machines. Technology is a complementary medium that supports the knowledge base.

In this paper, we argue that both types of knowledge can be represented as models, which in turn will facilitate the management of such knowledge. Explicit knowledge can be managed more easily because it exists in a tangible form such as: books, manuals, handbooks and so on, which facilitate communications (Choo 2000) although tacit knowledge can also be managed if it can be converted into explicit knowledge.

Knowledge conversion can be achieved through the processes of: socialisation, externalisation, combination and internalisation (Nonaka and Takeuchi, 1995). For example, through socialisation a manager can learn the tacit secrets of conducting market research from a senior manager (tacit to tacit). Through externalisation, the manager can then translate these secrets into explicit knowledge (tacit to explicit) and communicate it to subordinates (explicit to explicit). The subordinates then standardise this knowledge and put it into a marketing report. Finally, through internalisation, experience gained from conducting market research, enriches the manager's own tacit knowledge base.

2. Knowledge Management Systems

While technology is not the most important aspect of knowledge management, it does play a crucial rôle in facilitating communication and collaboration among knowledge workers in an organization. Both tacit and explicit knowledge can be managed better by using a knowledge management system: a specialised system that interacts with the organization's systems to facilitate all aspects of knowledge processing. For Schreiber *et al.* (1999), knowledge systems are the tools for managing knowledge, helping organizations in problem-solving activities and facilitating the making of decisions. Such systems have been used in the areas of medicine, engineering, product design, finance, construction and so on (Hendriks and Virens 1999, Davenport and Prusak 2000, Chau *et al.* 2002, Tiwana and Ramesh 2002).

Knowledge systems have evolved from knowledge-based systems, which were developed using knowledge engineering techniques (Studer *et al.* 1998). These are similar to software engineering techniques, but with an emphasis on knowledge rather than data processing. Traditional knowledge engineering techniques were widely used to construct expert systems,

M.S. ABDULLAH, I. BENEST, A. EVANS, and C. KIMBLE, Knowledge Modelling Techniques For Developing Knowledge Management Systems, 3rd European Conference on Knowledge Management, Dublin, Ireland, September 2002, ISBN:0-9540488-6-5, pp. 15-25.

systems that are built on the knowledge of one or more experts; essentially, a process of knowledge transfer (Studer *et al*. 1998).
More recently, there has been a paradigm shift in knowledge engineering. Knowledge engineering is no longer simply a means of mining the knowledge from the expert's head (Schreiber *et al*. 1999). It now encompasses *"methods and techniques for knowledge acquisition, modelling, representation and use of knowledge"* (Schreiber *et al*. 1999). The shift towards the modelling approach has enabled knowledge to be re-used in different areas of the same domain (Studer *et al*. 1998). In the past, most knowledge systems had to be developed from scratch every time a new system was needed, and it could not interact with other systems in the organization. The paradigm shift towards a modelling strategy has resulted in reducing development costs. In the next section, we seek to define what is meant by knowledge modelling before exploring some of the most popular approaches in section 4.

3.0 Knowledge Modelling

Models are used to capture the essential features of real systems by breaking them down into more manageable parts that are easy to understand and to manipulate. Models are very much associated with the domain they represent (Savolainen *et al*. 1995). That domain will define their practising communities, modelling languages and the associated tools used. *"A model is a simplification of reality*" (Booch *et al*. 1999). Real systems are large entities consisting of interrelated components working together in a complex manner. Models help people to appreciate and understand such complexity by enabling them to look at each particular area of the system in turn. Models are used in systems development activities to draw the blueprints of the system and to facilitate communication between different people in the team at different levels of abstraction. People have different views of the system and models can help them understand these views in a unified manner.

The modelling process constructs conceptual models of knowledge-intensive activities (Schreiber *et al*. 1999). During the knowledge acquisition stage, most of the knowledge is unstructured and often in tacit form. The knowledge engineer will try to understand both the tacit and the explicit part of the knowledge and then use simple visual diagrams to stimulate discussion amongst users and knowledge experts. This discussion process generates ideas and insights as to how the knowledge is used, how decisions are made, the factors that motivate and so on. The knowledge engineer then has to construct the conceptual model from what has been discussed during the knowledge acquisition stage. This communicates the knowledge to the information specialist who will transform the model into workable computer programs or codes. This approach is similar to that of software engineering where models are used to represent user requirements. The main difference here is that in knowledge engineering it is the modelling of knowledge and its related flows whereas software engineering models the information and process flow.

The importance of knowledge modelling in knowledge management has been discussed by (Wielinga *et al*. 1997). They argue that models are important for understanding the working mechanisms within a knowledge-based system, such as: the tasks, methods, how knowledge is inferred, the domain knowledge and its schemas. Conceptual modelling is central to knowledge engineering (Schreiber *et al*. 1999). Modelling contributes to the understanding of the source of knowledge, the inputs and outputs, the flow of knowledge and the identification of other variables such as the impact that management action has on the organizational knowledge (Davenport and Prusak 2000).

M.S. ABDULLAH, I. BENEST, A. EVANS, and C. KIMBLE, Knowledge Modelling Techniques For Developing Knowledge Management Systems, 3rd European Conference on Knowledge Management, Dublin, Ireland, September 2002, ISBN:0-9540488-6-5, pp. 15-25.

As the paradigm has shifted from the transfer approach to the modelling approach, knowledge modelling has become an important aspect in the process of building knowledge management systems. With the modelling approach, systems development can be faster and more efficient through the re-use of existing models for different areas of the same domain. Therefore, understanding and selecting the modelling technique that is appropriate for different domains of knowledge will ensure the success of the knowledge management system being designed.

4. A Review of Knowledge Modelling

Amongst the many techniques used to model knowledge, the most common are CommonKADS and Protégé 2000, the Unified Modelling Language (UML) with its attendant Object Constraint Language (OCL), and Multi-perspective modelling. The essential features of each are described in more detail below.

4.1 CommonKADS

CommonKADS has become the *de facto* standard for knowledge modelling and is used extensively in European research projects. It supports structured knowledge engineering techniques, provides tools for corporate knowledge management and includes methods that perform a detailed analysis of knowledge intensive tasks and processes. A suite of models is at the core of the CommonKADS knowledge engineering methodology (Schreiber *et al*. 1999) The suite supports the modelling of the organization, the tasks that are performed, the agents that are responsible for carrying out the tasks, the knowledge itself, the means by which that knowledge is communicated, and the design of the knowledge management system (Vollebregt *et al*. 1999, Schreiber *et al*. 1999).

The organization model is regarded as a feasibility study for the knowledge system (Schreiber *et al*. 1999). The study is conducted based on problems and opportunities; it can focus on such areas as: structure, process, people, culture and human power bases, resources, process breakdowns and knowledge assets. The organization model serves three main purposes: the identification of the area in an organization where knowledge-based applications can be implemented, the identification of what impact the knowledge-based application will have in the organization when it is implemented, and it provides the system developers with a "feeling" for where in the organization the applications will be deployed (de Hoog *et al* 1996).

The purpose of the agent model is to understand the rôles played by different agents when performing a task (Schreiber *et al*. 1999). Agents can be people, computers or any other entity that can perform a task. The agent model specifies its characteristics, its authority to perform the task and any associated constraints.

The purpose of the task model is to provide an insight in to the likely impact that introducing the knowledge system will have on the organization (Schreiber *et al*. 1999). The task model refers to the characteristics of the business processes such as: the inputs and outputs, the pre-conditions, performance and quality, the function of the agents that will carry out the processing, the structural coupling of those agents, the flow of knowledge between the agents, their overall control, the knowledge and competences of the agents and the resources available to deliver the business process.

The knowledge model is used to describe the application related knowledge used to perform tasks and the rôle of the knowledge in problem-solving activities (Schreiber *et al*. 1999, Vollebregt *et al*. 1999). The knowledge model of CommonKADS has three categories of knowledge (Motta 1999, Schreiber *et al.* 1999, Visser 1997): task knowledge that describes the order of execution for the reasoning (inference) steps, inference knowledge that describes the reasoning step (inference) performed using the domain knowledge and the domain knowledge itself including its properties, concepts, relations, and so on in the application domain. The communication model describes the inter-agent communication needed when performing the tasks. Finally, the design model is a technical specification of the system in terms of its architecture, platform, modules, constructs and computational mechanisms (Schreiber *et al*. 1999). It brings together all the other models.

.

CommonKADS incorporates an object-oriented development process and uses UML notations such as class diagrams, use-case diagrams, activity diagrams and state diagrams (Manjarres *et al*. 2002, Schreiber *et al.* 1999). CommonKADS also has its own graphical notations for task decomposition, inference structures and domain schema generation (Schreiber *et al*. 1999).

4.2 Protégé 2000

The original Protégé was developed for domain specific applications (Grosso *et al*. 1999, Motta 1999). Now in its latest version, Protégé 2000 is a modelling technique developed by Musen and colleagues from Stanford Medical Informatics. The Protégé 2000 knowledge modelling environment is a frame-based ontology editing tool with knowledge acquisition tools that are widely used for domain modelling (Noy *et al*. 2000; Noy *et al*. 2002). The frames are the main building blocks for a knowledge base (Noy *et al*. 2000). The Protégé ontology (that models the domain) has classes, slots, facets and axioms.

Classes are abstract representations of domain concepts. *"Classes in Protégé 2000 constitute a taxonomic hierarchy and are templates for individual instance frames"* (Noy *et al*. 2000). A sub-class can have all the instances of the class. Protégé 2000 allows multiple inheritance: a class can have two or more super-classes; it also supports a meta-class concept. Slots are properties or attributes of classes. There are two forms of slot. *"Own slots define intrinsic properties of class or individual instance frames. Template slots are attached to class frames to define attributes of their instances, which in turn define specific values for slots"* (Schreiber *et al*. 2001). Slots are first class objects in Protégé 2000; they can be used globally or locally. Facets are properties or attributes of a slot and are used to specify constraints on slot values. The constraints include slot cardinality, (i.e. it specifies the number of values the slot can have), value type for the slot (such as integer, string) and minimum and maximum values for a numeric slot. Axioms define additional constraints on frames; these may link values together, or exploit KIF-based predicate logic.

Instances information is acquired using on-line forms. They are composed of a set graphical entry field and provide an easy-to-use user-interface – an important feature of Protégé 2000. It automatically provides a form to acquire instances of a class when the user defines a class and attaches a template slot to it. The user can customize the form by changing the layout, or changing the form's field labels and can choose different ways of displaying and acquiring slot values (Noy et al. 2000). The knowledge acquisition process in Protégé 2000 consists of three steps. First, a class and its template slot have to be defined. Second, the form to acquire the instances of the class has to be laid out. Finally, the class instances are acquired.

Each class has an associated form and is used to get the instances of the class (Noy *et al*. 2000).

A knowledge base in Protégé is developed in the following sequence. First, concepts and their relationships are defined by an ontology. Second, the domain experts enter their knowledge of the domain area using the domain-specific knowledge acquisition tool. Finally, problem-solving techniques are used to answer questions and problems of the domain using the knowledge base.

4.3 Unified Modelling Language with Object Constraint Language

The Unified Modeling Language (UML) together with the Object Constraint Language (OCL) is the de-facto standard for object modelling as defined by the Object Management Group (OMG). UML is a unification of three main software development approaches: Jacobson's Object-Oriented Software Engineering (OOSE), Rumbaugh's Object Modelling Technique (OMT) and Booch's method (Chen-Burger 2001, Paige *et al*. 2000, Scott 2001).

UML is used to write software blueprints that are used for modelling various types of software-intensive system. *"The conceptual model of the language are UML basic building blocks, the rules that dictate how these building blocks may be put together and some common mechanisms that apply throughout the language"* (Booch *et al*. 1999). There are three components in UML: things, relationships and diagrams. *"Things are the abstractions that are first-class citizens in a model; relationship ties things together; diagrams group interesting collections of things"* (Booch *et al*. 1999).

Structural things are the nouns of the UML model that are conceptual or physical; there are seven types: class, interface, collaboration, use-case, active class, component, and node. "Class" describes objects that share similar attributes relationship, semantics and operations, interface which refers to a group of operations define a service of a class or component, collaboration which specifies an interaction and is a group of roles and other elements that collaborate together, use case which is a group of sequence of actions performed by a system that provides results to the action performer, active class which is a objects of a class that have one or more processes or threads that could perform control activity, component which refers to *a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces* and node which is a physical element that exists during execution and represents a computational resource. Behavioural things are dynamic parts and verbs of UML models and they are: interaction which refers to the communication between a group of objects and state machine which is a behaviour that specifies the sequences of states an objects go through during its lifetime in response to events. Grouping things are the organizational part of UML and package are the basic grouping thing used to organise UML models and annotation things are notes used to explain parts of the UML model.

Relationships are used to describe the connection between instances of model elements such as class. The relationships in UML are: the dependency relationship which is used to show the dependency between things (independent and dependent), association which refers to the structural relationship that links objects, generalization which is a specialised relationship where such an element's object can substitute for a general element's object, and realisation which is a semantic relationship between classifiers.

Diagrams are graphical representations of groups of elements used to visualize a system from different points of view. UML diagrams are: class diagrams, object diagrams, use-case diagrams, sequence diagrams, collaboration diagrams, state-chart diagrams, activity diagrams, component diagrams and deployment diagrams. UML has rules that are applied to models so that they are well-formed semantically. UML has semantic rules for the following: names, scope, visibility, integrity and execution. Common mechanisms are used to ensure the models conform to patterns with common features. The mechanisms are as follows: specifications, adornments, common divisions and extensibility mechanisms.

OCL is a text based language for constraints and queries specification, and is used for writing navigation expressions, Boolean expressions and queries in UML. It can also be used to construct expressions for constraints, guard conditions, actions, pre-conditions and post-conditions, assertions and other UML expressions. OCL's are non-executable and they have predefined operators. OCL are used to express rules in a knowledge system as most knowledge is in some form of rules. This is the powerful feature of OCL in knowledge modelling, as conventional object modelling could not express rules clearly.

UML can be used to model knowledge because, the techniques mentioned earlier supports object concepts such as Protégé 2000, which is developed using an object-oriented programming language, Java and CommonKADS which uses UML diagrams for knowledge modelling process. Earlier version of UML was not intended to support rule-based systems, but due to the powerful features of new versions of UML, OCL and object-oriented programming languages, it has started to get attention from knowledge modellers. UML can be used to model knowledge in the development of intelligence support systems, expert systems and other knowledge systems.

4.4 Multi-perspective Modelling

Multi-perspective modelling enables a number of techniques to be used together, each technique being the most appropriate for modelling that particular aspect of knowledge. Chen-Burger (2001) believes that the multi-perspective modelling technique is important because organizational knowledge is very complex and heterogeneous, and there is no single method that can model all these accurately and appropriately. The multi-perspective modelling technique is used to produce different models of the same artefact to support different viewpoints (Kingston and Macintosh 2000, Kingston 2002). It has its roots in software engineering where it is used to gather requirements for software development projects (Nuseibeh 1996).

Multi-perspective modelling is supported by an Information System Architecture framework (Zachman's framework) and has six categories (what, how, when, who, where and why) for viewing knowledge (Zachman 1987). "What" refers to resources given in the form of declarative knowledge about things (c.f. procedural knowledge about actions). "What" encompasses concepts, physical objects and states. It also includes knowledge about classifications or categorisations of those states. "How" refers to processes, that is, knowledge about actions or events. It includes knowledge about which actions are required if certain events occur; which actions will achieve certain states; and the required or preferred ordering of actions. "When" refers to timing and constraints, that is knowledge about when actions or events happen, or should happen. It includes knowledge about the controls needed to order events. "Who" refers to agents (human or automated); this is knowledge about the agents performing each action, their capabilities and authority to carry out particular actions.

"Where" refers to knowledge about communication, where the knowledge is needed and where it comes from, and how to input and output information. "Why" refers to knowledge about rationale, reasons, arguments, empirical studies and justifications for things that are done and the way they are done.

For example, in a systems development project, the manager has an overall view of the project; the systems analyst's view is that of the requirements for the proposed system; the system designer's view concentrates on aspects of the design; and the programmer's view is concerned with the construction programming code for each module. Different people are involved at different stages of the project and have different perspectives on the project. Different perspectives of the project require different levels of abstraction (Kingston and Macintosh 2000).

The appropriate modelling technique for multi-perspective can be selected from: business management techniques (such as soft system modelling and PERT charts), software engineering techniques (such as flow charts, entity-relationship diagrams and object-oriented analysis and design) and knowledge engineering techniques (such as CommonKADS and VITAL) (Kingston and Macintosh 2000). However, to provide a multi-perspective representation of knowledge, there are three main umbrella methods, namely: CommonKADS, UML, and IDEF. CommonKADS uses it own notation for task decomposition, inference structure and domain schema (Schreiber *et al*. 1999) and UML notations utilize class diagrams, use-case diagrams, activity diagrams and state diagrams (Manjarres *et al*. 2002). The Unified Modelling Language is a multi-model method, a collection of different object development techniques devised by Booch, Rumbaugh and Jacobson (Paige *et al*. 2000, Kingston 2002). IDEF is a suite of methods based on the Structured Analysis and Design Technique (SADT). IDEF0 is suitable for function modelling, IDEF1 is used for specifying entity-relationships, IDEF1X supports the design of relation databases, IDEF3 captures the process description, IDEF4 specifies object-oriented design and IDEF5 captures the ontology description (Kingston and Macintosh 2000).

5.0 Conclusion

In this paper, we have argued that knowledge can be represented in either a tacit or an explicit form. While both may be managed, we suggest that the management of the more explicit forms of knowledge pose fewer problems. We argue that the systems to process and manage such knowledge have evolved from knowledge-based systems developed using knowledge engineering techniques. We further argue that there has been a shift in the emphasis of knowledge engineering techniques recently away from knowledge extraction toward knowledge modelling.

Four knowledge modelling techniques were reviewed. Among the techniques mentioned in this paper, CommonKADS is the only technique that can be considered a knowledge engineering methodology. All these techniques support object-oriented approach in modelling activities and their models are platform independent. CommonKADS, multi-perspective modelling and UML are considered as hybrid approach in modelling as opposed to Protégé which is not a modelling tools in the sense that we use it to draw visual models or diagrams, but it is a tool that allows us to input the knowledge into it's knowledge base. The modelling part of Protégé is already incorporate into the editing tool that could not be seen by the users. UML is a standard for modelling defined by OMG, where else the other techniques are not standardised in a formal manner. All these techniques are fully documented in

various forms. CommonKADS and UML is fully documented in books and reports, Protégé documentations are online at their website, multi-perspective modelling are documented by the respective modelling technique. Most of these techniques are evolving; Protégé is undergoing further enhancement by the Protégé developers, multi-perspective by the respective technique developer and UML by the OMG members. These techniques are useful to model domains, ranging from medical, legal, engineering, business and up to social sciences. Protégé 2000 modelling technique supports Open Knowledge Base Connectivity (OKBC) knowledge model and can be adapted for editing models in different Semantic Web languages and supports RDF (Resource Description Framework) format for saving files. The modelling techniques and their features are listed in Table 1 below.

Technique ***Feature***	***CommonKADS***	***Protégé 2000***	***Multi-perspective***	***UML***
K.E. methodology	✔			
Object-oriented approach	✔	✔	✔	✔
Platform independent	✔	✔	✔	✔
Hybrid approach	✔		✔	✔
Editor tool		✔		
Standard modelling language				✔
Documentation	✔	✔	✔	✔
Evolving		✔	✔	✔
Domain	Medical, legal, engineering, business and up to social sciences	Medical, legal, engineering, business and up to social sciences	Medical, legal, engineering, business and up to social sciences	Medical, legal, engineering, business and up to social sciences
Other features (OKBC, RDF, Semantic web)		✔		

Models are widely used in developing software systems including knowledge management systems. There are various knowledge modelling techniques used to model knowledge, including the ones that are presented in this paper. Each of these techniques are currently complementing each other in one way or another to develop better models of knowledge. As the software modelling world is working on standardising modelling languages to ensure that systems are modelled on a common language, so that the vision of integration, reusability, interoperability will be achieved. Will the same happen to knowledge modelling? This is an important issue as systems of the future, including knowledge management systems are designed to work together with applications that are developed on various platforms. In addition, we have identified the need to assess knowledge modelling languages in the context of appropriate criteria and this will form the basis for future work.

M.S. ABDULLAH, I. BENEST, A. EVANS, and C. KIMBLE, Knowledge Modelling Techniques For Developing Knowledge Management Systems, 3rd European Conference on Knowledge Management, Dublin, Ireland, September 2002, ISBN:0-9540488-6-5, pp. 15-25.

References

1. Bloodgood, J.M. and Salisbury, W.D. (2001) Understanding the influence of organizational changes strategies on information technology and knowledge management strategies, *Decision Support Systems*, Vol. 31, pp. 55-69.
2. Booch, G., Rumbaugh, J. and Jacobson, I. (1999). *The Unified Modelling Language User Guide*, Addison Wesley, Reading, Massachusetts.
3. Carvalho, R.B. and Ferreira, M.A.T. (2001) Using information technology to support knowledge conversion processes, Information Research, 7(1) [*Available at http://InformationR.net/ir/7-1/paper118.html*]
4. Chau, K.W, Chuntian, C and Li, C.W. (2002) Knowledge management systems on flow and water quality modeling, *Expert System With Application*, Vol. 22, pp. 321-330
5. Chen-Burger, J. (2001) Knowledge Sharing and Inconsistency Checking on Multiple Enterprise Models, *Informatics Research Report EDI-INF-RR-0037*, AIAI, University of Edinburgh.
6. Choo, C.W. (2000) Working with knowledge: How information professionals help organisations manage what they know, *Library Management*, Vol. 21, No. 8, pp. 395-403.
7. Davenport, T.H. and Prusak, L. (2000) *Working Knowledge: How organizations Manage What They Know*, Harvard Business School Press, Boston.
8. de Hoog, R,, Benus, B., Vogler, M. and Metselaar, C. (1996) The CommonKADS Organization Model: Content, Usage and Computer Support, *Expert System With Application*, Vol. 11, No. 1, pp. 29-40.
9. Gao, F, Li, M and Nakamori, Y (2002) Systems thinking on knowledge and its management: systems methodology for knowledge management, *Journal of Knowledge Management*, Vol.6, No.1, pp.7-17.
10. Grosso, W.E., Eriksson, H., Fergerson, R.W., Gennari, J.H., Tu, S.W. and Musen, M.A. (1999) Knowledge Modeling at the Millennium (The Design and Evolution of Protégé-2000), *SMI Technical Report* – SMI-1999-0801.
11. Hendriks, P. and Virens, D. (1999) Knowledge-based systems and knowledge management: Friends or Foes?, *Information & Management*, vol.35, pp.113-125
12. Herschel, R.T., Nemati, H. and Steiger, D. (2001) Tacit to explicit knowledge conversion: knowledge exchange protocols, *Journal of Knowledge Management*, Vol. 5, No. 1, pp. 107-116.
13. Kingston, J and Macintosh, A. (2000) Knowledge management through multi-perspective modelling: representing and distributing organizational memory, *Knowledge-Based Systems*, Vol. 13, pp. 121-131.
14. Kingston, J. (2002) Multi-Perspective Modelling: A Framework for Knowledge Representation, Unpublished work, AIAI, University of Edinburgh.
15. Loucopoulos, P and Kavakli, V (1999) Enterprise Knowledge Management and Conceptual Modelling, *Lectures Notes in Computer Science*, Vol.1565, pp.123-143.
16. Manjarres, A., Pickin, S. and Mira, J. (2002) Knowledge model reuse: theraphy decision through specialisation of a generic decision model, *Expert System With Application* – Article in Press
17. Motta, E. (1998) Reusable Components For Knowledge Modelling: Case Studies in Parametric Design Problem Solving, IOI Press, Amsterdam.

M.S. ABDULLAH, I. BENEST, A. EVANS, and C. KIMBLE, Knowledge Modelling Techniques For Developing Knowledge Management Systems, 3rd European Conference on Knowledge Management, Dublin, Ireland, September 2002, ISBN:0-9540488-6-5, pp. 15-25.

18. Nonaka, I and Takeuchi, H. (1995) *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press, New York.
19. Noy, N.F, Sintek, M., Decker, S. Crubezy, M., Fergerson, R.W. and Musen, M.A. (2000) Creating Semantic Web Contents with Protégé-2000, *SMI Technical Report* – SMI-2001-0872
20. Noy, N.F., Musen, M.A., Mejino, Jr. J.L.V. and Rosse, C. (2002) Pushing the Envelope: Challenges in a Frame-Based Representation of Human Anatomy, *SMI Technical Report* – SMI-2002-0925
21. Nuseibeh, B. (1996) Towards a framework for managing inconsistency between multiple views, SIGSOFT 96 Workshop, San Francisco, pp. 184-186.
22. Paige, R.F., Ostroff, J.F., and Brooke, P.J. (2000) Principles of modelling language design, *Information and Software Technology*, Vol. 42, pp. 665-675.
23. Prusak, L. (1997) *Knowledge in Organisations*, Butterworth-Heinemann, Oxford.
24. Sallis, E and Jones, G. (2002) *Knowledge Management in Education: Enhancing Learning & Education*, Kogan Page, London.
25. Savolainen, T., Beeckmann, D., Groumpos, P, and Jagdev, H. (1995) Positioning of modelling approaches, methods and tools, Computers in Industry, Vol.25, pp.255-262.
26. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., de Velde, W.V. and Wielinga, B. (1999), *Knowledge Engineering and Management: The CommonKADS Methodology*, MIT Press, Massachusetts.
27. Schreiber, G., Crubezy, M. and Musen, M.A. (2001) A Case Study in Using Protégé-2000 as a Tool for CommonKADS, 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, pp. 33-48.
28. Scott, K. (2001) *UML Explained*. Addison Wesley, Reading, Massachusetts.
29. Studer, R., Benjamins, V.R., and Fensel, D. (1998) Knowledge Engineering: Principles and method , *Data & Knowledge Engineering*, Vol. 25, pp. 161-197.
30. Tiwana, A., and Ramesh, B. (2001) A design knowledge management system to support collaborative information product evolution, *Decision Support Systems*, Vol.31, pp.241-262.
31. Visser, P., Kralinger, R. and Capon, T. (1997) A Method for the Development of Legal Knowledge Systems, ICAIL'97, Melbourne, Australia, pp.151-160.
32. Vollebregt, A., Teije, A., Harmelen, F., Lei, J. and Mosseveld, M. (1999) A Study of PROforma, a development methodology for clinical procedures, *Artificial Intelligence in Medicine*, Vol. 17, pp. 195-221.
33. Wielinga, B., Sandberg, J. and Schreiber, G. (1997) Methods and Techniques for Knowledge Management: What Has Knowledge Engineering to Offer? *Expert System With Application*, Vol. 13, No. 1, pp. 73-84.
34. Wigg, K.M. (1997) Knowledge Management: Where Did It Come From and Where Will It Go?, *Expert System With Application*, Vol. 13, No. 1, pp. 1-14.
35. Zachman. (1987) *Enterprise Architecture – A Framework*, accessed at http://www/zifa.com/zifajz02.htm.