British Journal of Management, Vol. 10, 309–322 (1999)

# From Knowledge Engineering to Knowledge Management[1]

Nigel Shadbolt and Nick Milton
Artificial Intelligence Group, School of Psychology, University of Nottingham, University Park, Nottingham NG7 2RD, UK

**Knowledge management is seen by many to be a prerequisite for the successful organization, and one that relies heavily, though not exclusively, on a sound technological infrastructure. A major drawback, though, with current technology (e.g. Lotus Notes and www) is its focus on information management and communication rather than on knowledge itself. What knowledge management needs is tools and techniques that are more oriented towards knowledge – its creation, mapping, transfer and use. We show how many of the methods and tools used in the branch of artificial intelligence known as knowledge engineering can be adapted to provide such a knowledge-oriented technology, and lead to significant benefits for organizations. A number of case studies are presented which illustrate our points, including decision-making at Andersen Consulting and best practice at Rolls-Royce. A more elaborated use is shown in the context of business process re-engineering, where a new software tool kit called SPEDE is being applied and validated within the aerospace and automotive industries.**

## Introduction

The concept of knowledge management has undoubtedly become a major force in business thinking in recent years. Many large organizations are embracing knowledge management and claiming significant benefits. Books such as *Working Knowledge* (Davenport and Prusak, 1998) and *The Knowledge-Creating Company* (Nonaka and Takeuchi, 1995) demonstrate with multiple examples that many of the world's most successful organizations are those that are best at managing their knowledge.

Organizations use a number of methods and software tools to support knowledge management. Most of these tools, however, have been developed as part of an unstructured technological thrust: they are more concerned with new ways of storing and communicating information than with the actual ways in which people create, acquire and use knowledge. Lotus Notes and the world wide web are examples of these types of technologies. While we do not deny the importance of such tools, we believe that new methods and tools are needed that can supplement existing technologies, but that are specifically oriented towards knowledge. Such methods and tools would act as a bridge between people and current technologies. They would improve support for key activities such as knowledge creation, knowledge mapping, knowledge retrieval and knowledge use.

Our central thesis is that there are methods and tools, developed in the field of artificial intelligence, that can (and should) provide such a technology. This does not mean, however, that we believe managers should fill their companies with intelligent systems such as expert systems, neural networks and case-based reasoning systems. Instead,

[1] Parts of this work have been supported by an IMI grant from the Engineering and Physical Sciences Research Council of Great Britain. We thank all those involved in the SPEDE project for their help and assistance. We especially thank Paul Riley for work with Rolls-Royce, Hugh Cottam for ideas contained in this paper, and Jeni Tennison for invaluable technical editing.

as we describe later, the methods and tools that we consider particularly important for knowledge management are those that have been used in the *development* of intelligent systems, rather than the intelligent systems themselves. Such methods and tools have a track record in providing a bridge between people and computers.

The paper is structured as follows. We start, in the next section, with a brief background to knowledge engineering, and describe some of the principles that have emerged over the past 10–15 years. In the third, we describe how these principles can help alleviate three major concerns of knowledge management. The fourth section shows how the principles are realized in techniques and software tools. In the fifth section, we present three case studies to demonstrate how the techniques and tools are being used in real knowledge-management situations. Finally, we finish with a summary and concluding remarks.

## Knowledge engineering

### *Background*

Artificial intelligence has a relatively long history in dealing with knowledge from both a theoretical and practical perspective. The many influences on artificial intelligence (e.g. philosophy, psychology and linguistics) bring to it a rich heritage of ideas and a sound foundation in applied science. Although the early years were dogged by grand notions that machines would soon solve every problem as proficiently as the world's best expert, artificial intelligence now has goals that are more realistic.

Alongside this increased pragmatism, a major shift in emphasis came approximately 20 years ago. Before then, people believed that it was theoretically possible to build a problem-solving machine that could take a few pieces of information and use its massive computing power to solve any problem. When people tried to build such a machine, however, they found that far from power being the major element needed, knowledge was the major requirement (Minsky and Papert, 1974). For machines to do intelligent things, they required knowledge – lots of it and in a structured form, so that it could be retrieved and used when needed. Such machines are known as 'expert systems' since they are designed to mimic the knowledge and the reasoning processes of an expert practitioner, such as a physician, geologist or chemist.

However, the need to acquire knowledge for an expert system became a problem in itself. Knowledge engineers found that acquiring enough high-quality knowledge to build a robust and useful system was a very long and expensive activity. Indeed, acquiring knowledge became the 'bottleneck' in building an expert system (Hayes-Roth, Waterman and Lenat, 1983). This led to 'knowledge acquisition' becoming a major research field within knowledge engineering. The aim of this field is to develop methods and tools that make the arduous task of 'transporting' knowledge from inside an expert to inside a computer efficient and effective (Shadbolt and O'Hara, 1997).

Over the past 15 years, knowledge engineers have developed a number of principles, methods and tools that have made knowledge acquisition an efficient and effective activity. Interestingly, many of the issues and problems that are being written about in the knowledge management literature are familiar territory to those involved in knowledge engineering. Thus, many of the principles, methods and tools of knowledge engineering may have relevance to knowledge management. In recent years, knowledge engineers have begun to adapt, test and validate knowledge engineering methodologies within real knowledge management initiatives. In the next section, we cover some of the principles from knowledge engineering that are relevant and useful to knowledge management.

### *Knowledge engineering principles*

*Principle 1: Recognize that there are different types of knowledge.* Philosophers have been thinking about knowledge for thousands of years. Part of their endeavour has been the identification of different types of knowledge.

One distinction is between *declarative* knowledge (knowledge of facts) and *procedural* knowledge (knowledge of how to do things). A popular way of thinking about these is the difference between 'knowing that' and 'knowing how' (Ryle, 1949). This distinction has been recognized and used for a long time in psychology. In knowledge engineering, these two types are often referred to as *static* knowledge and *dynamic* knowledge: we will use these terms in the rest of this paper.

Another well-known classification of knowledge is that of *tacit* knowledge (cannot be articulated easily) and *explicit* knowledge (can be articulated easily). Since this is of particular relevance to knowledge management, we shall leave discussion of these until later.

A particularly important way of classifying knowledge is to what extent it is *abstract* (applies across many situations) or *specific* (applies to one or a few situations). Developing ways in which specific knowledge can be made more abstract and abstract knowledge can be made more specific has been a major effort in knowledge engineering.

The field of logic has also inspired other important knowledge types: *concepts*, *attributes* and *values*. For instance, within the statement 'it is a fast car', we refer to 'car' as being a concept, and 'fast' as being a value of the attribute speed. Concepts, attributes and values can be linked together to form another important class of knowledge – *rules*. For example, 'if it is a fast car, then it is an expensive car'.

There are other types of knowledge, some of which we will mention later in this paper. The important point is that it is easier to deal with knowledge if you can recognize different types of knowledge and understand how they relate to one another.

*Principle 2: Recognize that there are different types of experts – and expertise.* Not only are there different types of knowledge, but there are different types of experts. As such, it is important to identify the different types of experts involved in a project and tailor the methods accordingly.

The difference between knowledge that can be articulated easily and knowledge that cannot be articulated easily is a very important distinction for both knowledge engineering and knowledge management. Psychologists and knowledge engineers have found that experts vary in how well they can articulate their knowledge. This is because the nature of experts' knowledge varies depending on their training and experience (Chi, Glaser and Farr, 1988). Shadbolt and Burton (1995) identified different types of experts, ranging from those whose knowledge of a domain is almost completely tacit (called 'samurai'), to those whose knowledge is almost completely explicit (called 'academics').

In addition to differences in ability to articulate knowledge, experts vary in how well they recall information in a given context. Studies in psychology have repeatedly shown that experts are not able to remember the same things during interviews as they can when they are performing a task. In addition, the ability to recall the same information in different tasks can vary between individuals. For instance, those with experience of teaching others in a classroom setting are usually better at explaining their knowledge than those without such experience.

Finally, experts may vary in the validity of the knowledge they can articulate. It is a well-known phenomenon in psychology that people are biased and error prone when they try to explain something: they attempt to save face or show off, they rationalize or misinterpret what happens and so on. Again, people may vary in the extent to which they do this, and it is important to recognize this. Knowledge engineers have devised a number of methods to overcome such problems, such as aggregating knowledge from various sources and validating knowledge across sources.

*Principle 3: Recognize that there are different ways of representing knowledge.* The field of artificial intelligence may not have produced fully intelligent machines, but one of its major achievements is the production of a range of ways of representing knowledge. Explanations of these representations (which include various forms of logic, rules, semantic networks and frames) is beyond the scope of this paper. However, producing different knowledge representations is a vital part (arguably *the* vital part) of artificial intelligence, since the ease of solving a problem is almost completely determined by the way the problem is conceptualized and represented.

The same is true for the task of communicating knowledge. A well-chosen analogy, anecdote or diagram can make all the difference when trying to communicate a difficult idea to someone, especially someone who is not an expert in the field. Indeed, the history of science contains many examples of people only understanding new theories when theoreticians use the appropriate representation.

*Principle 4: Recognize that there are different ways of using knowledge.* People use knowledge in different ways depending on the task they are performing. From a knowledge perspective, it is useful to be able to classify tasks. A number of

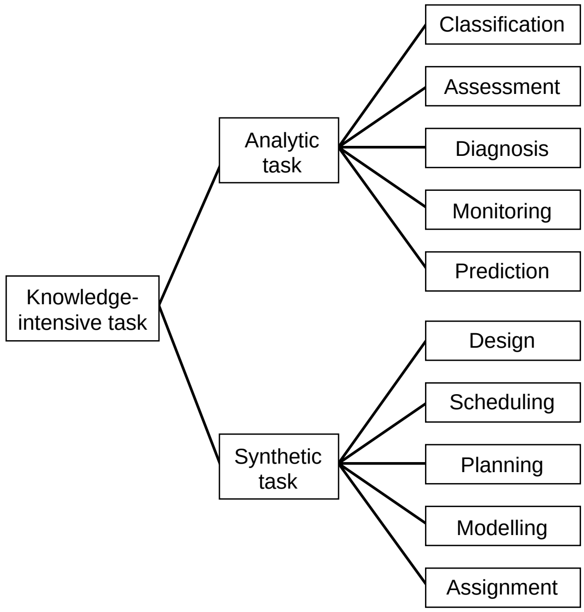


*Figure 1. Hierarchy of knowledge-intensive task types (adapted from Schreiber* et al.*, forthcoming)*

ways of classifying tasks have been created in knowledge engineering. One classification, adapted and refined from ideas in psychology, is a hierarchy of knowledge-intensive tasks based on the type of problem being solved (Schreiber *et al.*, forthcoming), as shown in Figure 1. Knowledge engineers have created models that define the types of knowledge that form the inputs and outputs of the task, as well as how the knowledge is transformed for each of the tasks in Figure 1. Using these models, the knowledge involved in a task can be defined in terms of the way it is used to satisfy a goal or set of goals. This can not only increase the efficiency of task analysis and process modelling, but also enables one to identify how the same piece of knowledge is used differently depending on the context in which it is used.

*Principle 5: Use structured methods.* This principle leads on from the first four. We have shown that there are different types of knowledge, different types of experts, different ways of representing knowledge and different ways of using knowledge. Therefore, you need a way of relating these types of knowledge, experts, representations and tasks together to perform a knowledge-oriented activity. To do this, requires well-defined methods: methods that are structured such that the right techniques and tools are used depending on the situation and, especially, the goals involved. Using these methods, one will not try to interview experts about knowledge they cannot articulate, represent it in a form no one will understand and eventually find they do not really need it anyway. To avoid such situations, knowledge engineers have developed three aids:

1. a range of techniques and software tools, described later;
2. formal descriptions of general static knowledge, called ‘ontologies’;
3. formal descriptions of general dynamic knowledge, that is, models of problem solving such as grammars.

The use of ontologies and problem-solving models is still in its infancy within the context of knowledge management, and is beyond the scope of this paper. However, the essence of these topics is that knowledge can and should be reused both within, and across, domains.

## Three important problems

The problems faced in knowledge management are many and varied: we certainly do not believe that technology generally (let alone knowledge engineering technology) is the panacea for them all. We do believe, however, that the principles described in the previous section can begin to provide solutions to some important problems. Three knowledge management problems that repeatedly crop up are listed below.

1. Organizations contain such a vast amount of knowledge that mapping all of it would be both impossible and a waste of time.
2. Tacit knowledge is vital to an organization, yet it is very difficult to acquire and map.
3. Ordinary language is the main form of communication, yet it is so full of jargon, assumptions and ambiguities that people often fail to understand what others are trying to say.

In the rest of this section, we will show how knowledge engineering principles can help address these problems.

*Mapping the right knowledge in the right way*

Avoiding the need to map too much knowledge requires a number of techniques and tools based on the principles described in the previous section. Seven of the more important considerations (some of which have already been mentioned) are:

1. choose what knowledge needs to be mapped by first capturing the goals and uses to which the knowledge is to be put;
2. decide on the scale (i.e. the level or granularity) of the knowledge to be acquired;
3. choose from a range of tools, so the right one is used for the job;
4. reuse knowledge (do not reinvent the wheel);
5. piece together knowledge from multiple sources;
6. validate knowledge by checking it with other experts;
7. structure knowledge as it is captured, so that it is easier to search databases and retrieve the relevant knowledge.

Some of these may look familiar to those involved in knowledge management. If so, this supports our idea that knowledge engineering can be of use to knowledge management. However, if you feel there is nothing new here, our plea is that the utility of knowledge engineering for knowledge management is in the techniques and tools which allow us to satisfy the seven aims listed above. These will be covered in the next section.

*Making tacit knowledge explicit*

*Tacit knowledge*, first described by Polanyi (1966), is knowledge that is impossible to articulate (e.g. how to ride a bicycle). *Explicit knowledge,* on the other hand, is knowledge that can be articulated (e.g. a bicycle has two wheels). Tacit knowledge is often thought of as knowledge of how to do things (i.e. procedural knowledge). However, not all procedural knowledge is tacit: for example, it is possible to articulate how to boil an egg. Similarly, not all tacit knowledge is procedural: for example subconscious ideas are not procedural in nature. Tacit knowledge is, by definition, impossible to articulate: thus the acquisition and sharing of tacit knowledge can be a long and difficult task. However, some believe that it is worth the effort, given the vital role that tacit knowledge plays in organizations.

In their influential book *The Knowledge-Creating Company,* Nonaka and Takeuchi (1995) argue that the success of Japanese companies is a result of their skills and expertise at organizational knowledge creation. They state that ‘the key to knowledge creation lies in the mobilisation and conversion of tacit knowledge’. Their theory of organizational knowledge creation explains how tacit and explicit knowledge interact, and how knowledge is converted from one to the other. They demonstrate that the main ways in which tacit knowledge is made explicit is through metaphors, analogies, concepts, hypotheses and models.

While we agree with much of what Nonaka and Takeuchi have to say, we feel two aspects need addressing. The first is their assumption that knowledge is either tacit or explicit. In our experience, we have found that knowledge is often neither exclusively tacit nor exclusively explicit, but lies on a continuum between the two. A simple everyday example of this is a person trying to remember something saying, ‘it’s on the tip of my tongue’. ‘Tip of the tongue’ knowledge is far more explicit than the term ‘tacit knowledge’ would suggest: it often only requires a small prompt to be made explicit (e.g. by being given the initials of a person’s name you cannot quite remember). Another example, mentioned previously, is the existence of different types of experts: given two experts in the same area of expertise (i.e. domain), one expert may find it easy to articulate his or her knowledge, while the other may find it difficult.

Our second problem with Nonaka and Takeuchi is the lack of advice on how technology can help the process of making explicit tacit knowledge. As we will show in the next section, techniques such as repertory grids can be used to expose tacit knowledge. Other techniques, such as analysing sequences of behaviour, can uncover knowledge that the expert cannot articulate, assumes is too obvious to mention, or has not previously noticed. Prompts are an important way of moving knowledge from the tacit end of the continuum to the explicit end. Providing the right prompts requires you to identify what types of knowledge you are trying to capture, what type of expert is involved and what representations to use. Structured methods, along with the techniques and tools described later, can help achieve this.

### *Avoiding the Tower of Babel*

People talk different languages. They use jargon, acronyms and shortcuts when they talk to other experts within their domain. They have common assumptions that they do not need to articulate. When experts talk to people who are not experts in their domain, they find it difficult to break out of this: they assume their audience has a lot more knowledge and understanding than it really does. Language is also rather imprecise. People use the same word to mean different things (ambiguity) and use different words to mean the same thing (synonyms). These characteristics of language can lead to major problems for organizations – lack of knowledge dissemination, misunderstandings and the hardening of functional boundaries.

One technique in knowledge management that attempts to alleviate such problems is the construction of a thesaurus of company terminology. We believe this is a useful first step. However, other actions need to be taken. A prime need is to form a clear and precise 'language' in which to represent the terminology of the domain. Forming the 'language' requires first classifying the terminology in terms of the types of knowledge present, then determining where it resides on a hierarchy of knowledge – abstract knowledge at the top, specific knowledge at the bottom. The location of a term is determined by the attributes associated with it, which also need to be analysed and classified. Using this technique, one can build a more precise description of the domain, and associate it with the everyday language that people use. This is an example of an 'ontology', that is, a formal description of the static knowledge in a domain. Work in knowledge engineering has developed a number of ontologies.

As part of the SPEDE project, which we describe later, we have been developing a set of ontologies. One of these is based on linguistics and is specifically used to help people communicate. Experience has shown that familiar grammatical categories (e.g. nouns, verbs, and adjectives) are a good means of analysing a domain into its ontological primitives (e.g. concepts, tasks and attributes). Novice knowledge engineers, in particular, find the mapping intuitively appealing and useful. A special grammar, based on this ontology, has been created to facilitate the process of transforming natural language descriptions into more formal notations, and then back again. Use of this grammar aims to help employees deal with the jargon, synonyms and ambiguous terminology that act as barriers between them and the knowledge they need.

## Techniques and tools

### *Techniques*

A number of techniques have been developed in knowledge engineering to embody the principles described earlier. Unfortunately, lack of space does not allow us to describe many of the techniques: a fuller catalogue is given by Wielinga, Sandberg and Schreiber (1997). Instead, we shall focus on one particular set of techniques that constitutes a methodology for acquiring knowledge from an expert during 'knowledge acquisition'.

Two types of techniques are used. One type is natural techniques (e.g. interviews and on-the-job observation); the other type is 'contrived' techniques. Knowledge engineers have developed the latter to capture various types of knowledge that are either inefficient or impossible to acquire using natural techniques. These contrived techniques generally involve special ways of representing knowledge and/or special tasks that the expert is set (Hoffman *et al.*, 1995). Many of the tools described here support these contrived methods. The methodology we shall briefly cover is a process of moving from natural to contrived techniques. It is summarized in the following steps.

1. Conduct an initial interview with the expert to (a) scope what knowledge should be acquired, (b) determine to what purpose the knowledge should be put, (c) gain some understanding of key terminology, and (d) build a rapport with the expert. This interview (as with all encounters with experts) should be recorded on either audiotape or videotape for later analysis.
2. Transcribe the initial interview and analyse the resulting document (called a 'protocol') to produce a set of questions that cover the essential issues across the domain and that serve the goals of the knowledge-acquisition exercise.
3. Conduct a second interview with the expert using the pre-prepared questions to provide structure and focus. (This is called a 'semi-structured interview'.)
4. Transcribe the semi-structured interview and analyse the resulting protocol, looking for knowledge types: concepts, attributes, values, classes of concepts, relationships between concepts, tasks and rules.
5. Represent these knowledge elements in a number of formats, for example, hierarchies of classes (taxonomies), hierarchies of constitutional elements, grids of concepts and attributes, diagrams and flow charts. In addition, document, in a structured manner, anecdotes ('war stories') and explanations that the expert gives.
6. Use the resulting representations and structured documentation with contrived techniques to allow the expert to modify and expand on the knowledge you have already captured.
7. Repeat the analysis, representation-building and acquisition sessions until the expert is happy that the goals of the project have been realized.
8. Validate the knowledge acquired with other experts, and make modifications where necessary.

This is a very brief description of one method for knowledge acquisition. It does not assume that any previous knowledge has been gathered, or that any generic knowledge can be applied. In reality, the aim would be to reuse as much previously acquired knowledge as possible. Knowledge engineers have developed techniques to assist this. For instance, knowledge engineers use ontologies and grammars to suggest ideas to the expert such as general classes of concepts in the domain and general ways in which tasks are performed. This reuse of knowledge is the essence of making the knowledge-acquisition exercise as efficient and effective as possible. The science of knowledge acquisition is an evolving process: as more knowledge is gathered and abstracted to produce generic knowledge, the whole process becomes more efficient.

In the next section, we describe a set of software tools that can be used to support the kind of methodology described above.

### *Tools*

PC-PACK is a commercially available set of knowledge engineering tools that we developed following research on a variety of projects (Shadbolt, Motta and Rouge, 1993). Further information, as well as demonstration software, is available on the Internet (http://www.epistemics.co.uk). Although PC-PACK comprises twelve tools, we have found that a subset of these is of particular relevance to knowledge management: these are described below.

*Protocol Editor.* The Protocol Editor tool is used to analyse transcripts of interviews or any other text-based information, e.g. reports, specifications or manuals. The aim is to identify the important aspects of the domain, for example, the concepts, attributes, tasks and relationships. The tool simulates the way someone would mark a page of text using different coloured highlighter pens. Each type of knowledge is associated with a different colour, for example, green for concepts, yellow for attributes. Using this tool, the user can go quickly through a document highlighting the important knowledge. Figure 2 illustrates the use of the Protocol Editor.

When the user has completed highlighting a document, the marked text can be automatically saved in the PC-PACK database for use in all other tools.

*Laddering tool.* The Laddering tool enables the user to build various hierarchies of knowledge. It

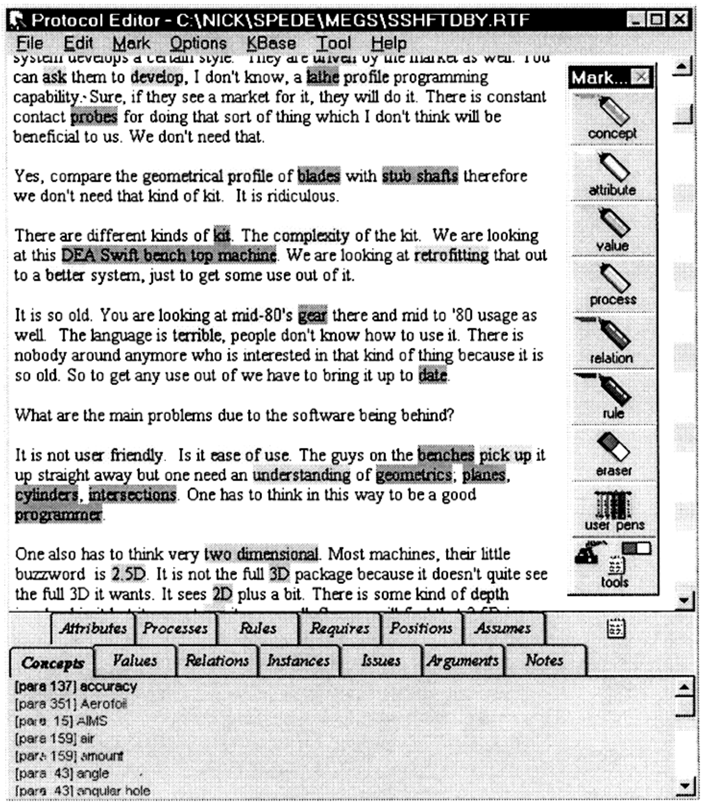


*Figure 2. Screen shot from the PC-PACK Protocol Editor*

displays each hierarchy as a tree, with abstract categories towards the left, and specific categories towards the right. Instances (i.e. unique entities) are shown on the extreme right. The term 'laddering' comes from a particular way of acquiring knowledge that involves a structured questioning technique (Corbridge *et al.*, 1994). PC-PACK provides three basic types of ladders: a ladder of concepts, a ladder of attributes and values and a ladder of tasks. Within each of these, the user starts by using the knowledge elements identified in the Protocol Editor tool, and adds new concepts, attributes and relationships as the laddering continues. Full editing facilities are available: for example, an element in a ladder can be moved to another position using a simple 'drag and drop' operation. Figure 3 illustrates the Laddering tool in use.

A fourth type of ladder is also included. This representation, suggested by Andersen Consulting, is a ladder of requirements (called the 'Requirements Ladder'). This is covered later.

*Matrix tool.* The Matrix tool supports associating attributes to concepts, as well as entering values. It presents a matrix (i.e. grid) with concepts listed down the left, and attributes listed across the top. This tool is similar to a spreadsheet; however, it has some special features that help to save time when entering knowledge. One feature is that the hierarchical structures from the concept and attribute ladders are retained in the matrix. This enables a method called 'inheritance': when an attribute is associated with a high-level concept, it, and the values it takes, are inherited by all its sub-concepts. This is an example of 'default reasoning'. For example, one does not want to have to enter the number of wheels that the concept 'car' has for every type of car. It is easier to assume a default value of four wheels for *all* cars, then only overwrite the value for cars without four wheels (e.g. three-wheelers).

*Repertory Grid tool.* The Repertory Grid tool provides a number of facilities. The first feature

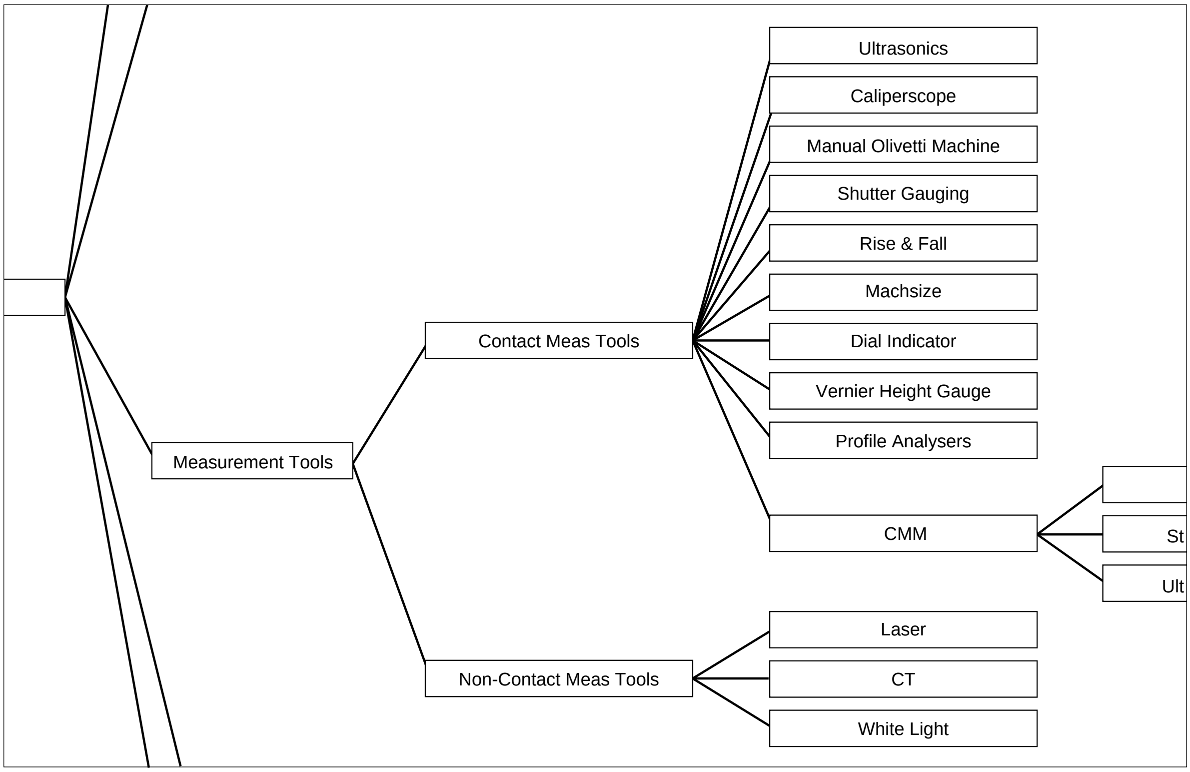


*Figure 3. Screen shot from the PC-PACK Laddering tool*

allows the user to generate new attributes using a technique called 'triadic elicitation'. Simply put, the tool asks the expert what is similar and different about three randomly-chosen concepts, that is, in what way are two of them similar and different from the third. This is an effective way of eliciting attributes that experts cannot immediately articulate. The second feature of this tool is based on rating concepts along a continuum for which the end points are opposing attributes: for example, fast–slow, or accurate–inaccurate. This idea is based on a psychological theory called personal construct psychology (Kelly, 1955), which is widely used in areas such as clinical assessments. A 9-point scale is used in PC-PACK, so that the expert can make fine distinctions between concepts. For instance, the relationship between various measuring methods can be established by assessing each of them on scales such as accurate–inaccurate, cheap–expensive and reliable–unreliable. A third feature of this tool applies the resulting scales to a statistical technique called cluster analysis. This produces a grid of concepts against attributes in which the position of concepts and attributes in the grid shows how similar they are to other concepts or attributes, i.e. it clusters together those with similar scores on the scales. Figure 4 shows such a grid for various measuring methods used in manufacturing.

The lines to the bottom and right of the grid in Figure 4 indicate similarity ratings – joining lines that are closer to the grid indicate more similarity between linked concepts or attributes. In Figure 4, we can see that *accuracy* and *ease of use* are similar, indicating that accurate machines are also easier to use. This relationship may be something the expert had not articulated before, or not have noticed before. Thus, the tool can be used to uncover hidden correlations and causal connections, i.e. tacit knowledge.

*Control Editor.* Unlike the previous tools, which deal essentially with static knowledge, the Control Editor is used for acquiring and representing dynamic knowledge: it is used for building

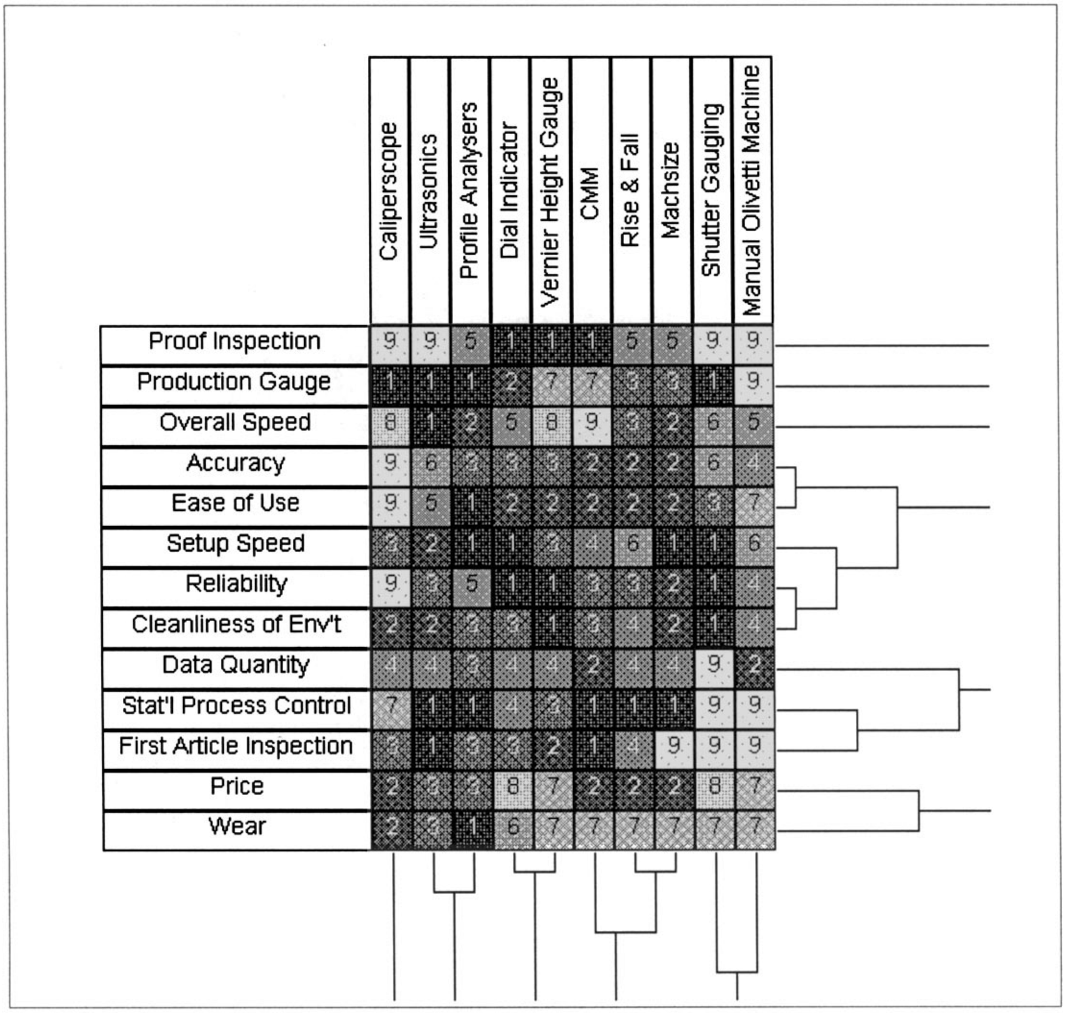


*Figure 4. Screen shot from the PC-PACK Repertory Grid tool*

process-control diagrams for individual tasks or groups of tasks. This tool has different symbols for tasks, for the inputs and outputs of tasks, for decision points, for data flow and for control flow, in a similar way to a flow chart. It also has a number of specialized symbols, including one for the ‘agent’ (i.e. person, group or computer) that performs a task. A particularly important feature of the Control Editor is the way a hierarchy of tasks can be built and displayed. Hierarchies are navigated using a three-dimensional feel, whereby double-clicking on a task symbol will reveal a ‘lower’ page representing how that task is performed. Using this facility, the user can start by modelling a task at a very high level (made up of a few sub-tasks), then double-click on each of these sub-tasks to build the next layer of the task hierarchy. This is much clearer than having a single large two-dimensional diagram, and allows the user to set the scale (i.e. level or granularity) of what knowledge is acquired.

*Hyperpic tool.* The tools described so far have mostly supported activities such as analysis and codification. The Hyperpic tool allows the user to take a more flexible approach to storing, acquiring and using knowledge. Right-clicking on any of the concepts, attributes or tasks shown in the other PC-PACK tools invokes the Hyperpic tool. The tool allows text and pictures to be entered in order to document/annotate the knowledge already represented. The tool supports a hypertext format that allows words to be highlighted and 'linked' to other pages in the document (in the way the www operates). This allows structured knowledge documents to be constructed, which can be based on the hierarchies produced in the Laddering tool. We often use the Hyperpic tool to produce a thesaurus of relevant terminology. For those who want to produce Internet or intranet knowledge documents, the Hyperpic tool can automatically produce files in the correct format (i.e. Hypertext Markup Language [HTML]) at the 'press of a button'. Organizations use PC-PACK in this way to provide a smooth and efficient transition from inside the expert's head to inside the company's Internet-based knowledge base.

## Case studies

To illustrate the ways in which the principles, techniques and tools from knowledge engineering can be of real practical benefit to knowledge management, we present four case studies in this section. All cases involved the use of the PC-PACK knowledge tool kit either in its original form, or in a form adapted to the needs of the particular organization and application. The case studies we present here are not the only cases that illustrate our message.

### *Andersen Consulting*

Andersen Consulting is a global management and technology consulting organization whose mission is to help its clients to become more successful. The organization helps its clients link strategy, people, processes and technology.

Andersen Consulting use PC-PACK for both expert-system development and knowledge management. Montero and Scott (1998) give a full description of its use in systems development, so we will focus on its other role – as an aid to knowledge auditing, knowledge sharing and decision-making.

Numerous decisions are made to arrive at the best solution during processes such as planning and design. However, often the organization loses much of the knowledge of why a particular decision was made and what alternatives were discarded (i.e. the 'audit trail'): those involved either forget what happened or fail to keep adequate records. Thus, stakeholders in such decisions (e.g. managers, decision implementers, and future decision-makers) have no access to knowledge that could be vital for them. Studies show that this is especially important in large-scale development projects (Ramesh and Dhar, 1992). A tool that can capture the rationale behind decisions would, therefore, be an important aid to knowledge management.

Andersen Consulting wanted such a tool to be added to PC-PACK. At their suggestion, part of the Laddering tool was configured to allow a requirements engineer to ladder requirements, issues, positions, arguments and assumptions, and the relations between them, as well as to define and visualize alternative decisions (as sets of positions on issues). Requirements engineers use the Laddering tool, with this addition, in combination with the Protocol Editor. According to Andersen Consulting, the result is not only useful for modelling, but also a 'powerful brainstorming tool for the interaction between expert and analyst' (Montero and Scott, 1998).

### *Unilever*

Unilever is an international organization with over 300 000 people worldwide and having over 1000 brands. The importance of knowledge management to Unilever's continued success is recognized at the highest levels of management. Unilever currently use PC-PACK in both in its 'traditional' role during the development of expert systems, and in more non-technological areas of knowledge management. It is the latter that we briefly describe below.

The main use of PC-PACK to support knowledge management at Unilever is in 'knowledge workshops'. By projecting what would normally appear on the VDU on to a large screen, a group of experts can interact and represent the current state of their thinking. In this way, they can build and critically analyse an ontology of the domain in question. Tools, such as the laddering and matrix tools, are used dynamically to add, delete and modify elements. Thus, the group can reach a consensus on the main knowledge categories and relationships in the domain. Since this is not known beforehand, these sessions are as much about 'brainstorming' as they are about knowledge mapping. One can see this as an example of what Leonard-Barton (1995) refers to as 'creative abrasion' leading to creativity and innovation.

One use of such a method is to identify certain product ranges and their associated attributes. This provides knowledge for such activities as production restructuring or devising new marketing strategies. A more elaborate use is to build ladders of typical problems, their causes and possible solutions. Attributes of these are then identified using the repertory grid and matrix tools. Automatic translation of ladders and matrices produce various graphical representations. These can be used to summarize what causes and solutions apply to a particular problem.

*Rolls-Royce*

Rolls-Royce is a world-leading power-systems business, providing products and services to commercial and military customers in propulsion, electrical power and materials-handling markets around the world.

Alongside its involvement in the SPEDE project (covered below), Rolls-Royce has introduced a programme using PC-PACK to capture and disseminate best practice. The results are being included in their company 'Capability Intranet' for knowledge sharing both within and across departments. In line with the advice of Davenport and Prusak (1998), Rolls-Royce are gathering best practice in the context of a wider programme of knowledge mapping and transfer.

A key feature of this programme is that relatively novice users are applying the methods and tools of knowledge engineering for knowledge management. We see this as a vital aspect of a knowledge technology. Companies have such vast amounts of knowledge that knowledge management must be performed by all employees rather than a small group of in-house experts or consultants. Hence, the use of PC-PACK in Rolls-Royce is being performed by company employees as part of a three-month secondment to a special facility set up at the University of Nottingham. This secondment, which currently comprises 16 employees, includes an intensive two-week training programme, followed by a ten-week project. Such projects have focused on design and manufacturing and have included work in the following areas:

- capture and dissemination of knowledge required for the design of jet engine components, e.g. compressor blades and annulus;
- identification of best practices within manufacturing inspection, especially co-ordinate measuring machines;
- development of a materials knowledge base and advice system for designers (who typically have limited knowledge of the properties and uses of new types of materials);
- analysis of human resource issues within design, especially selection and training.

For an area with no previous knowledge on the Capability Intranet, a pair of trained employees usually produce around 350 quality web pages from scratch by the end of their ten-week project. Estimates by Rolls-Royce suggest that this is around five times more than would be produced with conventional methods. Not only that, but the knowledge gathered is seen as better structured, better validated and better focused on user requirements, such that knowledge retrieval and use are enhanced. Furthermore, there is more than a ten times reduction in the time needed from experts.

Based on the results, Rolls-Royce estimate that introduction of a knowledge technology based on PC-PACK can provide potential savings of several million pounds in quality documentation alone.

Results from this pilot study demonstrate that knowledge engineering techniques and tools are relatively easy to learn and apply by novice users within real knowledge management projects.

*The SPEDE project*

SPEDE is a large IMI-funded project involving both industry and academia. The industrial partners are Rover, Rolls-Royce and Computervision; the academic partners are the universities of Leeds, Nottingham and Warwick. The overall objective is to produce a set of widely applicable methods and software tools to assist and guide the business process engineer in the task of business process improvement (BPI); mainly BPR, but including other improvement approaches. For validation purposes, the focus is on the general applicability of these methods within the product introduction process at component level.

Many methods and tools are available that claim to support BPR and BPI. An analysis by Bach *et al.* (1996) reveals both the large number of methods in existence and the great extent to which they contradict one another. They conclude their examination with a list of requirements for the next generation of methods and tools. These include the need for an integrated set of tools which can support: (a) the reuse of existing in-house solutions, (b) the use of reference models of industry-wide best practice, and (c) the dissemination of company know-how. SPEDE aims to satisfy these requirements by having three main elements.

1. It provides knowledge acquisition tools that allow rapid elicitation of BPI requirements, structured acquisition of heterogeneous process information and validation of captured information. These tools aid the dissemination of company know-how by establishing links to corporate memory and knowledge management resources.
2. It establishes and makes use of a database of generic business processes to minimize the effort in the creation and use of business process models. This, combined with an ontology-based retrieval system, allows the reuse of existing in-house solutions, as well as making available reference models of industry-wide best practice.
3. It provides a demonstration environment that facilitates organizational change using process simulation and process enactment.

Another important aspect of SPEDE is that analyses of implemented solutions are used to modify and update the database of generic models, so that the database evolves to become a repository of organizational best practice.

The SPEDE tool set includes a number of PC-PACK tools, some of which have been modified and extended to be more useful for a BPI initiative. In particular, work has focused on the development of process modelling tools, grammars and ontologies. These have allowed creation of generic models of business processes and the design of an ontology-based retrieval system. Ongoing work is testing and refining the tools and methods at Rover and Rolls-Royce.

## Summary and concluding remarks

### *Summary*

We have attempted to convey our belief that principles, techniques and tools, developed for knowledge engineering, can be modified and extended to help organizations with knowledge management. There have been four strands to this argument:

1. With its basis in artificial intelligence, knowledge engineering can draw on a rich heritage from a number of fields such as philosophy and experimental psychology, from which it has inherited many ideas, and sound practical guidance.
2. Knowledge engineering has been dealing with knowledge for the past 15 years, from both a theoretical and practical perspective, and has built up a stock of principles, techniques and tools that have a good track record in a wide range of applications.
3. There is a substantial overlap between the concerns and problems encountered in both knowledge engineering and knowledge management because, in many areas, they are trying to do the same thing – use technology to ‘transport’ knowledge from an expert in a domain to those who are not experts.
4. A number of organizations (e.g. Andersen Consulting, Rolls-Royce and Unilever) are finding real and significant benefits in using the knowledge engineering techniques and tools for a number of different knowledge management activities such as brainstorming, identifying best practice, population of intranets, improving decision-making and process improvement.

### *Concluding remarks*

There are two important caveats we would like to mention. First, the techniques and tools described in this paper have only been applied in real knowledge management settings for the past couple of years, and though results look very promising, they have not been thoroughly tested with the empirical rigour that we would like to see. A full double-blind test of knowledge engineering technology against current knowledge management practices is yet to take place (although we plan to do this in the next year). Thus, claims such as a five-fold increase in efficiency over existing methods must be taken cautiously. In some ways, initial use of these new techniques and tools has been too successful. Consequently, we find we have to manage organization’s expectations and reiterate that knowledge engineering methods are not the panacea for all their knowledge management problems.

The second caveat is our belief that technology (any technology) *alone* will not allow an organization to manage its knowledge assets as effectively as possible. As Davenport and Prusak point out in their recent book *Working Knowledge,* knowledge technology must be supported by both the right culture and the right organizational structure if it is to work effectively. An organization might have the most efficient intranet in the world – easy to use, easy to understand, on everybody's desk, populated by all the useful knowledge that an employee needs – and yet it may still be overlooked if people are too stubborn, proud, cynical or demotivated to use it.

Some might say that cultural problems are insurmountable using knowledge technology. We disagree. We believe that with further modification and adaptation our techniques and tools can be used to capture the ways in which behavioural, cultural and organizational change takes place, and how it can be managed best. There is expertise involved in such processes, which can and should be captured and disseminated. The application of technology to capture knowledge about knowledge management might, we believe, herald a new era for both knowledge technology and organizational effectiveness.

## References


Bach, V., L. Brecht, T. Hess and H. Osterle (1996). *Enabling Systematic Business Change*. Verlag Vieweg, Wiesbaden.

Chi, M. H., R. Glaser and M. Farr (eds) (1988). *The Nature of Expertise*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Corbridge, C., G. Rugg, N. Major, M. Shadbolt and A. M. Burton (1994). 'Laddering: Technique and Tool Use in Knowledge Acquisition', *Journal of Knowledge Acquisition*, **6**, pp. 315–341.

Davenport, T. H. and L. Prusak (1998). *Working Knowledge: How Organizations Manage What They Know.* Harvard Business School Press, Boston, MA.

Hayes-Roth, F., D. A. Waterman and D. P. Lenat (eds) (1983). *Building Expert Systems*. Addison-Wesley, Reading, MA.

Hoffman, R., N. R. Shadbolt, A. M. Burton and G. Klein (1995). 'Eliciting Knowledge from Experts: A Methodological Analysis', *Organizational Behavior and Decision Processes*, **62**(2), pp. 129–158.

Kelly, G. A. (1955). *The Psychology of Personal Constructs*. Norton, New York.

Leonard-Barton, D. (1995). *Wellsprings of Knowledge*. Harvard Business School Press, Boston, MA.

Minsky, M. and S. Papert (1974). *Artificial Intelligence.* Condensed lectures, Oregon State System of Higher Education, Eugene.

Montero, L. and C. T. Scott (1998). *Improving the Quality of Component Business Systems with Knowledge Engineering.* To be presented at the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop.

Nonaka, I. and H. Takeuchi (1995). *The Knowledge-Creating Company.* Oxford University Press, New York.

Polanyi, M. (1966). *The Tacit Dimension.* Routledge & Kegan Paul, London.

Ramesh, B. and V. Dhar (1992). 'Supporting Systems-Development by Capturing Deliberations During Requirements Engineering', *IEEE Transactions on Software Engineering*, **18**(6), pp. 498–510.

Ryle, G. (1949). *The Concept of Mind.* Hutchinson, London.

Schreiber. A. Th., J. Akkermans, A. Anjewierden, R. De Hoog, N. Shadbolt, W. Van De Velde and B. Wielinga (forthcoming). *Engineering and Managing Knowledge*.

Shadbolt, N. R. and A. M. Burton (1995). 'Knowledge Elicitation: A Systematic Approach'. In: J. R. Wilson and E. Corlett (eds), *Evaluation of Human Work: A Practical Ergonomics Methodology,* (2nd revised edn). Taylor and Francis, London.

Shadbolt, N. R. and K. O'Hara (1997). 'Model-Based Expert Systems and the Explanation of Expertise'. In: P. J. Feltovich, K. Ford and R. Hoffman (eds), *Expertise in Context*. AAAI Press/MIT Press, Menlo Park, CA.

Shadbolt, N., E. Motta and A. Rouge (1993). 'Constructing Knowledge-Based Systems', *IEEE Software*, November, pp. 34–39.

Wielinga, B., J. Sandberg and G. Schreiber (1997). 'Methods and Techniques for Knowledge Management: What has Knowledge Engineering to Offer?', *Expert Systems with Applications*, **13**(1), pp. 73–84.