Knowledge-Based Systems 13 (2000) 121–131
www.elsevier.com/locate/knosys

Knowledge management through multi-perspective modelling: representing and distributing organizational memory

J. Kingston[a,*], A. Macintosh[b]

[a]AIAI, Division of Informatics, The University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, Scotland, UK
[b]International Teledemocracy Centre, Napier University, 10 Colinton Road, Edinburgh EH10 5DT, Scotland, UK

Received 9 November 1999; accepted 15 November 1999

Abstract

Full and accurate representation of an organization's knowledge assets, which together constitute "organizational memory", requires multi-perspective modelling at a number of levels of detail. We propose that the perspectives which need to be represented can be characterized as *who*, *what*, *how*, *when*, *where* and *why* knowledge; these perspectives, and necessary levels of abstraction, are captured by the Zachman framework for Information Systems Architecture. We suggest modelling techniques that might be appropriate for different perspectives and levels of abstraction, and illustrate using examples from a medical domain. We also describe how an individual perspective can become the user interface of a knowledge distribution system, and illustrate this by describing the protocol assistant, a Web-based knowledge-based system capable of representing and reasoning with best practice guidelines ("protocols") in the medical domain. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Organizational memory; Knowledge modelling; Zachman framework

1. Introduction

Management of knowledge within organizations has become a critical activity because many of the activities of organizations today, and of our economic and social life, are knowledge-driven. As Tom Stewart puts it:

> The quintessential raw materials of the Industrial Revolution were oil and steel. Now more than 50% of the cost of extracting petroleum from the earth is information gathering and information processing … more and more of what we buy and sell is knowledge. Knowledge is now the principal raw material [1]

It is therefore critical to an organization's future success to manage this knowledge in a coherent manner, leading to the concept of *knowledge management*.Knowledge management is

> the identification and analysis of available and required knowledge assets and knowledge-asset-related processes, and the subsequent planning and control of actions to develop both the assets and the processes so as to fulfil organizational objectives [2].

The sum of all knowledge assets held by an organization can be considered to be its *organizational memory*.

The above definition of knowledge management implies that it is necessary for organizations:

- to be able to *capture and represent* their knowledge assets;
- to share and re-use their knowledge for differing applications and differing users; this implies *making knowledge available* where it is needed within the organization;
- to *create a culture* that encourages knowledge sharing and re-use.

In this paper, we will focus on knowledge management through information technology—that is, the representation and distribution of knowledge assets using technologies such as knowledge-based systems, ontology support systems, electronic documents, databases and so on. Looking at the above three statements from this viewpoint:

- The explosive growth in use and availability of the World Wide Web and of corporate intranets has provided an unparalleled opportunity to distribute knowledge assets throughout organizations, even multi-national organizations. If these knowledge assets can be represented in an

* Corresponding author. Tel.: +44-131-650-2732; fax: +44-131-650-6513.

E-mail addresses: j.kingston@ed.ac.uk (J. Kingston), a.macintosh@napier.ac.uk (A. Macintosh).

PII: S0950-7051(00)00053-8

Table 1
Descriptions of perspectives

Perspective	Description
What	Declarative knowledge about things as opposed to procedural knowledge about actions. *What* knowledge encompasses concepts, physical objects and states. It also includes knowledge about classifications or categorizations of those states.
How	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	When actions or events happen, or should happen; it is knowledge about the controls needed on timing and ordering of events.
Who	The agents (human or automated) who carry out each action, and their capabilities and authority to carry out particular actions.
Where	Where knowledge is needed and where its comes from—communication and input/output knowledge.
Why	Rationale: reasons, arguments, empirical studies and justifications for things that are done and the way they are done.

accessible and comprehensible computer-based format, the problem of making knowledge assets widely available is, therefore, more or less solved. The choice of format is dependent on user requirements, knowledge form and organizational context; discussion of this is outside the scope of this paper.

- Developing a knowledge sharing culture must be preceded by making "knowledge distribution systems" accessible and comprehensible, for if knowledge re-use is difficult or frustrating, it is likely to be ignored in favour of knowledge re-invention—and a lack of observed knowledge re-use is a powerful disincentive to knowledge sharing.
- The true challenge for technology-assisted knowledge management lies in accurate capture and accessible representation of knowledge assets. These are the areas to which the science of knowledge engineering has the most to contribute.

Previous work on structuring organizational memory has been carried out by van Heijst et al. [3] and Conklin [4]. Van Heijst et al. suggest classifications of organizational memories according to active or passive collection and distribution of knowledge, and proposes that organizational memories should be indexed by hierarchies, attributes of knowledge items and/or knowledge profiles of employees. While these suggestions are helpful, we believe that a broader framework is needed to represent not only knowledge assets, but also the context within which the assets are deployed. This is supported by Conklin, who asserts that:

> knowledge work … requires tools and processes which preserve the context of the work as it evolves, and preserving merely the artifacts of the work (the formal documents) fails to do this. The preserved context [should] take the form of a web of information which includes facts, assumptions, constraints, decisions and their rationale, the meanings of key terms, and, of course, the formal documents themselves.

2. Multi-perspective modelling

The proposal of this paper is that, for a knowledge asset to be represented in a manner that is accurate, complete, embedded in its context and yet comprehensible, *multi-perspective modelling* is required. As the name implies, multi-perspective modelling requires that a knowledge asset should be represented using a collection of knowledge models, each of which takes a different viewpoint on that knowledge. The different viewpoints can be thought of as different managers' views on the organization: for example, an operations manager might view the organization as a user, consumer and producer of resources, while a personnel manager might view the organization as a network of interactions between agents. Knowledge models are often represented as diagrams, using nodes (boxes, circles, diamonds, etc.) to represent items of knowledge and arcs (arrows, lines) to represent relationships between knowledge items. The diagram formats may (and probably should) differ between perspectives, but all knowledge items are drawn from a single underlying repository.

What perspectives are required in order to represent a knowledge asset comprehensively? As a starting point, we draw on the CommonKADS methodology [1] which is intended for supporting the development of knowledge-based systems. CommonKADS proposes six models at successively deeper levels of detail: knowledge engineers are encouraged to model the *organization* in which the system will be introduced; the *task* (business process) which the KBS will support; the *agents* who are or will be involved in the process; required *communication* between agents during the process; the *expertise*[1] which is applied to performing the knowledge-based process; and the *design* of the proposed KBS. Some of these models are sub-divided further; for example, the Expertise model is divided into models of domain concepts and relationships, a model of the required inferences and a model of the control required on the inferences. Despite these models being designed to solve a different type of problem at different levels of abstraction, certain perspectives recur throughout the abstraction hierarchy.

We propose that the various perspectives that are recommended by the CommonKADS methodology can be summarized under the following headings: *how* a process is carried out, *who*does it, *what* information is needed,

[1] The recently published CommonKADS book [1] refers to this model as the "knowledge model". However, to avoid confusion within this paper, we will continue to use the previous term of "expertise model".

Table 2
The Zachman framework (from [7])

	Data (*what*)	Function (*how*)	Network (*where*)	People (*who*)	Time (*when*)	Motivation (*why*)
Objectives/scope "*contextual*"	List of things important to the business	List of processes the business performs	List of locations in which the business operates	List of organizations important to the business	List of events significant to the business	List of business goals/strategies
Enterprise "*conceptual*"	Semantic model	Business process model	Business legacy systems	Work flow model	Master schedule	Business plan
System "*logical*"	Logical data model	Application architecture	Distributed systems architecture	Human interface architecture	Processing structure	Business rule model
Technology constrained "*physical*"	Physical data model	System design	System architecture	Presentation architecture	Control structure	Rule design
Detailed representations "*out-of-context*"	Data description	Programs	Network architecture	Security architecture	Timing description	Rule specification
Functioning enterprise	Data	Function	Network	Organization	Schedule	Strategy

where that information comes from, *when* each activity must be carried out and (less explicitly) *why* the process is performed. Table 1 gives more detail on the expected contents of these perspectives.

A previous paper [5] demonstrated how these six perspectives, as well as the various levels of detail represented by the CommonKADS models, are represented in the Information Systems Architecture framework proposed by Sowa and Zachman [6]. The framework (also called the Zachman framework) has six columns representing *who, what, how, when, where* and *why*perspectives on knowledge, and six rows representing different levels of abstraction (see Table 2). Zachman illustrates the different levels of abstraction using examples from design and construction of a building, starting from the "scope" level (which takes a "ballpark" view on the building which is primarily the concern of the architect, and may represent the gross sizing, shape and spatial relationships as well as the mutual understanding between the architect and owner), going through the "enterprise" level (primarily the concern of the owner, representing the final building as seen by the owner, and floor plans, based on architect's drawings) and on through three other levels (the "system" level, the "technology constrained" level and the "detailed representation" level, respectively, the concerns of the designer, the builder and the subcontractor) before arriving at the "functioning enterprise" level (in this example, the actual building). Zachman describes this framework as

> a simple, logical structure of descriptive representations for identifying 'models' that are the basis for designing the enterprise and for building the enterprise's systems [7].

In practice, the models recommended by the Zachman framework provide "a basis for designing the enterprise" because the higher levels of abstraction (the top to rows) represent the organizational context in detail, and provide "a basis for building the enterprise's systems" because the lower level models act as a comprehensive design specification for enterprise systems. We argue that multi-perspective modelling is essential for accuracy, because it is much simpler to identify when all the knowledge for a single viewpoint has been captured, than to assess the completeness of a complex artefact; it is useful for comprehensibility, because it is known that diagrams of organizational structures or business processes can be readily understood. Multi-perspective modelling at different levels of abstraction overcomes one of the main criticisms of knowledge modelling as an approach to knowledge management: that models are flawed because they eliminate contextual information which may be vital to devising an acceptable approach to leveraging a knowledge asset. Conklin [4] implied that organizational context should encompass "facts, assumptions, constraints, decisions and their rationale", and a set of models that cover all the perspectives of the Zachman framework should contain all this information.

Examples of multi-perspective models are given in Section 4.

3. Modelling techniques

Which modelling techniques can or should be used in order to represent all these perspectives at appropriate levels of abstraction? We believe that a wide range of modelling techniques can be applied, from disciplines as diverse as business management, computer science, psychology and knowledge engineering. A "broad brush" distinction between techniques from different disciplines is that techniques from psychology and knowledge engineering which are designed for representing knowledge usually display a single perspective on knowledge, while many business management and computer science techniques are intended to represent two different perspectives in a single diagram since this is helpful for analysis. Examples of techniques which are appropriate for particular perspectives (several of which are described in a survey of enterprise modelling techniques [8]) include:

- Business management techniques such as soft systems modelling [9] which has been used to represent *how* business processes are performed and *where* communication occurs between processes; role activity diagrams [10] which represent *who* performs particular activities and *where* they communicate with each other (see [5] for an example); and PERT charts, which emphasize *when* particular activities can be carried out, especially if critical path analysis is applied.
- Software engineering techniques such as flow charts, which represent *how* processes are carried out and *when* each step may be carried out; entity-relationship methods which represent *what* knowledge belongs to entities and *where* they obtain information from; and object-oriented analysis and design techniques which cover the same perspectives as entity-relationship methods but in more detail, and may also represent *how* objects process information.
- Representations drawn from research in cognitive psychology, such as classification hierarchies which represent *what* knowledge is used and its categories; repertory grids [11] which gather information about *what* attributes and values are associated with a concept, and which concepts are similar; and decision trees, which represent *how* decisions are taken in a step-by-step manner.
- Knowledge engineering techniques, both those designed to cover many domains (such as VITAL—the name of both a methodology project and its support tool. A good starting point is Domingue [12] or CommonKADS) or those designed for specific domains (such as PRO*forma* [13], a method for representing protocols in the medical domain).

In addition, there are at least three families of methods that have been designed to provide a range of models that can provide a multi-perspective representation of knowledge: CommonKADS, UML and IDEF.

- CommonKADS covers most of the perspectives (as described above), and also most of the levels of abstraction: the Organizational model is suitable for addressing the "scoping" level; the Task and Agent models for the "enterprise" level; the Expertise and Communication models for the "system" level; and the Design model for the "technology" level.
- The Unified Modeling Language (UML) [14] recommends a range of diagrams, including use case diagrams (*who* knowledge), class diagrams (*what* knowledge), activity or state chart diagrams (*how* knowledge), sequence or collaboration diagrams (*where* knowledge) and component or deployment diagrams (the same types of knowledge already covered, but at lower levels of abstraction).
- The IDEF suite of methods [15] provides a range of modelling techniques for different tasks: IDEF0 is a "function modelling" method based on the Structured Analysis and Design Technique; IDEF1 is an entity-relationship modelling method; IDEF1X is a method for designing relational databases; IDEF3 is a process description capture method; the IDEF3 Object State Transition network shows the changes in state of objects are used by the processes; IDEF4 is an object-oriented design method; IDEF5 is an ontology description capture method, and so on. Like the CommonKADS models, different IDEF models are most useful at differing levels of abstraction.

To summarize, several modelling techniques are capable of representing each perspective, and it matters little which one(s) is used as long as every perspective is covered, the technique is appropriate for the desired level of abstraction, and the representation produced is sufficiently detailed for the problem being tackled.

4. Multi-perspective modelling example: clinical procedures

For reasons of space, it is impossible to give a fully detailed example of multi-perspective models showing all six perspectives at all six levels of abstraction. Instead, we will look at all six perspectives at a single level of abstraction, using an example of following clinical procedures in an otolaryngological (ear, nose and throat) department. The example focuses on the treatment of a particular condition (parotid swellings—possibly malignant growth on the parotid gland, which is in the side of the neck). A model of a particular procedure within an organization is usually considered to be at the "enterprise" level of abstraction.

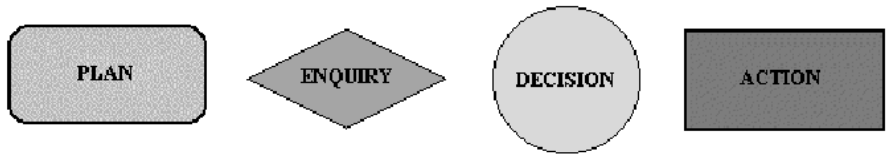

Fig. 1. The four basic node types of PRO*forma*.

4.1. The how perspective: clinical protocols

The expected contents of the *how* perspective, at higher levels of abstraction, are addressed by Cook in the context of building enterprise information architectures [16, pp. 103–104]. She suggests that the *how* perspective should include functions at the scope level, primitive functions at the enterprise level and processes at all lower levels. Functions are defined as "a high-level group of business activities … that likely creates more than one entity", primitive functions are defined as "a high-level business activity … that creates a single entity" and processes as "a business activity … they create a single state of an entity and then move it forward into the next state".

Using this definition, the enterprise level of the *how* knowledge should create a single entity; in this case, the entity being created is a clinical diagnosis with a subsequent treatment plan. In this domain, the *how* knowledge can be represented by a clinical protocol. A clinical protocol gives a step-by-step guide for carrying out a certain specialized procedure, drawing on all (and only) the experiments and other knowledge relevant to that procedure. The motivation behind developing clinical protocols is to capture and represent "best practice".

In this example, the clinical protocol will be represented using PRO*forma*. PRO*forma*[13] is a simple but expressive technique for modelling best-practice guidelines in medicine. Its knowledge representation language supports four basic node types (Fig. 1):

- A *Plan* is a sequence of sub-tasks, or components, which need to be carried out to achieve a clinical objective, such as a therapeutic objective. Plan components are usually ordered, to reflect temporal, logical, resource or other constraints.
- An *Enquiry* is a task whose objective is to obtain an item of information that is needed in order to complete a procedure or take a decision. The specification of an enquiry includes a description of the information required (e.g. a lab result) and a method for getting it (e.g. by query on a local patient record or a remote laboratory database).
- A *Decision* occurs at any point in a guideline or protocol at which some sort of choice has to be made, such as a diagnostic, therapeutic or investigative choice.
- An *Action* is a procedure that is to be enacted outside the computer system, typically by clinical staff, such as the administration of an injection.

Fig. 2 shows a portion of the protocol that will be used throughout this example: it guides clinicians through the decision on how to treat a progressive lump (i.e. a lump that is growing progressively larger). See [17] for more details on this protocol.

FNAC stands for "fine needle aspiration cytology", which represents a clinical test that can be performed on fluid drawn from a parotid swelling. One of several forms of scan may then be performed; treatment choices include excising the lump (formal parotidectomy), chemo- or radio-therapy or no treatment (because the risks of treatment outweigh the benefits).

4.2. The where perspective: inter-department communication

The *where* perspective shows communication that is needed during a procedure. At the enterprise level of abstraction, communication is generally concerned with the transfer of information or artefacts between individuals or departments. In the example above, the clinician must communicate with the laboratory that performs the FNAC tests, with the radiology department that performs scans, and with the surgical unit that arranges operations.

This information can be represented in a Role Activity Diagram (RAD) [10], which shows which departments (or, more generally, which roles) perform which activities; by including the sequence of activities, the needs for communication become obvious. An example of an RAD can be seen in Fig. 3.

4.3. The who perspective: agent modelling

In addition to the information captured in an RAD, there is a need for the *who* perspective to represent the *capability* of agents, departments or other role-players to perform certain actions and the *authority* that certain agents have to perform those actions or to use, consume or modify resources (Fig. 4). At an enterprise level of abstraction, capability and authority may be expressed by defining the *rights* and *responsibilities* of an agent. For example, a doctor may have right to add to a patient's medical record, implying both authority to change an artefact and the capability to do so, as well as responsibilities such as making sure a patient's medical record is kept up to date.

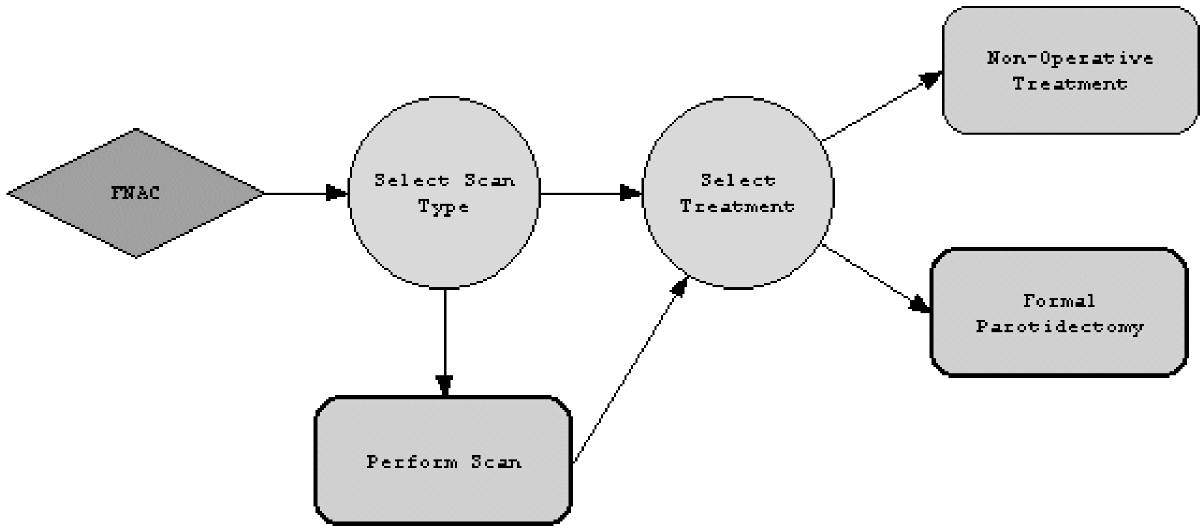

Fig. 2. Protocol for diagnosing a progressive lump.

The modelling technique used here is loosely based on the ORDIT method for requirements definition [18] and the ComonKADS Agent Model [19]. Capability, authority, rights and responsibilities are represented by four different arcs: these arcs are labelled “can”, “may”, “has rights to” and “must”.

4.4. The what perspective: data, information and resources

The *what* perspective considers the *data* and *information* that are referred to and the *resources* that are used, consumed, modified, manipulated or otherwise involved in the overall process. Cook argues that

> the data architecture is more critical than the process architecture because most business processes exist to manage the assets, not the other way around [16].

She proposes that the enterprise level of the *what* perspective should contain data classes, which are sub-classes of global data classes; the relationships between classes can be defined using entity-relationship diagrams. In practice, these data classes often subsume information (“data with relevance and purpose … data that makes a difference” [20]) such as summations or categorizations as well as data. In this example, data classes might include clinical tests and patients; information represented in data classes might include results of tests; and resources include the machines required for scanning, the chemical solutions required as “markers” for scanning, and the needles required for extracting fluid for an FNAC. The resources may have

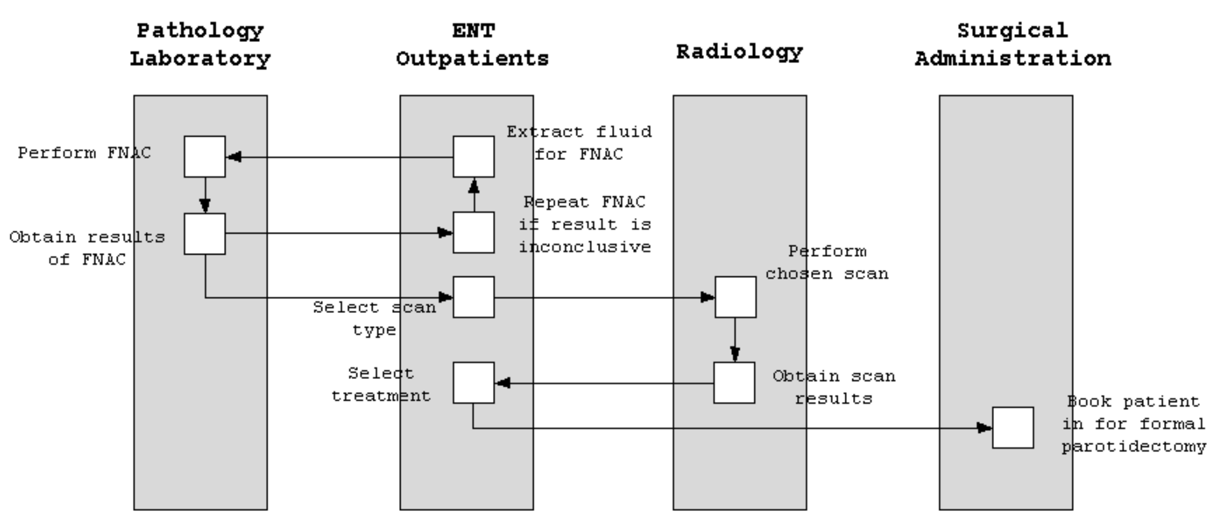

Fig. 3. The “where” perspective—RAD.

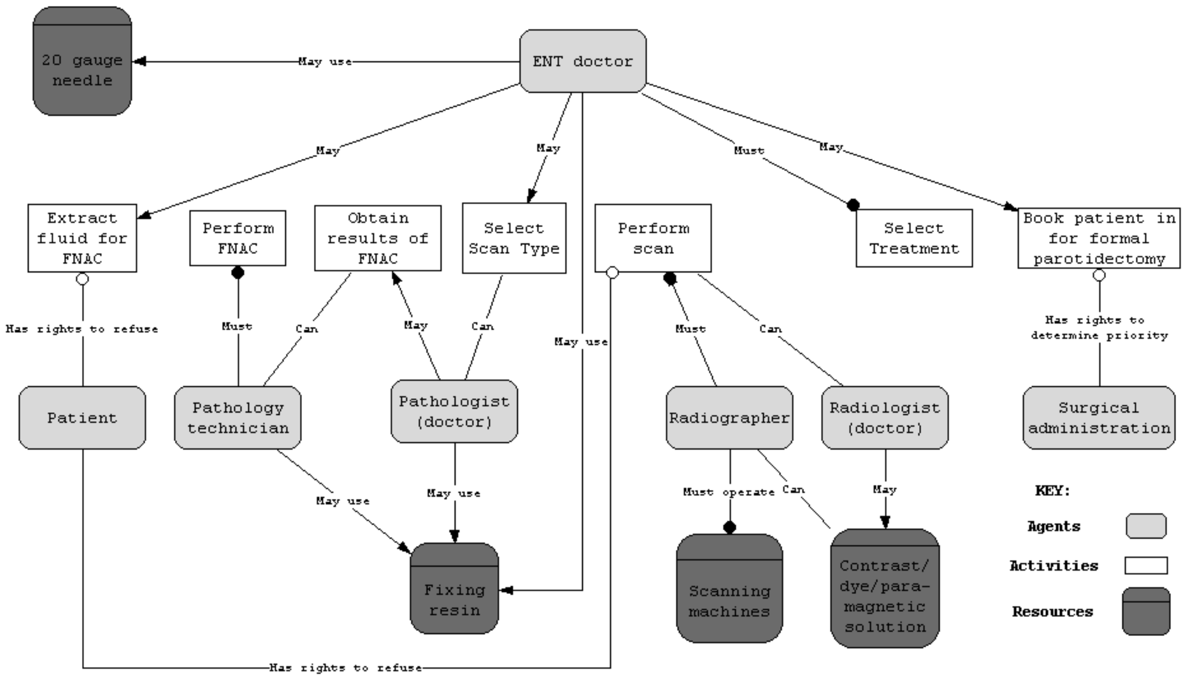

Fig. 4. Capabilities, authorities, rights and responsibilities of agents.

associated constraints; for example, that use of a scanning machine requires several weeks notice, or that patients might be allergic to the iodine-based "contrast" that is injected as a marker for CT scans.

At the system level of abstraction, where resources, constraints and information artefacts are identified individually, there are several ways in which resources might need to be modelled. If the resources can be grouped into classes, then a taxonomic hierarchy might be advantageous; for example, it might be helpful to know if scanning machines belong to a class of machines that uses computerized timing chips, and if so, whether they belong to the sub-class of machines that uses millennium-compatible chips. If a detailed representation of relationships between resources was needed, then a semantic network could be drawn. However, at the enterprise level of abstraction, a more general representation is more appropriate; an entity-relationship diagram could be used, but we have chosen to use a UML class diagram, to represent constraints more clearly. Fig. 5, therefore, shows a UML class diagram representing a simple hierarchy of resources and a simple hierarchy of test results.

4.5. The when perspective: schedules

Information about timing of activities and actions is very important in a planning problem; for other tasks, such as this diagnostic task, there is less need for such information. It is, nevertheless, advisable to draw a PERT chart, GANTT chart or a simple timeline of activities and any necessary inter-activity delay (such as the waiting list for scanning appointments) in order to highlight any time-related issues (such as the fact that the chemicals used for marker solutions have a limited shelf life) or bottlenecks.

Fig. 6 shows a PERT chart of activities and inter-activity delays; the duration (which appears at the bottom of the activity nodes) is in hours. It shows the two bottlenecks in the process (waiting lists for scanning and for operations) clearly. N.B. For illustrative purposes, it has been assumed that the "select scan type" activity can be carried out in parallel with awaiting the results of the FNAC.

4.6. The why perspective: published clinical evidence

The *why* knowledge for a clinical protocol consists of clinical evidence—published results of clinical trials, meta-studies and expert opinions. The relative importance of different forms of clinical evidence has been discussed in Ref. [21]. For the small part of the clinical process that we are considering, the *why* knowledge consists of all known articles published to date; at the time when this protocol was prepared, there were eight relevant published articles. Five of them argue for or against a particular type of scan, the others argue for or (primarily) against formal parotidectomy.

These justifications can be represented in a rationale diagram; Fig. 7 uses and extends the QOC (Questions, Options and Criteria) notation [22] to represent rationale for the "select scan type" decision.

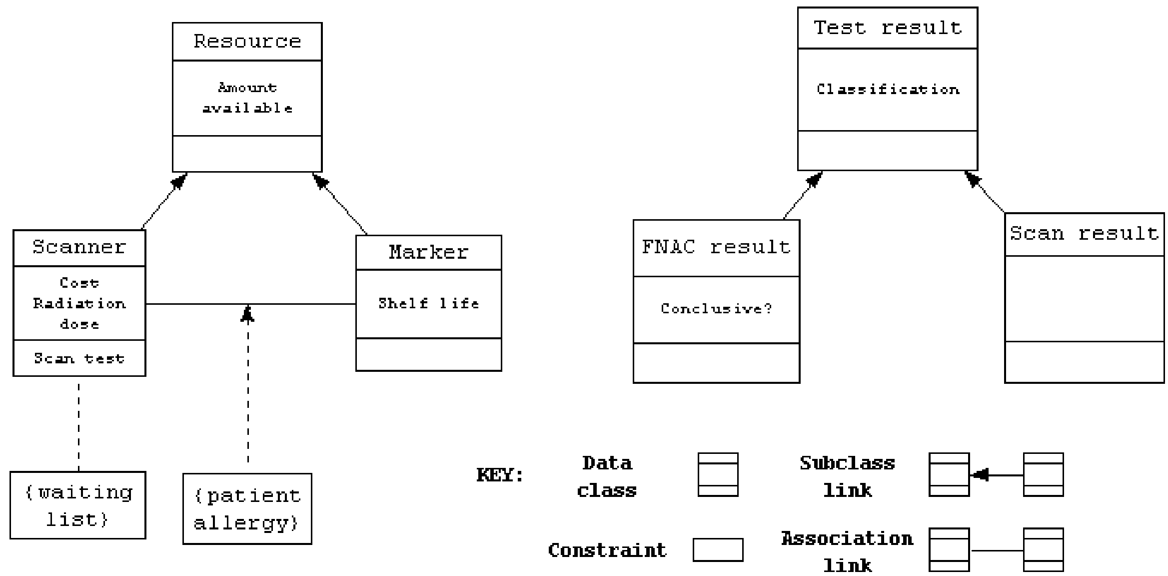

Fig. 5. UML class diagram: resources and test results.

5. Knowledge distribution systems: using models as user interfaces

We have shown how multi-perspective modelling uses different viewpoints to provide a complete yet comprehensible representation of knowledge; see Ref. [2] for a further discussion of the value of this approach in knowledge management. However effectively organizational memory is represented this representation is of little use unless it can be accessed and understood by members of the organization. In this section, we argue that a good user interface for a knowledge management system can be obtained by presenting the user with one of the graphical models representing a single knowledge perspective.

5.1. How a single perspective acts as a knowledge index

By using a single-perspective model as a user interface, a single coherent view on the entire knowledge asset is presented to the user, thus aiding comprehension. Since most knowledge items appear in more than one perspective, this model can also be used as an index to other

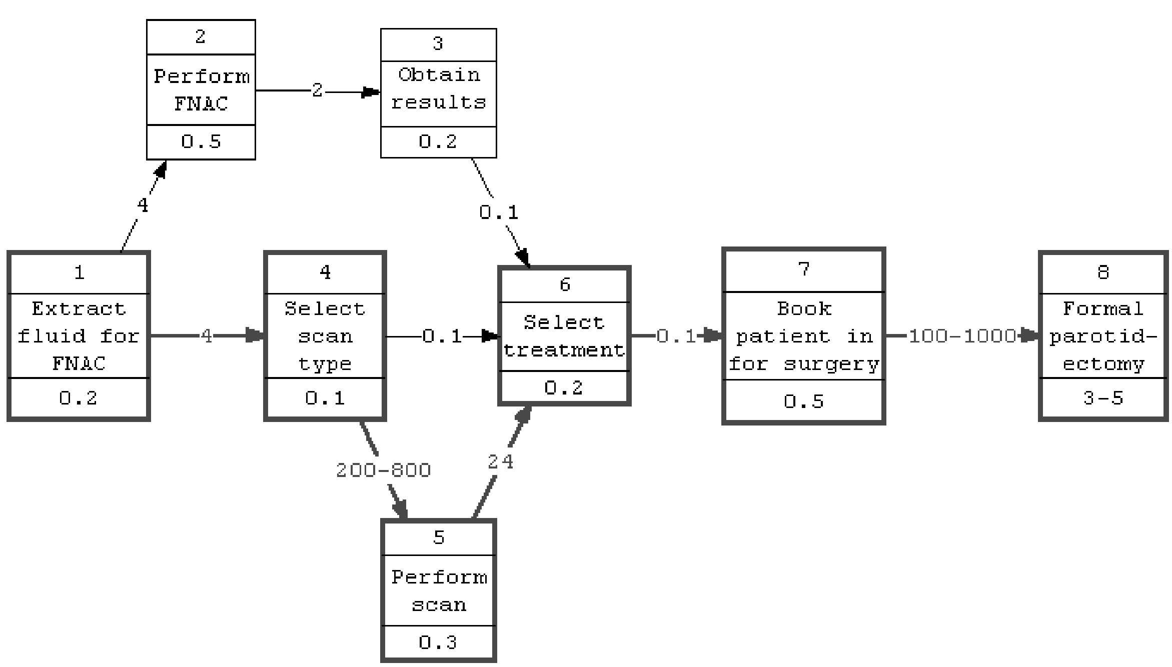

Fig. 6. A timeline of activities in diagnosing progressive lumps.

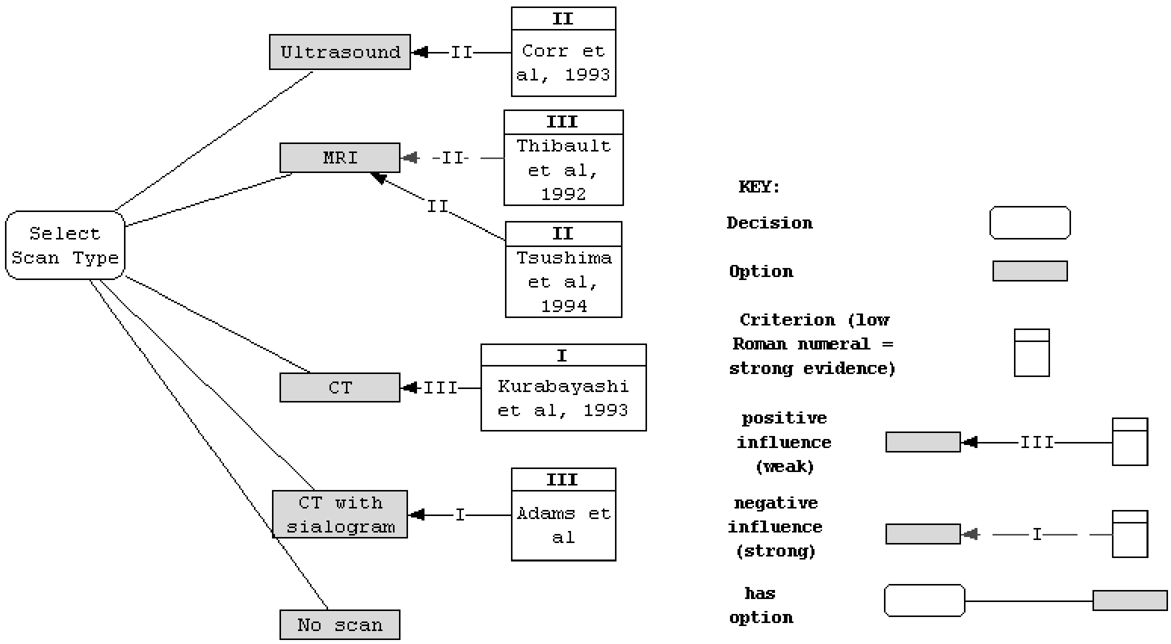

Fig. 7. Extended QOC diagram showing the rationale for a decision.

perspectives, by linking information from other perspectives to appropriate knowledge items. This provides users with a structured way of navigating through knowledge from the various perspectives. By using a coherent (and hopefully validated) knowledge model as a user interface, users should be able to understand the structure of the knowledge better. In fact, a knowledge model user interface may not only assist comprehension but also concurrence and commitment to the knowledge. Conklin [4] proposes that knowledge development teams should use "knowledge display systems", arguing that:

> a remarkable thing happens when knowledge teams use a display system to treat informal knowledge as if it were valuable … there is less repetition in meetings, more rigor in decisions, and it is easier to bring others up to speed on the team's thinking and learning. In other words, when you take process-oriented knowledge seriously, the process itself immediately improves.

It seems possible, even likely, that giving a "display system" of organizational knowledge to the knowledge users will generate a similar effect of taking the knowledge seriously, and thus create process improvements over and above those expected.

The suggestion that a single-perspective model can act as an index to the other perspectives is an important one, because it is relatively easy to put into practice, since it is usually possible to identify knowledge that "belongs", or *should* belong to a knowledge item. In the *how* perspective, knowledge relevant to procedural steps may include justifications, case histories, organizational policies, descriptions of how to perform the step or entire sub-procedures; in the *who* perspective, knowledge relevant to agents might include experience, authority, responsibilities, legal restrictions and appraisals; for the *what* perspective, knowledge relevant to classes might include ontological definitions, default attributes, class members and key differentiators; and so on. Some of this knowledge is drawn from other perspectives (for example, justification of decisions may be drawn from the *why* perspective), while other knowledge will be specific to a perspective (for example, instructions on how to carry out a procedure). If this knowledge can be linked directly to the relevant knowledge item in the model (via hypertext, pop-up menus or whatever), then the user interface becomes an index to all knowledge relevant to particular knowledge items, thus providing the fundamentals for a good decision support system. In addition, knowing what knowledge is expected to "underlie" a particular component of a model greatly simplifies the identification of what knowledge *needs* to be captured in order to provide an adequate representation, which is a major stimulus for overcoming one of the biggest problems of knowledge management: the existence of large amounts of tacit knowledge, whose existence and contents are not only unrecorded, but may even be outside the conscious awareness of those who have and exercise the knowledge [23].

Choosing which perspective to use for the user interface depends on the purpose of the knowledge distribution system. If it is purely a "knowledge browser", then deciding which model to use as the user interface probably comes down to which model contains the most detail. For other purposes, however, particular perspectives may be more appropriate. Some suggestions are given below:

- The *how* perspective is useful for representing standard organizational procedures or regulated procedures.
- A taxonomy of classes and sub-classes, representing the declarative (*what*) perspective, can be used to distribute organizational ontologies, in order to support either structured addition to the organizational memory or structured searching for knowledge.
- An agent-based model (the *who* perspective) can go beyond organizational structure charts to indicate the roles, capabilities and responsibilities of different agents within the organization, thus providing a basis for project and resource management.
- The *when* perspective provides additional control information to the *how* perspective. Such knowledge can be used for planning and scheduling applications.
- The *where* perspective can constitute a front end for real-life multi-agent tasks such as distributed collaborating project groups.
- The *why* perspective can be used as the front end of a justification system, such as a system for supporting legal argumentation, or a system which represents the justifications for regulations (e.g. PLINTH [24]).

5.2. A knowledge distribution system: the protocol assistant

The protocol assistant [17] is a system that illustrates these principles. It is a prototype expert-decision support system for diagnosis and treatment of parotid swellings that can be run through an Internet browser; the many advantages of such a distribution method include ease of knowledge maintenance—if the evidence base should change through publication of new clinical studies, a single designated “protocol maintainer” can introduce the change into the knowledge base and make it immediately available to all users of the system. It was designed with four purposes in mind:

- to represent clinical protocols that describe specialized medical procedures in an accurate yet accessible format;
- to capture and represent the “evidence base” of clinical studies that provide evidence for or against particular steps in the protocol;
- to demonstrate how these protocols can be distributed over the Internet using HTML diagrams, and how abstracts of the relevant clinical studies can be made available to medical practitioners;
- to allow the system to reason with these clinical studies, according to the principles of “evidence-based medicine”, thus allowing the entire protocol to be “run” as well as browsed.

A clinical protocol constitutes a standard organizational procedure, so the most appropriate perspective for the user interface is the *how* perspective. The protocol assistant used PRO*forma* models as its interface, and was able to function both as a decision support system (due to the hyperlinking of relevant underlying knowledge) and as an expert system.

In identifying which knowledge from other perspectives was relevant to each node, we were assisted by the PRO*forma* methodology, which specifies attributes that should be associated with each of its node types. Every node has attributes for preconditions, post-conditions and triggers (part of the *when* information); in addition:

- Decisions have attributes for argument rules for different candidate options (*why* knowledge) as well as rules for commitment to a candidate (more detailed *how* knowledge drawing on *what* knowledge) and information sources required (*where* knowledge).
- Plans have attributes for scheduling constraints, abort conditions and termination conditions (more *when* knowledge).
- Actions have attributes for associated procedural descriptions (*how* knowledge).
- Enquiries have attributes for sources (primarily *where* knowledge) and capture *what* knowledge.

By specifying available knowledge in shorthand form in these attributes (for example, encoding a clinical study that strongly supports CT scanning as “++CT: ref Kurabayashi”), and making some further knowledge available through hyperlinks (for example, the abstract of the Kurabayashi study and other clinical studies were hyperlinked to the HTML representation of the “select scan type” decision node), the PRO*forma*model acted as a very effective index for most of the relevant underlying knowledge. In addition, an expert system was implemented, based on the knowledge model, allowing the user the choice of browsing the available knowledge through a Web browser or being guided through it by an expert system.

6. Conclusions

We have argued that the two main technology challenges for knowledge management are to capture and represent knowledge assets accurately and accessibly. Knowledge engineering methods such as CommonKADS have come a long way towards meeting these challenges; they provide disciplined multi-perspective approaches to designing and building knowledge-based applications.

The six models from the CommonKADS methodology are a good starting point for the modelling of knowledge assets. However, these models were not defined with the goal of developing organizational memories. The Zachman framework helps us to define what aspects of organizational memory need to be represented; the necessary perspectives and detail will vary with the type of knowledge being acquired, the expected users of the system and the purpose of the knowledge management effort. We suggest that knowledge engineers should apply whatever modelling techniques they prefer, as long as all the necessary perspectives are covered. We have suggested some techniques that are appropriate for particular perspectives or levels of abstraction.

We have further claimed that knowledge distribution systems based on the organizational memory can employ one of these perspectives as a user interface; which perspective is chosen will depend on the goal of the application. The individual elements of the chosen perspective provide a useful index for accessing organizational knowledge relevant to that element, as well as aid the user in understanding the structure of the knowledge and the system. We have illustrated this argument by describing the protocol assistant, which represents, distributes and reasons with a clinical protocol, along with its supporting evidence base, within a Web browser.

References

[1] G. Schreiber, H. Akkermans et al., The Common KADS Methodology, MIT Press, December 1999.
[2] A. Macintosh, I. Filby, J. Kingston, Knowledge Management Techniques: Teaching and Dissemination Concepts, International Journal of Human Computer Studies, vol. 51(3), Academic Press, New York, 1999 (special issue on organizational memories and knowledge management, September).
[3] G. van Heijst, R. van der Spek, E. Kruizinga, Organizing Corporate Memories. Proceedings of KAW'99, http://ksi.cpsc.ucalgary.ca/KAW/KAW96/vanheijst/HTMLDOC.html, Banff, Alberta, November 1996.
[4] E.J. Conklin, Designing Organizational Memory: Preserving Intellectual Assets in a Knowledge Economy. http://www.gdss.com/DOM.htm, 1996.
[5] J.K.C. Kingston, T.J. Lydiard, A. Griffith, Multi-perspective modelling of air campaign planning, Proceedings of AAAI-96, AAAI Press, 1996, Menlo Park, California, USA.
[6] J.J. Sowa, J.A. Zachman, Extending and formalizing the framework for information systems architecture, IBM Systems Journal 31 (3) (1992) 590–616 (IBM publication G321-5488).
[7] J. Zachman, The Framework for Enterprise Architecture, http://www.zifa.com/zifajz02.htm.
[8] J. Fraser, A. Macintosh (Eds.), Enterprise State of the Art Survey, The Enterprise Consortium, available from AIAI, The University of Edinburgh, Edinburgh, UK, http://www.aiai.ed.ac.uk, September 1994.
[9] P. Checkland, J. Scholes, Soft Systems Methodology in Action, Wiley, New York, 1990.
[10] M. Ould, Process Modelling with RADs, IOPener: The Newsletter of Praxis plc, 1-5 to 2-2, 1992–1993.
[11] K.M. Ford, J.M. Bradshaw, Knowledge Acquisition as Modeling, Wiley, New York, 1993 (chap. 1).
[12] J. Domingue, The VITAL Workbench, http://kmi.open.ac.uk/~john/vital/vital/vital.html.
[13] J. Fox, N. Johns, et al., Protocols for medical procedures and therapies: a provisional description of the PRO*forma* language and tools, Proceedings of AIME'97, Springer, Berlin, 1997.
[14] UML Resource Center, Rational Software, http://www.rational.com/uml/, 1995–1999.
[15] R.J. Mayer, P.C. Benjamin et al., A Framework and a Suite of Methods for Business Process Reengineering, http://www.idef.com/articles/framework/framework_pt1.htm, Knowledge Based Systems Inc., 1998.
[16] M.A. Cook, Building Enterprise Information Architectures: Reengineering Information Systems, Prentice-Hall, Englewood Cliffs, NJ, 1996.
[17] J. Simpson, J. Kingston, N. Molony, An expert system for best practice in medicine, Proceedings of Expert Systems 98, SGES Press, Cambridge, 1998 (also published in Knowledge Based Systems 12 (5–6) (1999) 247–255.).
[18] J.E. Dobson, A.J.C. Blyth, J. Chudge et al., The ORDIT Approach to Requirements Identification, Technical Report 394, Computing Laboratory, University of Newcastle upon Tyne, 1992.
[19] A. Waern, S. Gala, The CommonKADS Agent Model. Esprit Project Report P5248 (KADS-II), KADS-II/M3/TR/SICS, Swedish Institute of Computer Science, 1994.
[20] T.H. Davenport, L. Prusak, Working Knowledge: How Organizations Manage What They Know, Harvard Business School Press, Cambridge, MA, 1998.
[21] A.G. Maran, N.C. Molony, M.W.J. Armstrong, et al., Is there an evidence base for the practice of ENT surgery?, Clinical Otolaryngology 22 (1996) 152–157.
[22] A. Maclean, R. Young, V. Bellotti et al., Design space analysis: bridging from theory to practice via design rationale, in: Proceedings of Esprit'91, Brussels, November 1991, pp. 720–730.
[23] I. Nonaka, H. Takeuchi, The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation, Oxford University Press, New York, 1995.
[24] A. Casson, PLINTH: Integrating hypertext, semantic nets and rule-based systems in an expertext shell for authors and readers of regulatory information, in: Proceedings of CIKM'93, Workshop on Intelligent Hypertext, Arlington, VA, 1993 (also available as AIAI-TR-142, http://www.aiai.ed.ac.uk/).