Chapter 2

Knowledge Management

Overview

Today there is a proliferation of information addressing the knowledge economy and the belief that the future of business success will be based on the ability to capture, manage, and leverage an organization's knowledge. What does this mean? How do you create an environment to capture and manage enterprise knowledge? More precisely, what is KM? Before we begin to construct a KM initiative, we must first agree on a definition. If you were to speak to ten different KM practitioners, you would probably receive ten different definitions. For us to move forward, we will use the following definition to set the framework for our continuing discussion about KM.

KM consists of methodology practices, new software systems, processes, and operating procedures that are developed to validate, evaluate, integrate, and disseminate information for users to make decisions and learn. Now that we have a definition of KM, what exactly are we managing? In other words, what is knowledge?

Let us start by distinguishing between data, information, and knowledge (see Figure 2.1). At the beginning of the spectrum, you have *data*. Data consists of random bits and pieces of something. This "something" can be numbers, text, video, or voice. On the other hand, information puts these random bits and pieces of "something" into a logical order that is meaningful to its user. The results of this logical order could be a report of some kind (e.g., a stock report for an investor, voice recording of a business meeting, a patient summary for a nurse, or a spreadsheet for an accountant).

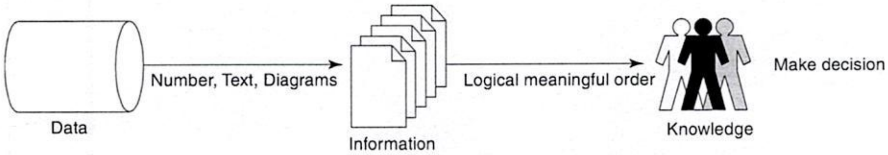

Figure 2.1 Data-Information-Knowledge

Furthermore, knowledge enables the user of information to make a decision or learn something from the information that has been presented. For instance, from a stock report, an investor can ascertain what stock she should buy or sell; a video may be delivering instructions about a procedure or process; and from a patient summary, a nurse may be able to determine when a certain medication should be administered to a patient.

Now that we have a clear picture of the evolution of knowledge, it is appropriate to continue with our understanding of KM. Remember our above-stated definition. With any definition, we must be aware that a narrow definition will tend to produce results that will lead to simple human resource policies and procedures leaving much of the value of KM unrealized. However, a definition that is too broad will be too abstract and lead to an unclear implementation of KM policies, practices, and procedures. Therefore, our definition reflects theories of KM that differentiate knowledge from information and integrate people with policies, practices, and procedures while allowing technology to aid in its implementation.

To give you a frame of reference, KM has connections with several established management strategies and practices. These practices include change management, risk management, and business process reengineering. There is a common thread between these practices, which recognizes that knowledge is a corporate asset and organizations need strategies, policies, practices, and tools to manage these assets. Discussions about KM always lead to discussions of intellectual capital both tacit and explicit. This has brought about the implementation of technology-driven methods for accessing, controlling, and delivering information that the corporate culture can transform into knowledge. This enables the corporate culture to create new knowledge value while leveraging existing knowledge. The concept of knowledge value will be discussed in further detail later in this chapter.

Intellectual capital consists of three major components:

1. Human resources — consist of the employee's collective experience, skills, and expertise of how the organization operates and the uniqueness of how it operates vs. its competitors.
2. Intellectual assets — consist of any piece of knowledge that becomes defined, usually by writing it down or inputting it into a

computer, such as inventions, design approaches, and computer programs. Intellectual assets represent the source of innovations, which firms commercialize.

3. Intellectual property — consists of intellectual assets, which can be legally protected. This includes patents, copyrights, trademarks, and trade secrets.

Intellectual capital takes two forms — explicit and tacit. Explicit knowledge is knowledge contained in documents, computer programs, databases, etc., and can be articulated easily; tacit knowledge resides in the minds of individuals. It is the tacit knowledge that never is quantified into a manual or other accessible form, but resides in the minds of the people who have worked with and developed that information. The problem is that when someone leaves the company or for a different assignment within the company, this intellectual capital leaves also. To capture this tacit knowledge, knowledge acquisition techniques must be utilized.

Knowledge Value

Historically, KM programs can take a considerable amount of time to show results or visible return on investment (ROI) for an organization. However, there is an approach in which to estimate the value of the intangible benefits of KM. The Knowledge Value Equation (KVE) simply states that the value created from managing knowledge is a function of the costs, benefits, and risks of the KM initiative. Thus, mathematically stated:

> KM value = F (cost, benefit, risk), which equals total discounted cash flow (DCF) created over the life of the KM investment.[1]

This formula attempts to quantify the intangible impacts of KM, relating it back to cash flow. This includes improved problem solving, enhanced creativity, and improved relationships with customers.

KM projects produce a stream of benefits over time. This is why we use the KM Value Model. This will enable KM projects to be evaluated based on a series or stream of cash flows. In doing this, we must understand the concepts of time value of money and DCF. To take the intangible aspects of KM and turn them into a series of cash flows that can be discounted over time, we must first start with ways to increase DCF. The following list represents several ways in which to do this:

- Increase revenue by selling more products or by introducing new products and services
- Lower expenses by decreasing quality, transactional, administrative, production, and other costs

- Improve margins by increasing operational and economic efficiency to improve profit
- Lower taxes through smart strategies that minimize the tax liabilities of the firm
- Lower capital requirements by decreasing the amount of capital needed by regulation to run the business
- Lower costs of capital by decreasing the cost of loans, equity, and other forms of financing[2]

To model the benefits of KM as cash flows we must tie them back to one or more of the ways to increase DCF as mentioned above. We must also be aware of how KM projects transform business processes and practices to improve operations and generate DCF.

Knowledge-Value Tree

The knowledge-value tree is a treelike graphical representation that is used to make the connection between knowledge and value more visible. The mapping is as follows:

KM functionality → business transformation → DCF → value[3]

To depict this connection we have constructed the knowledge-value tree of XYZ Shipping Company (see Figure 2.2).

There is a connection between new KM functionality and business processes and individual practices:

KM functionality → processes and practices → change in business metrics

For example, review the knowledge-value tree of our fictitious XYZ Shipping Company below.

There is a link between the change in business metrics and one or more aspects of DCF. A change in business metrics will have an effect on one or more of the drivers of DCF. The presentation of knowledge-value trees has to be convincing to business stakeholders and senior management. This has to be positioned in order to show how we would achieve a ROI on the intangible benefits of a KM investment (i.e., KM functionality → processes and practices → business metrics → DCF drivers → value).

Building knowledge-value trees tends to get complex, and they are difficult to read. However, a robust theory of business knowledge provides the necessary drivers to demonstrate the relationship between KM

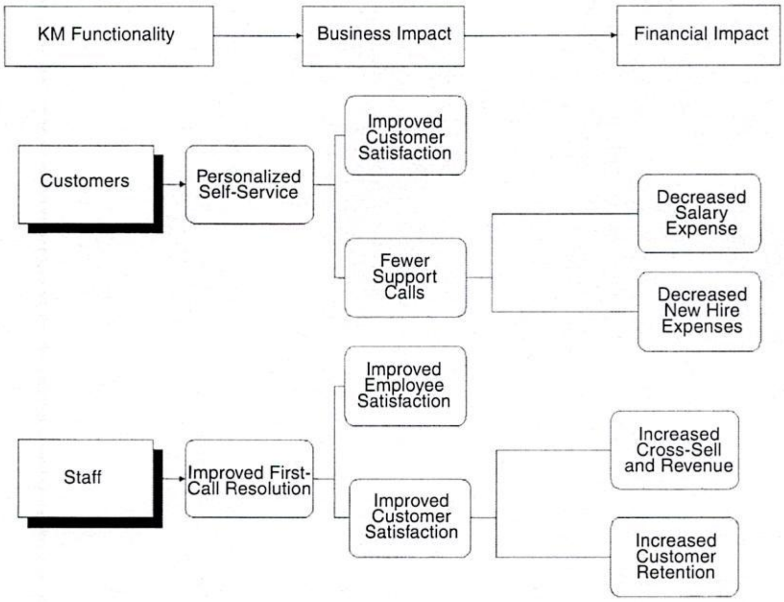

Figure 2.2 XYZ Shipping Company — Knowledge-Value Tree

functionality, business practices, and the creation of value for an organization. Knowledge-value trees also provide a mechanism for determining where and how economic value is being created. Discovering these knowledge-value drivers is one of the central tasks of KM.

Knowledge-value trees and the calculations associated with them involve some assumptions. To reduce the risks associated with these assumptions we must consider the following:

- Use financial reports and other summary documents to make informed judgments.
- Review all assumptions with the appropriate business experts.
- Quantify risks associated with your assumptions by determining how a change in the assumption influences the total DCF.
- Use computations, rather than absolute evaluation by developing a set of scenarios that look at a range of assumptions.
- Use models to frame assumptions whenever possible. The assumptions that go into knowledge-value trees should be based on the best business data and experience available.

Developing a knowledge-value tree provides a way to see and quantify key risks and refine theories to drive KM initiatives.

Why manage knowledge? We manage knowledge because organizations compete based on what they know. We manage knowledge because the products and services that are produced are increasingly complex, commanding a significant investment in information and knowledge. Finally, we manage knowledge because there is a need to facilitate corporate learning through knowledge sharing. The result of managing knowledge has presented the opportunity for achieving significant improvements in human performance and competitive advantage.

Knowledge Management Systems

A knowledge management system addresses the needs of an organization that desires not to reinvent knowledge and not to spend excess time locating difficult-to-find knowledge; an organization that desires to successfully absorb and use the growing volumes of new knowledge flowing into and out of that organization every day. All of which cost millions of dollars annually. KM also combines cultural and process changes along with enabling technology to achieve bottom-line results.

KMS components consist of customer relationship management (CRM), document management, knowledge acquisition, collaboration, workflow, and E-learning (see Figure 2.3).

Knowledge Acquisition

Knowledge acquisition is a key component of the KMS architecture as shown in Figure 2.3. Knowledge acquisition includes the elicitation, collection,

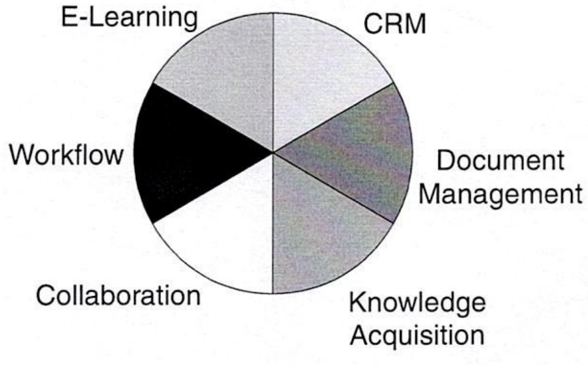

Figure 2.3 Knowledge Management System Components

analysis, modeling, and validation of knowledge for knowledge engineering and KM initiatives.[4]

Any application constructed will depend directly on the quality of the knowledge acquired. During this process, it is vital to determine where in the organization the knowledge exists, how to capture it, and how to disseminate this knowledge throughout the enterprise. The approach to knowledge capture may take on many forms. Developing a framework for knowledge acquisition will allow for a consistent method for capturing the knowledge of a particular enterprise, organization, or human (domain) expert.

Knowledge acquisition is the most expensive task in building and maintaining knowledge in a KMS. Although there are many techniques and methodologies to help the knowledge engineer elicit knowledge, none of these techniques incorporates the use of UML as a construct. UML provides a direct modeling medium that will give this framework a standard notation in which many readers are familiar.

The use of knowledge-based systems has expanded enormously in recent years with applications ranging from medicine to engineering and aerospace. Knowledge-based systems are software systems that apply advanced technical knowledge, or expertise, to problem solving. Applications of knowledge-based systems typically replicate some form of human expertise to support the solution of problems in a target domain. Knowledge-based systems are being called on to provide automation in increasingly complex domains. As domain complexity increases, so does the complexity and difficulty of building a knowledge-based system.

Knowledge acquisition is the process of acquiring, organizing, and studying knowledge about a certain domain. The object of knowledge acquisition is to identify the rules, policies, procedures, and practices that are reflective about the domain under development. This is an identification of requirements, which provides an in-depth understanding of the domain area.

To elicit or acquire knowledge about a domain, the knowledge engineer interviews the domain experts or subject matter experts (SMEs). A knowledge engineer is a "software development professional who applies engineering methodologies to the analysis of human knowledge."[4] A domain expert is a professional within an industry who has a core competence in the policies, procedures, and practices followed to solve problems or to perform a task within the domain. The objectives of the interviews are to uncover concepts and problem-solving methods used by the domain expert. Each session will progressively drill deeper and deeper into the problem area and explore increasingly more detailed information.

This should come as no surprise to any software professional — knowledge acquisition is a form of requirements analysis, which has long been known to play a critical role in building quality software. Like other forms of requirement analysis, knowledge acquisition is one of the most difficult and error-prone tasks encountered when implementing a system. The cost and performance of the application depends directly on the quality of the knowledge acquired.

To elicit knowledge from an expert, the traditional approach to knowledge acquisition is that, regardless of the variation used, it is costly because at least two (typically) expensive people are involved, i.e., the domain expert and the knowledge engineer.

The second thing to note is that the methods are error prone. Surprisingly, people cannot easily say what it is that they do in a manner that can be understood by others. This is mostly because skills are usually learned through apprentice-style learning, and the small, faltering steps required by the expert during initial learning have long since become embedded in longer phases of automated behavior, and the constituent steps are no longer readily accessible. Therefore, interpretations of what experts do are often faulty and incomplete and sometimes based on rationalizations by the experts of what they think they are doing rather than what they actually are doing. These misinterpretations are often easily committed by well-trained knowledge engineers, let alone less well-trained practitioners.

The third thing to note about the traditional approach to knowledge acquisition is that it is time consuming because errors, gaps, and inconsistencies may be difficult to discover, requiring many interactions between the domain (subject matter) experts and knowledge engineers to debug a field-ready application.

Clearly, costs must be reduced, errors eliminated, and development time shortened. An approach to solving these issues is to augment the knowledge engineer with a framework for knowledge elicitation that will provide a guide for consistent method in which to capture the knowledge of a particular enterprise, organization, or human (domain) expert. Use of a framework will have a significant effect on the quality and cost of these applications.

Incorporating UML as a direct modeling medium from which the domain expert can verify the representation of knowledge will ensure that the expert system reflects the domain expert's view of the domain instead of the intermediary's view.

The use of UML has an ancillary benefit in training knowledge engineers. There are many techniques and methodologies to help the knowledge engineers — but related to each of them there are various sets of skills that are required by the knowledge engineer. These may include exhibiting the skills of a cognitive psychologist, a communication expert, a programmer, and a domain expert. There could be situations in which the knowledge engineer may be required to play more than one role. Because this publication will present standard representations, we can eliminate much of the special training required by the knowledge engineer.

Knowledge Acquisition Process

Acquiring knowledge about a certain domain is inherently an exploratory effort. Many in management lack the understanding of the tasks and problems to be solved by capturing knowledge about the domain and automating this knowledge in the form of a knowledge base (expert) system. These systems serve as the cornerstone of the KMS (see Figure 2.3). Therefore, a general understanding or framework has to be established and then used as a guide for further probing for additional information.

An important part of the process concerning knowledge acquisition is identifying the sources of where to uncover and identify the rules, policies, procedures, and practices applied to the domain. Such sources include SMEs where tacit knowledge is uncovered, and literature, application code, and database-stored procedures where explicit knowledge is uncovered. These sources and others will be examined in later chapters. However, it is important to look at the role SMEs have in the knowledge acquisition process.

Subject Matter Experts

The SME can take on many forms. These individuals range from the business analyst, end user, or business manager to an experienced professional with several years working in the specific domain under development. Analysts will often view a problem from a low-level detail-oriented perspective. The end user will often view the problem from a high-level, considering the major issues. Working with the end user early in the development effort is of particular value when the knowledge engineer lacks an initial general understanding of the problem area. The end user is also valuable later in the development cycle for identifying shortcomings or oversights in the gathering of knowledge about the domain area.

One effective technique is to involve multiple SMEs, even though this approach tends to lengthen the acquisition process. It offers several distinct advantages. No one person has a perfect comprehensive competency in any problem area. Using multiple SMEs combined with several analysts and end users allows the knowledge engineer to identify conflicting information, errors, contradictions, etc., more readily. These multiple sources also allow the knowledge engineer to resolve issues by consensus.

Most development efforts that choose to use multiple SMEs will identify one to be the primary resource. This approach helps to reduce confusion and resolve conflicts caused when SMEs provide conflicting information. More often, multiple SMEs exhibit expertise in separate areas within the domain; in such cases, expanding the number of SMEs is a way of distributing the sources of information within the domain.

Knowledge Acquisition Tasks

The tasks involved in acquiring knowledge begin with identifying a suitable domain area for a knowledge-based system or KMS implementation. The next step involves collecting information about the domain area, which is regarded as the most difficult task of the knowledge engineer.

Collection tasks in the early cycles of the acquisition process focus on attaining a basic understanding of the problem area. As the cycles progress, increasingly more detailed information about the specifics of the domain area is obtained. This is an iterative style of collecting information and is in line with the latest and more efficient ways of software and process engineering.

The task of interpretation follows collection. This task involves a review of the collected information and the identification of key policies, procedures, practices, and processes. Because the knowledge gathered early on is general, it is the job of the knowledge engineer to focus on defining the specification of the overall domain area. This involves informal review of all the materials collected. This review is to establish the domain's goals, constraints, and scope. During further iterations, more formal methods (e.g., frameworks) should be used to interpret, model, and construct the knowledge base or bases of the domain area.

From tasks performed during interpretation, key pieces of identified information will provide insight into forming theories on the organization and classification of the knowledge. During early iterations, the knowledge engineer identifies the important concepts and forms a general understanding of the policies, procedures, practices, and processes as well as how they are used to solve issues within the domain area.

Following the completion of the collection, interpretation, and analysis tasks, the knowledge engineer will have formulated an understanding of the domain area that can aid in further investigations. Thus far, the knowledge engineer has been exposed to new concepts and problem-solving strategies that will need further exploration. This information will contribute to guiding the design and establishing the goals and agenda for collecting additional information during future sessions.

An Iterative Approach

As mentioned earlier, the tasks of collection, interpretation, and analysis are done in an iterative fashion. This is done because the knowledge acquisition process is repeated a number of times, each time building on knowledge gained from the previous session. The collection task of the acquisition cycle requires only a short time to complete relative to the timeframe of the entire cycle. Most interview sessions with SMEs will last more than a few hours in length. However, the other phases of the cycle are more time-consuming efforts.

Roles of Knowledge Acquisition Team Members

Forming a team of both talented and cooperative individuals is an important initial step in the knowledge-gathering process. Each of the team members plays an important role in the process and provides specific and necessary contributions. In most cases, the acquisition process involves at least two key individuals or skill sets:

1. SME — Knowledge of business, customers, and domain
2. Knowledge engineer — Knowledge of acquisition methods, practices, and frameworks

Larger projects will require the participation of additional personnel. Teams may be broken along lines of functional system components or specialization within certain aspects of a complex domain.

Role of the Subject Matter Expert

It is recommended that the SME be involved in the project throughout the project's life cycle. This is not only a characteristic of building knowledge-based systems or KMSs, but also a characteristic of any software application development. Knowledge engineers recognize the importance of this teaming with the SMEs and encourage this partnership, incorporating joint application development (JAD) and joint application requirements (JAR) sessions as an essential aspect of a successful system development effort.

SMEs can help the process extensively during acquisition by providing an initial high-level insight into the domain area. They can help to define the system's operational requirements as well as provide useful feedback throughout the development effort. Some of the other roles that the SMEs can fulfill are the following:

- Provide an overview of expected functionality
- Define system input and output requirements
- Define explanation facility requirements
- Define operational requirements
- Assist in testing and evaluation
- Provide feedback on the validity of the knowledge captured
- Assist in the interpretation and analysis of policies, practices, procedures, and processes

Role of the Knowledge Engineer

Knowledge engineering is an interdisciplinary effort. The knowledge engineer must perform a range of tasks, beginning with the introduction of knowledge-based system or KMS technology to all individuals involved and ending with the integration of the knowledge components into the application under development.

The major tasks of the knowledge engineer are the following:

- Collecting information about the domain area
- Interpreting the information about the domain area
- Analyzing the information about the domain area
- Coordinating efforts of the team members
- Assigning specific tasks
- Arranging JAD and JAR sessions with the SMEs

This iterative process is repeated with each new knowledge acquisition session, because the issues to be pursued in any one session depend on what was learned in prior sessions. The tasks performed by the knowledge engineer during the knowledge-gathering process require both technical and personal skills. The knowledge engineer must have the technical skills for interpreting and analyzing the collected body of domain knowledge. On the personal side, the knowledge engineer must have excellent communication skills, sensitivity to the interest and politics of the customer's working environment, and general project management skills.

Issues in Knowledge Acquisition

Some of the most important issues in knowledge acquisition center on the following issues:

- Most knowledge is in the heads of experts (tacit knowledge).
- The experts have vast amounts of knowledge and do not realize all that they know, which makes it difficult to describe and capture.
- The experts are usually too busy to enable someone to gain from their knowledge.
- There are usually multiple experts in a particular domain, which makes it difficult to validate the knowledge captured.
- Knowledge has an expiration date: knowledge about a certain domain may become obsolete over time as new techniques are discovered or technology is developed.

Knowledge Acquisition Techniques

Many techniques have been developed to help elicit knowledge from an expert. These are referred to as knowledge elicitation or knowledge acquisition (KA) techniques. The term *KA techniques* is commonly used.

The following list gives a brief introduction to the types of techniques used for acquiring, analyzing, and modeling knowledge:

- Protocol-generation techniques — Include various types of interviews (unstructured, semistructured, and structured), reporting techniques (such as self-report and shadowing), and observational techniques.
- Protocol-analysis techniques — Used with transcripts of interviews or other text-based information to identify various types of knowledge, such as goals, decisions, relationships, and attributes. This acts as a bridge between the use of protocol-based techniques and knowledge modeling techniques.
- Hierarchy-generation techniques — Techniques such as laddering are used to build taxonomies or other hierarchical structures such as goal trees and decision networks. Laddering lends itself well to UML notation, a technique we explore further in Chapter 9 and Chapter 10.
- Matrix-based techniques — Involve the construction of grids indicating such things as problems encountered against possible solutions. Important types include the use of frames for representing the properties of concepts and the repertory grid techniques used to elicit, rate, analyze, and categorize the properties of concepts.
- Sorting techniques — Used for capturing the way people compare and order concepts. Sorting techniques can lead to the revelation of knowledge about classes, properties, and priorities.
- Limited-information and constraint-processing tasks — Limit the time or information available to the expert when performing tasks.

For instance, the 20-questions technique provides an efficient way of accessing a domain's key information in a prioritized order.

- Diagram-based techniques — Include the generation and use of concept maps, state transition networks, event diagrams, and process maps. The use of these is particularly important in capturing the "what, how, when, who, and why" of tasks and events. The diagram-based techniques are most amendable to the use of UML in its implementation, which we explore further in Chapter 9 and Chapter 10.

Differential Access Hypothesis

Why have so many techniques? The answer lies in the fact that there are many different types of knowledge possessed by experts, and different techniques are required to access the different types of knowledge. This is referred to as the Differential Access Hypothesis and has been shown experimentally to have supporting evidence.

Comparison of Knowledge Acquisition Techniques

Figure 2.4 — Knowledge Acquisition Techniques presents the various techniques described above and shows the types of knowledge they mainly are aimed at eliciting. The vertical axis of the figure represents the dimension from object knowledge to process knowledge; the horizontal axis represents the dimension from explicit knowledge to tacit knowledge.

Typical Use of Knowledge Acquisition Techniques

How and when should the techniques described above be used in a knowledge acquisition project? To illustrate the general process, a simple method will be described. This method starts with the use of natural techniques and then moves to using more contrived techniques. It is summarized as follows.

Conduct an initial interview with the expert to (a) scope what knowledge is to be acquired, (b) determine to what purpose the knowledge is to be put, (c) gain some understanding of key terminology, and (d) build a rapport with the expert. This interview (as with all sessions with experts) is recorded on either audiotape or videotape.

Transcribe the initial interview and analyze the resulting protocol. Create a concept ladder of the resulting knowledge to provide a broad representation of the knowledge in the domain. Use the ladder to produce a set of questions that cover the essential issues across the domain and

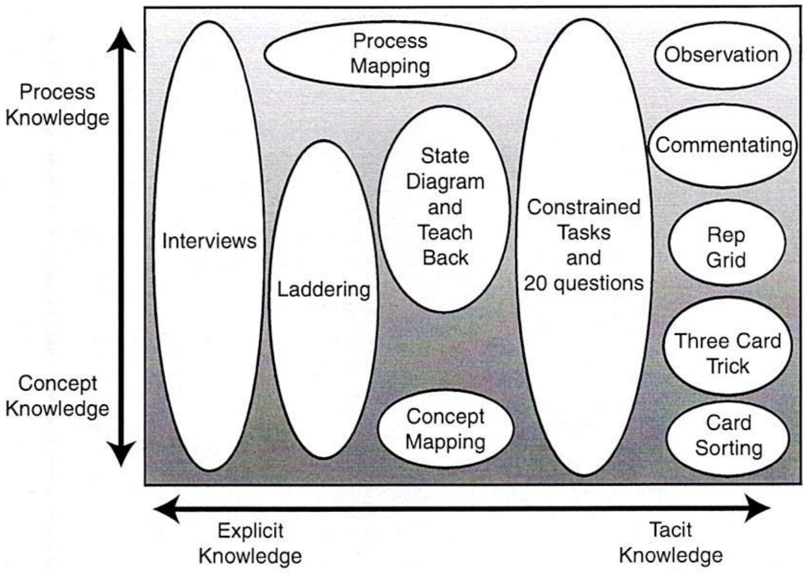

Figure 2.4 Knowledge Acquisition Techniques (Epistemics Web site.[4] Used with permission.)

that serve the goals of the knowledge acquisition project. Conduct a semistructured interview with the expert using the preprepared questions to provide structure and focus.

Transcribe the semistructured interview and analyze the resulting protocol for the knowledge types present. Typically, these would be concepts, attributes, values, relationships, tasks, and rules. Represent these knowledge elements using the most appropriate knowledge models, e.g., ladders, grids, network diagrams, hypertext. In addition, document anecdotes, illustrations, and explanations in a structured manner using hypertext and template headings.

Use the resulting knowledge models and structured text with contrived techniques such as laddering, think-aloud problem solving, 20 questions, and repertory grid to allow the expert to modify and expand on the knowledge already captured. Repeat the analysis, model building, and acquisition sessions until the experts and knowledge engineer are happy that the goals of the project have been realized.

Validate the knowledge acquired with other experts and make modifications where necessary.

This is a very brief description of what happens. It does not assume 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. Techniques have been developed to assist in this, such as the use of ontologies and problem-solving models. These provide generic knowledge to suggest ideas to the expert such as general classes of objects in the domain and general ways in which tasks are performed. This reuse of knowledge is the essence of making the knowledge acquisition process as efficient and effective as possible. This is an evolving process. Hence, as more knowledge is gathered and abstracted to produce generic knowledge, the whole process becomes more efficient. In practice, knowledge engineers often mix this theory-driven (top-down) approach with a data-driven (bottom-up) approach.

Recent Developments

A number of recent developments are continuing to improve the efficiency of the knowledge acquisition process. Four of these developments are examined below.

First, methodologies have been introduced that provide frameworks and generic knowledge to help guide knowledge acquisition activities and to ensure that the development of each expert system is performed in an efficient manner. A leading methodology is CommonKADS. At the heart of CommonKADS is the notion that knowledge engineering projects should be model driven. At the level of project management, CommonKADS advises the use of six high-level models:

1. Organization model
2. Task model
3. Agent model
4. Expertise model
5. Communications model
6. Design model

To aid development of these models, a number of generic models of problem-solving activities are included. Each of these generic models describes the roles that knowledge plays in the tasks, hence providing guidance on what types of knowledge to focus on. As a project proceeds, CommonKADS follows a spiral approach to system development, such that phases of reviewing, risk assessment, planning, and monitoring are visited and revisited. This provides for rapid prototyping of the system, such that risk is managed and there is more flexibility in dealing with uncertainty and change.

A second important development is the creation and use of ontologies. Although there is a lack of unanimity in the exact definition of the term *ontology*, it is generally regarded as a formalized representation of the knowledge in a domain taken from a particular perspective or conceptualization. The main use of ontology is to share and communicate knowledge, both between people and between computer systems. A number of generic ontologies have been constructed, each having application across a number of domains, which enables the reuse of knowledge. In this way, a project need not start with a blank sheet of paper, but with a number of skeletal frameworks that can act as predefined structures for the knowledge being acquired. As with the problem-solving models of CommonKADS, ontologies also provide guidance to the knowledge engineer regarding the types of knowledge to be investigated.

A third development has been an increasing use of software tools to aid the acquisition process. Software packages, such as PCPACK4 by Epistemics, contain a number of tools to help the knowledge engineer analyze, structure, and store the knowledge required. The use of various modeling tools and a central database of knowledge can provide various representational views of the domain. Software tools can also force good knowledge engineering discipline on the user, so that even novice practitioners can perform knowledge acquisition projects. Software storage and indexing systems can also facilitate the reuse and transfer of knowledge from project to project. More recently, software systems that make use of generic ontologies are under development to provide for automatic analysis and structuring of knowledge.

A fourth recent development is the use of knowledge engineering principles and techniques in contexts other than the development of expert systems. A notable use of the technology in another field is as an aid to KM and construction of KMSs.

This approach has been a major influence in the past few years as companies recognize the vital need to manage their knowledge assets. A number of principles and techniques from knowledge engineering have been successfully transferred to aid in KM initiatives, such as the construction of Web sites for company intranet systems. This is an important precedent for the aim of this thesis to apply practices from knowledge engineering to the realm of personal knowledge.

What Is Knowledge?

To develop a framework to capture knowledge we must first know what knowledge is. Although we have given our definition of knowledge, which was established earlier in this chapter — "knowledge enables the user of information to make a decision or learn something from the information, which has been presented" — it really depends on the person's perspective. To some, knowledge is a commodity; others consider it a learning tool, and still others see it as a set of best practices.

Notes

1. Clare, M. Solving the Knowledge-Value Equation (Part One), *Knowledge Management Review 5* (2), May/June 2002.
2. Ibid.
3. Ibid.
4. Epistemics. Information on knowledge acquisition. Available online: http://www.epistemics.co.uk/Notes/63-0-0.htm (Accessed November 2002).