Knowledge-based knowledge management[1]

Rosina Weber and Randy Kaplan

College of Information Science and Technology, Drexel University, Philadelphia, PA 19104, USA

Abstract. Knowledge-based knowledge management (KBKM) focuses on applications of knowledge-based systems (KBS) tailored to knowledge management (KM) problems. Although KM is primarily concerned with how people and organizations utilize their knowledge assets, one key to doing so efficiently is to employ technology to facilitate the KM process. One particular kind of technology has shown itself to be extremely useful in this context – specifically the technology of knowledge-based systems. In this chapter we discuss the relationship of KM to knowledge-based technology and provide an exposition of three fundamental knowledge-based methodologies that can facilitate knowledge management – expert systems, case-based reasoning and ontologies.

Introduction

Knowledge-based knowledge management (KBKM) focuses on applications of knowledge-based systems (KBS) tailored to knowledge management (KM) problems. The term was first coined in [[1]] to express the use of KBS to enable knowledge management. KM practitioners and research scientists have been implementing various frameworks to address pragmatic KM problems reusing decades of technology developed for knowledge-based systems. Therefore, when we talk about knowledge-based knowledge management, we talk about the overlap of knowledge-based systems and knowledge management.

$$\boxed{\text{KBKM} = \text{KBS} \cap \text{KM}}$$

It is easy to understand the scope of KM by focusing on a knowledge process that collects, stores and reuses knowledge leveraging it and making organizational the knowledge that once was individual. This knowledge process supports organizational goals by controlling the collection, storage, and use of knowledge. Accordingly, KM applications can be envisioned along the dimensions of a knowledge

[1] In Innovations in Knowledge Engineering, Editors: Ravi Jain, Ajith Abraham, Colette Faucher and Berend Jan van der Zwaag. International Series on Advanced Intelligence. Volume 4, July 2003, pp. 151-172. Adelaide:Advanced Knowledge International .

process that fundamentally performs knowledge tasks to support, steer and control organizational goals.

Expert systems (ES), case-based reasoning (CBR), and ontologies are examples of relevant knowledge-based methodologies that have much to contribute to KM systems because they manipulate knowledge to implement various tasks. Although KM practitioners frequently comment that KM is not a technology problem, it is also the case that most KM solutions include an element of technology. It would seem that using technologies for collecting, storing, and reusing knowledge would be critical to consider in any KM effort. In the remainder of this chapter we substantiate this claim describing knowledge-based knowledge management applications and some recent developments of three representative knowledge-based methodologies: expert systems, case-based reasoning, and ontologies.

Table 1. Knowledge-based knowledge management systems

	Share knowledge	Distribute knowledge	Capture, codify knowledge	Create knowledge
Traditional Systems	Email, Group Collaboration, Discussion Groups, Peer to peer technology, Intranet portals	Word processing, desktop publishing, repository-based text databases	All systems that codify knowledge are knowledge-based	Brainstorming software, mind mapping, statistical analysis
Knowledge-based systems	Ontology-based systems	Case-based lessons-learned and best-practices systems	All knowledge-based systems codify knowledge; knowledge acquisition and coding methods	Knowledge discovery systems, ES for knowledge creation; creativity systems

We make a distinction between KBKM applications and applications that support KM but are not necessarily knowledge-based in **Table 1**. With these systems, we illustrate our view of the spectrum of KM applications and the distinction from the perspective given in [[2]], which describes information systems that support knowledge management.

To begin our discussion of knowledge-based knowledge management we will propose a knowledge process in which to consider the respective knowledge-based methodologies. After establishing this framework in the next section we will describe each of the technologies and give examples of the ways they support knowledge tasks. We finalize with a discussion of a case study on expert systems and business rules and conclude by illuminating some of the open issues and problems with knowledge-based technologies for KM.

Knowledge processes and knowledge tasks

Knowledge management refers to knowledge processes managed within an organization. Therefore, it is natural that the understanding of a knowledge process differs in different organizations; e.g., a knowledge process in one organization has a different set of tasks than in another. Sometimes these knowledge processes differ on the surface, though their conceptual model is the same.

Table 2. Knowledge processes adapted from [[3]]

Knowledge process by	Knowledge tasks
Liebowitz	1. Transform information to knowledge 2. Identify and verify knowledge 3. Capture and secure knowledge 4. Organize knowledge 5. Retrieve and apply knowledge 6. Combine knowledge 7. Create knowledge 8. Distribute/sell knowledge
Wiig	1. Creation and sourcing 2. Compilation and transformation 3. Dissemination 4. Application and value realization
van der Spek	1. Developing new knowledge 2. Securing new and existing knowledge 3. Distributing knowledge 4. Combining available knowledge
Ruggles	1. Generation consisting of creation, acquisition, synthesis, fusion, adaptation 2. Codification consisting of capture and representation 3. Transfer
Staab et al.	1. Creation and import 2. Capture 3. Retrieve/Access 4. Use
Weber & Aha	1. Collect 2. Verify 3. Store 4. Disseminate 5. Reuse

Table 2 illustrates a selection of these processes adapted from Liebowitz [[3]]. We added two more instances of knowledge processes, from Staab et al. [[4]] and Weber and Aha [[5]], to the original table from reference [[3]].

The underlying conceptual cycle we envisage in all these instances can be summarized in the knowledge process laid out in **Fig. 1**. This knowledge process bears marked similarity with each of these sample processes and consists of tasks

create, understand, distribute and reuse. We now describe the steps of this knowledge process in detail.
Create. The creation process focuses on how knowledge comes into existence in processes of organizations and individuals.
Understand. Knowledge understanding comprises verification, representation, synthesis, adaptation, storage and organization of the new knowledge with knowledge existing within the organization. For knowledge to be utilized by others in an organization, it must be linked to existing knowledge in the organization in a format that is accepted and understandable by the organization.

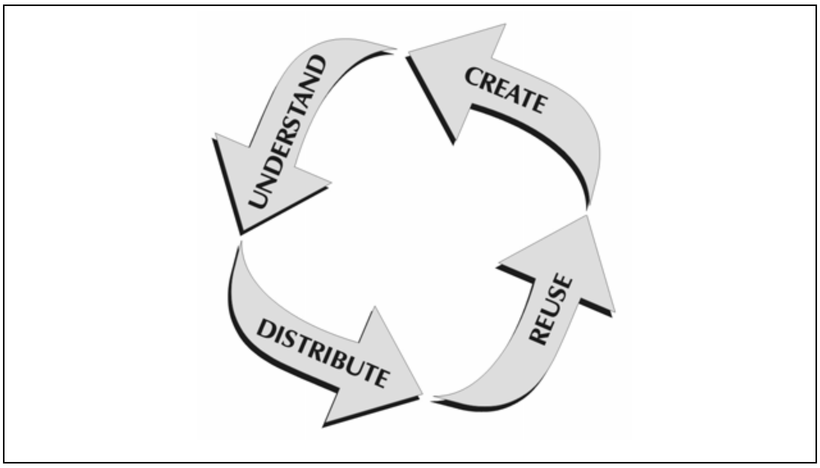

Fig. 1. Conceptual knowledge process

Distribute. Knowledge distribution embodies different dissemination methods (e.g., active, passive [[6]]). Once knowledge is captured and understood, it has to be available for access by individuals in the organization.
Reuse. Knowledge reuse consists of applying knowledge to an organizational process. It is not sufficient to make knowledge available; the reuse step is responsible for promoting reuse by encouraging users to apply available knowledge. This may result in the creation of new knowledge, which in turn begins the knowledge process again.

One may note that such a process could equally be applied to personal and organizational knowledge and to both systems that employ technology and those that do not. With respect to the processes described in **Table 2**, the one given on the last row ([[5]]) is an example of a process that is supported by technology but its collection is not knowledge-based. We note that when it happens, these processes tend to include knowledge verification, which is part of understanding, as an independent process. When humans perform knowledge processes, understanding complements creation, as should a *genuine knowledge process*. The requirement to implement a genuine knowledge process that performs understanding as a complement of creation is that it is supported by knowledge-based technologies.

It is difficult to distinguish a knowledge process from KM and that may be the reason why there are so many definitions on KM as a discipline. On the one hand we represent a knowledge process as a cycle – one in which KM manifests itself but can we also say that each of the tasks that are part of a knowledge cycle represent a kind of KM? Our observation is that we can examine a cycle as the knowledge process for KM or we can examine its parts as types of KM. This then gives us a context and framework in which to consider the technologies we will describe in the next sections. We can describe what role they play in the knowledge process and we can also describe how they support the tasks of the knowledge process (another form of KM).

Expert Systems

An expert system can be defined as "*an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions* [[7]]."

Over time, a human being in the process of becoming an expert will gain significant experience in their field of expertise. This experience is in turn used to solve problems that the expert encounters. Expert systems represent one way that expertise can be captured, coded, and reused.

Fundamentally, an expert system consists of some representation of expertise, some representation of a problem to be solved, and some mechanism to apply the expertise to a problem. Although expertise may be represented in various forms, one common representation for expertise is in the form of rules. The representation of an expert's knowledge would be a collection of rules that were derived from the expert.

A rule consists of two parts: an antecedent and a consequent. The rule antecedent consists of one or more conditions that specify when and where to apply the rule. If the conditions of the rule are met, then the second part of a rule – the consequent – specifies the actions to take when the conditions of the rule are met.

For example, we might consider the following rule for diagnosing the condition of an automobile: If the automobile has fuel, and the key is in the ignition, and the key is turned to the start position, and the engine does not start, then check to see if the engine turns over. This rule is applied when the automobile has fuel, the key has been engaged, and the engine does not start. Its action suggests further investigation required to diagnose the problem.

The rules that are encoded as expertise in an expert system are contained in a knowledge base – the repository of knowledge for the expert system. The mechanism that chooses rules to see if they can be used in a problem is called an inference engine. The inference engine checks antecedents of rules, and based on their values performs the actions specified in the consequents of the rules. To maintain the state of a problem being solved, the inference engine uses a special structure to store the state of problem solution. The structure is called short-term memory. A schematic of a typical rule-based expert system is shown in **Fig. 2**.

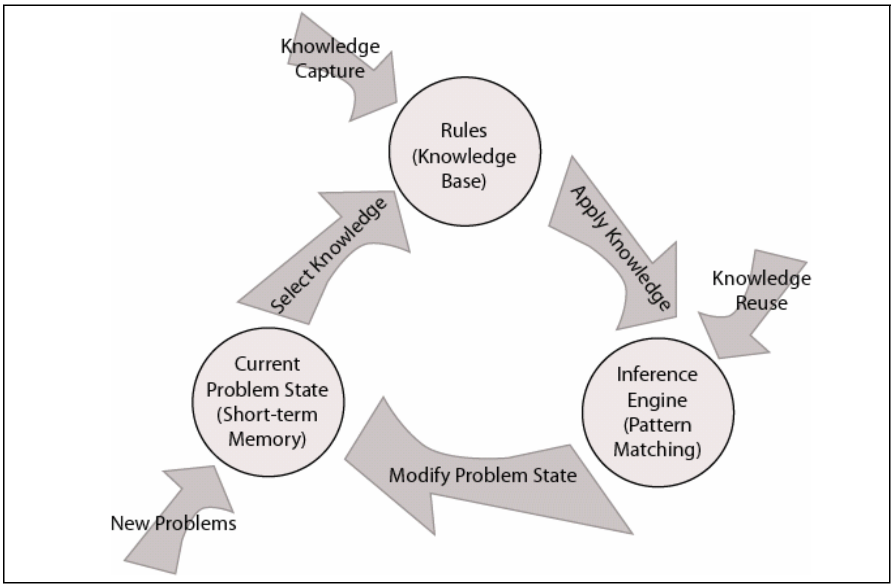

Fig. 2. Expert systems cycle

Rules are scanned from the knowledge base to determine which apply to the current problem state in short-term memory. When the inference engine identifies a rule, the actions of that rule are carried out, which may result in a change to the problem state in short-term memory. The process repeats itself until it solves the problem, or no condition can be fulfilled or the expert system is explicitly stopped.

This description in **Fig. 2** represents an expert system in its most basic sense – as a system that selects rules and applies these rules to solve a problem. In addition to the basic mechanisms provided, expert systems use various strategies to select rules to apply (e.g., fuzzy logic, time stamp, grouping) to facilitate intelligent rule selection. One significant advantage of expert system technology is their ability to explain their decision-making or problem solving process. Expert systems can trace back which rules were triggered and show them to explain the performed reasoning. This capability has long been identified as a key element of expert system technology and one of the drivers of their implementation and use [[8]].

Applications of Expert Systems for Knowledge Management

To reiterate, expert systems can be used to implement knowledge tasks (create, understand, distribute, and reuse) or an expert system can implement the knowledge process. In order to illustrate the application of expert systems to knowledge management we describe two examples of expert systems: CLUES [[9]] and EULE [[10]].

The Countrywide Load-Underwriting System (CLUES) [[9]] is an expert system that performs loan underwriting. It assesses an applicant's situation and analyses it to decide whether or not to award a requested loan.

When a customer applies for a loan, the loan processor will ask the applicant to complete an application that initiates a data collection process involving many elements of data. The application is turned over to an underwriter who evaluates the information and makes a decision about it. The underwriter represents a bottleneck in the loan approving process, as they are individuals with significant experience.

The role of CLUES is to reduce this bottleneck by extending the availability of the underwriting expertise. The decision to grant or deny a loan will affect the performance of the financial institution offering the loan. Underwriting expertise represented in CLUES is a valuable asset to the financial organization.

Reimer et al. [[10]] describe an expert system named EULE that encodes business processes. Individuals in a business organization execute business processes to accomplish the goals of that organization. In many organizations, staffing has been reduced significantly leaving more to do for a fewer number of people. Because of severe reduction in staff, business process knowledge is often lost as staff is eliminated (business process knowledge leaves with the staff). The setting for Reimer et al.'s system is an insurance company where the office staff must handle approximately 60 complex tasks to meet the needs of the insurance business.

EULE was constructed to support the execution of arbitrary office tasks, as an organization-oriented system. From novice to expert, office personnel can use the system to carry out their day-to-day tasks. The basic process used to represent the knowledge of tasks was to model all of the data objects in a task, process steps, and regulations where they exist. EULE also has the ability to explain why certain tasks need to be carried out in certain ways. Through the explanation capability EULE users can gain insight into why and how decisions in the expert system are made. Thus, expert systems can also be used as tutoring systems to aid personnel training.

Closer to what we would consider to be knowledge-based knowledge management, EULE directly implements the processes of a business. In doing this, business processes are moved from a tacit representation into a systematized knowledge representation and are not subject to the vagaries of shifting personnel. In fact, the company can entirely avoid knowledge loss through attrition through the use of EULE – an important goal for any business enterprise.

EULE and CLUES are representative of the two types of expert systems that we mentioned at the beginning of this section – namely those that implement a knowledge process and those that implement the tasks of a knowledge process to achieve some other goal – like solving a particular problem. EULE is an example of the former while CLUES is an example of the latter. EULE participates in the knowledge processes of the office while CLUES solves a specific problem.

Case-Based Reasoning

Case-based reasoning is a methodology that reuses previous episodes to approach new situations. When faced with a situation, the goal is to retrieve a similar previous one to reuse its strategy. When CBR is applied to problem solving, a previous solution is reused to solve a new problem. The choice of which previous solution to reuse is based on how similar the previous problem is to the current one. This reasoning process can be referred to as the similarity heuristic and its underlying hypothesis is that *similar problems have similar solutions*. This is a very natural form of human reasoning that relies on previous experience, it is relatively easy to implement computationally successfully mimicking this very common method to solve problems and deal with knowledge and experience.

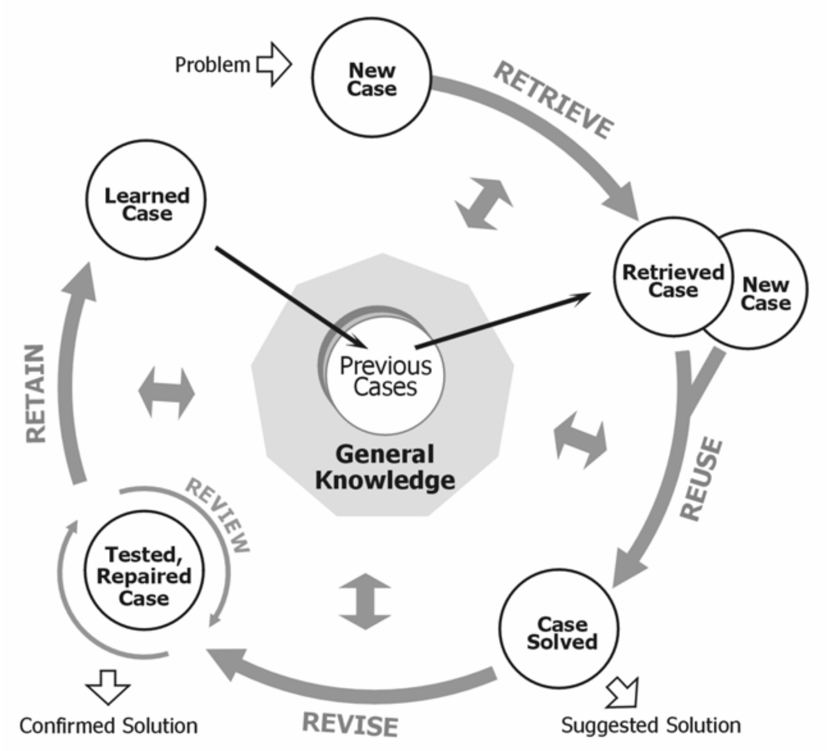

Fig. 3. CBR cycle

The underlying hypothesis of CBR that says that similar problems have similar solutions guides what is known as the CBR cycle (**Fig. 3**), originally conceived by Aamodt and Plaza [[11]]. In problem solving, the cycle starts with a new problem and the goal of the system is to provide a solution to this new problem. This new problem is compared with a collection of past experiences (previous cases) that are composed by problem-solution pairs. The previous cases are stored in the case base, also called case library. The result of this comparison is represented in terms of a score of similarity and based on this score, previous experiences (cases) are sorted. When a list of similar cases is presented in order of similarity, one case is chosen (probably the one with highest score) and the *RETRIEVE* (**Fig. 3**) step is completed. This chosen case lends its solution part (or its strategy) to append to the new problem as its proposed solution, which may be adapted when needed. Adaptation methods substitute or transform items in the solution to suit the new problem; this step is called *REUSE*. This proposed solution is revised and if satisfactory it becomes the solution to the new problem thus generating a new case. This step, called *REVISE*, verifies the appropriateness of reusing the previous case's solution to solve the new problem. Sometimes it is necessary to repair the retrieved solution or record the result of its application to enrich the case. The step REVIEW was introduced by Aha [[12]] to describe the repeated iterations of revision and repair that some types of case-based reasoners need to carry out. This new case can be added to the case base causing the CBR system to learn (*RETAIN*).

Let us consider an example. Suppose a case-based reasoner is prepared to prescribe workouts. The new problem is a woman in her thirties who wants a workout program to help her lose some weight. This woman enters a number of parameters that are considered relevant in this task such as her weight, health history, etc. These parameters are case features that represent the problem in a workout prescription system. They are used to assess the similarity between this new problem and the problems in the previous cases stored in the case base. The similarity functions compare the candidate cases' problems from the case base with the new problem along the features. For example, the feature knee problem is binary and can only be assigned *yes* or *no*. The similarity function evaluates each feature and results *one* if both the new problem and the problem in the candidate case have the same value, either (*yes,yes*) or (*no,no*). The result is *zero* if they are different. The similarity measure represents the distance between each two problems through a score of similarity for each candidate case with respect to the new problem. The retrieval is completed with the selection of one case[2], usually the one with the highest similarity score. Then the reuse step appends the solution component of the selected case as the proposed solution to the new problem and verifies how well this proposed solution fits the new problem by searching for inconsistencies that would require adaptation. Adaptation can be implemented through rules. For example, if the case has a value *yes* for knee problems and the selected case prescribes running, then the system should have a rule to substitute exercise type for swimming. The remaining features (e.g., frequency and intensity) would be kept to guarantee that the goal weight loss is achieved. Revision will determine if the suggested solution was successful and confirm it. The review step authorizes this new case to be incorporated to the case library causing the CBR system to learn with experience.

[2] In some case-based reasoners the suggested solution can also be created by combining a set of cases.

As many other human cognitive processes, case-based reasoning is permeated with uncertainty. When we employ CBR to solve real world problems, the inherent uncertainty in the CBR hypothesis tends to propagate and may cause the system's outcome to be of poor quality. Therefore, the quality of a CBR system's outcome tends to increase as we address its uncertainty. Uncertainty in CBR can originate in the case vocabulary. For example, when cases are described with linguistic concepts that are fuzzy. Fuzzy sets are frequently chosen to represent this uncertainty. Uncertainty also originates in the case base; it most commonly arises when case bases are incomplete. Both fuzzy and non-fuzzy methods can be used to deal with it. The uncertainty that originates in the CBR hypothesis can be dealt with methods that assign a measure of confidence to the CBR result, like the one implemented in PROFIT [[13]] (discussed in Section 0).

Despite the uncertainty posed by the case-based reasoning process, CBR systems, such as expert systems, can provide a good explanation to justify their problem solving. The simplest form of explanation in CBR is by showing the most similar case or most similar cases that support, through experience, a decision.

Case-Based Reasoning and Knowledge Management

CBR systems have continually been appointed as an appropriate method to support knowledge management and its processes [[6],[14],[15]]. One reason can be due to the affinities between the CBR process and knowledge processes. For example, [[5]] draws the parallel between the CBR cycle and the military lessons learned process. Other can be because the similarity heuristics, in which the CBR methodology lies upon, is originally a common form of human reasoning that is commonly employed in knowledge processes. Moreover, CBR systems have successfully been performing knowledge tasks like creation, understanding, dissemination and reuse.

The CBR methodology has been used to perform a variety of different tasks. For example, a past case can be used as a reference to suggest a categorization of a new case instead of being used to lend its strategy for reuse. One can visualize the spectrum of CBR uses by considering that cases embed pairs that may vary in nature, they can be a problem-solution pair, a story-interpretation pair, a symptom-diagnosis pair, a situation-design pair, a situation-plan, a task-lesson, an instance-classification, or a document-key features, among others.

Most of theoretical guidelines to develop CBR systems [[16]] are intended to cover one task only (task-oriented) and therefore variations of such requirement may result in deviations of its expected performance. However, the days of limited performance of multiple-task CBR systems may be soon ending given to current research on multi-case-based reasoning [[17]], where different case bases conceived for different tasks can be used to solve one problem. The use of CBR for pure knowledge management purposes is not task-oriented; it is a form of organizational CBR. Organizational CBR is designed to manage organizational knowledge by performing a generic task. The nature of its pairs can be, for instance, a task-lesson. The scope is limited within communities of practice, that is, groups of organization's members whose goals and tasks are common.

Applications of CBR Systems for Knowledge Management

The most successful commercial application of CBR is conversational case-based reasoning (CCBR) [[18]]. In CCBR, case features are represented in the form of question-answer pairs. In the example of prescribing workout programs, the feature *knee problems* is actually represented with a question, "*Do you have knee problems?*" An interaction with a CCBR tool starts by the user typing a question in free text about the problem that needs to be solved. For example, "*I am 24 years old and I want to improve my fitness level to play basketball.*" The system attempts to find cases (from its case library) that are similar to the case just typed. The system responds by presenting two lists. One list shows a set of case titles with respective scores (the higher the score, the better the case matches the initial situation). The user has access to these complete cases at anytime through double-clicking on the case title. The second list displays a set of questions with respective scores. Higher scores in this question list indicate better chances of finding better cases by answering that specific question. The interaction can continue until the user is satisfied or until the system can no longer improve.

This form of CBR is the basis of a tool offered by E Gain (http://www.egain.com) that has been used to support customer service help desks [[19]] for more than 800 clients. The use of CCBR for customer service help desks implements the task of knowledge creation when the initial case library is acquired; when each new case is stored with answers to questions that are pertinent to the organization the system supports, it implements understanding; knowledge distribution occurs when one of these cases is retrieved; then reuse may occur or not.

The substantial list of deployed applications of CBR includes, for example, the property financial information technology (PROFIT) [[13]]. PROFIT valuates residential properties to evaluate mortgage packages for a division of General Electric Mortgages. Values of a property change with market conditions, so estimates have to be updated constantly according to real estate transactions, which validate the estimations. This system is constantly collecting information from real estate transactions and creating knowledge from it. Interestingly, it is an instance of a knowledge task that is better performed by a computer system because it requires large memory resources and speed. Moreover, this application represents an important competitive advantage because it allows knowledge to be synthesized into a valuable decision-making parameter, making investment choices more reliable.

The CARMA system [[20]] is another example of a deployed CBR system that leverages knowledge. CARMA is designed to provide expert advice on handling rangeland grasshopper infestations. CARMA has reused its expertise combined with model-based methods to devise policies on pest management and the development of industry strategies.

Another application that directly implements knowledge processes was implemented at General Motors [[21]]. It is an organizational CBR system to support the goals of dimensional management, an area in the manufacturing of mechanical structures (e.g., vehicle bodies) that enforces quality control by reducing manufacturing variations that occur in fractions of millimeters. This organizational CBR collects, stores, and shares a collection of previous experiences common to the dimensional management team.

The WWW is a secondary means of supporting knowledge sharing. As a CBR system collects, stores and distributes knowledge, the web facilitates the distribution of CBR knowledge, at least to the extent that it does not require all its potential users to have the complete case libraries physically installed in their computers. The feasibility of some businesses frequently depends upon the management of its sales task force and how quickly it interacts with the engineering staff and quote experts. Western Air is an Australian distributor of heat and air conditioning systems; they have chosen to use a web-based CBR application [[22]] to guarantee a competitive advantage that also poses an entry barrier to competition. They guarantee the precision of the specifications of each new system and the accuracy of the quotes by relying in knowledge captured in previous installations. The company management has realized that previous similar installations provide the required guidance better than other method to delineate good specifications and produce accurate quotes even by inexperienced sales representatives. This system implements knowledge creation, understanding, distribution and facilitates reuse.

Case-based recommendation [[23]] is another task that CBR systems can perform that synthesizes knowledge tasks resulting in the leveraging of knowledge for the benefit of its users. The generic task is to recommend a product to a client in e-commerce applications. This is somewhat different from other CBR systems because in recommender systems, knowledge has not actually been created until a user interacts with the system and together they create knowledge. The process involves an interaction where users (clients) input their preferences by critiquing products that are offered to them. These critiques are used to guide the following retrievals that select recommendations based on similarities inferred between products according to domain knowledge. This type of application combines domain knowledge with user's preferences and a product database, enabling the easy and fast recommendation of products that otherwise might not ever reach these users.

An application of a knowledge-based recommender system that uses CBR is Dublet [[24]]. It recommends apartments for rental in Dublin, Ireland, based on a description of the user's preferences. Dublet also makes use of the sharing ability of the WWW by employing information extraction to create cases dynamically by extracting information from the web of apartments for rent and retrieves units that match the user's preference. Dublet performs knowledge synthesis (creation) and extends the power of knowledge distribution of the CBR system by being operational in cell phones.

NEC [[25]] is using knowledge stored in a case library to reach unprecedented levels in quality control and marketing of its products. One important issue they have addressed refers to maintenance of case bases that used to be updated manually. The obvious problems arising from manual updates were overcome with the creation of FAQ case bases automatically, allowing creation and understanding to be implemented without human intervention. SignFinder is a system that detects variations in the case bases generated automatically from customer calls. When they detect variations on the content of typical customers requests, they can discover knowledge about defects on their products faster than with any other method. This system illustrates many knowledge processes in an organizational system that collects, understands, distributes, and also reuses its knowledge to discover more knowledge. Besides, the automatic update of the case library allows the timely sharing of new defects solving the knowledge sharing obstacle they face due to the magnitude of their organization. In addition to the increased efficiency obtained by sharing the new discoveries among all agents, they gain valuable time by detecting defects in time for the company to take appropriate measures and minimize the damage to the brand's image.

As the previous system reuses knowledge to create more knowledge, the following application implements automatic reuse for specific organizational tasks. Note that KM systems have to encourage users to apply distributed knowledge to promote reuse. One way of doing this is by distributing knowledge to the user in the same context where it can be reused, and the user can implement it with just the press of a button (though it can be configured to reuse without human acknowledgement). This approach is implemented in the active lessons delivery system (ALDS) [[5]], which has been conceived to implement the military lessons learned process (last column in **Table 2**).

ALDS is a case-based module that contains lessons learned represented as cases in an organizational CBR system. Each case embodies a pair task-lesson, whose tasks are part of an organization's process. ALDS can be incorporated to any information system where organizational processes are performed (e.g., a decision support system, enterprise resource planning system). ALDS keeps track of the task the user is performing and some other variables (state conditions) to retrieve and display an applicable lesson. Because lessons are represented in terms of their applicability to each task and how this applicability may vary based on the state conditions, ALDS can display applicable lessons whenever they are needed. This form of just-in-time knowledge facilitates reuse because lessons are presented to the user in the context where they can be applied. When the target process allows, and the reuse can be executed computationally, it can be done through the press of a button. Weber and Aha [[5]] describe an example of automatic reuse incorporating ALDS to a plan authoring tool.

The examples above combine many knowledge tasks, though CBR can be also used with emphasis on implementing knowledge collection. Once there is organized knowledge in a repository, they can be used as previous cases to guide knowledge collection from human users. The benefit is the use of previous knowledge artifacts in a CBR tool to instruct users of the format in which knowledge should be communicated and immediately understood by the system. We prescribe implementing knowledge creation with knowledge-based methods to guarantee that understanding complements creation. This is how humans collect, validate and store knowledge, comprising a *genuine knowledge process* (see Section 0).

Ontologies

Ontologies represent another knowledge-based methodology for the development of systems that implement knowledge tasks. The main purpose of ontologies is to provide a common language to support knowledge sharing. In practice, a common language guarantees effective knowledge sharing, and thus should be a requirement for all systems intended to implement knowledge tasks. An excellent discussion of the collaborative creation of an ontology to provide a shared language is presented in [[26]].

Ontologies are an old solution to support knowledge processes. Philosophers have relied on ontologies as a formalism that explains the nature of everything. Artificial intelligence (AI) researchers have adopted ontologies as a comprehensive knowledge representation formalism to provide commonsense reasoning in support of knowledge tasks such as knowledge acquisition and reuse (see Lenat [[27]]).

Initially, AI researchers supported the development of ontologies to provide commonsense to other intelligent systems, like ES and CBR, which lack this feature. Today, ontology development and reuse is a topic of AI that encompasses an independent methodology. Ontology-based systems are commonly equipped with reasoning methodologies. Because the representation language used is typically based on first order logic, rule-based reasoning is easy to implement within ontology applications.

The knowledge-based community envisages ontologies as a knowledge model that meets the requirements of being explicit, consensual, and conceptual [[28]]. It is explicit because its knowledge has to be explicitly defined; an explicit representation of a concept can guarantee its consistency. It is consensual because its knowledge must reflect agreement among its users. It is conceptual because its knowledge abstracts a meaning of the elements it entails, representing a model of a domain or the world.

Top-level ontologies represent mundane elements of the world such as air and mass, typically consensual concepts. When applying ontologies, a top-level ontology can be appended to a domain ontology that specifies the concepts of the specific domain of application.

To design an ontology for a specific domain, it is required to start from an ontological analysis of this domain. Ontological analysis are very likely to be reused, however a different application might require adaptations with respect to the specific facets that are considered in one application and not in others. This is a sample of the possible challenges posed in this reuse, though the consensus is that it pays off to reuse. After choosing the right ontological analysis, the following step is to use the analysis to organize common concepts. An important step is to choose a proper top-ontology to reuse. Depending upon the application, you may need to develop task and method ontologies.

Right now, it is still a challenge to find the right ontological analysis and one is likely to need to formulate it. However, this formulation can and shall be performed with a combination of existing ontological analyses.

Ontologies and Knowledge Management

The use of ontologies has been consistently associated with knowledge management [[29],[30]] because of its main conceptual purposes such as supporting knowledge sharing, acquisition, and knowledge representation. These are intrinsically related to knowledge engineering itself and its tasks, which are, de facto, knowledge tasks. The biggest challenges faced by knowledge engineering in developing knowledge-based systems are mainly related to the high costs of knowledge acquisition and the lack of commonsense knowledge. The adoption of an ontology can help overcome these challenges.

The AI movement towards ontologies is relatively young and thus there are not many successful applications to illustrate its use. Some applications use parts of ontologies as a representation formalism, but most applications that are considered to be ontology-based are research prototypes. The promises of facilitating knowledge sharing with the reuse and merging of ontologies seem to be a bit tougher than first predicted. In the next subsection we describe some promising applications and useful methods to help put ontologies to work.

Applications of Ontologies for Knowledge Management

Developing ontologies for a specific domain is costly; therefore to make this effort beneficial, a domain ontology has to be reused in more than one application. Large organizations are then the obvious audience for such efforts, particularly if the goals are to achieve knowledge management goals. The larger the organization, the harder it is to promote knowledge sharing and guarantee the flow of knowledge among its members; in addition, there are more experiences that need to be shared, and more idiosyncrasies the organization may encompass. All this makes knowledge management tasks more significant in achieving an organization's goals and at the same time more difficult to perform. Examples of large organizations that can potentially benefit from developing ontologies are in the government (e.g., space agencies, IRS), the military (e.g., Navy), and large manufacturers that have many plants and products (e.g., General Motors).

An example is the ontology developed for the JFACC (Joint Forces Air Component Commander) to support many different knowledge-based systems for air campaign planning [[31]]. Among the goals achieved in these projects were the individual KBS, and the integration of knowledge acquisition and modeling efforts to streamline KBS development. In addition, the ontology can also be used in systems that are not knowledge-based. The experience reported in this project revealed that when attempting to share the same ontology throughout different applications, several problems come up. For example, merging knowledge bases into ontologies, importing knowledge from ontologies into knowledge-based applications, and translating a publicly available ontology into a new language. Consequently, most research in this field today addresses these challenges.

In some applications, ontologies play a secondary role, and some illustrative examples go unnoticed. For example, the organizational CBR developed at GM (described in Subsection 0) to support the dimensional management team, uses an ontology of objects, verbs, verb phrases, and also includes a library of pictures, which can be very helpful in real domain applications. The purpose of this ontology is to explicitly define common concepts to support knowledge acquisition and reuse.

The first niche of applications for ontologies in intelligent systems was natural language because of the advantages of using a shared vocabulary to support tasks such as disambiguation, content identification through the use of semantic categories, and metonymy resolution [[32]]. Knowledge-based machine translation became feasible with the use of explicit common set of concepts and axioms [[33]].

The use of linguistic ontologies has been categorized along different applications to perform machine translation, parsing, automatic translation of natural language words into a lexicon, use of wrappers, automatic translation of ontologies, and search [[34]]. Examples of abstract ontologies are CYC and WordNet [[32]]. The use of ontologies to support natural language also plays an important role in text mining and information extraction.

It seems as if the main contribution of ontologies to knowledge engineering and knowledge management is the ability to extend previous knowledge engineering methods to open domains. First, the limitations imposed by the knowledge acquisition bottleneck are neutralized by the benefits of acquiring knowledge with the support of an ontology. Second, ontologies can give structure to knowledge by indicating the bounds and the concepts within a domain, thus alleviating the limitations imposed by restricted domains. For example, [[35]] proposes the semi-automatic creation of local ontologies through the interaction of users and the use of machine learning methods employed over a collection of documents. The result of the machine learning methods is the creation of templates to support information extraction to this new domain. This illustrates how the use of ontologies can aid in overcoming current limitations that hamper the use of techniques such as information extraction to open domains. The semi-automatic creation of a new ontology guided by user-interaction is also used in a more ample application to merge ontologies [[36]]. Merging ontologies refers to the process of generating one ontology from two or more source ontologies, which is necessary in overlapping problem domains.

One of the strongest demonstrations of an ontology's ability to implement knowledge sharing is when it supports community web portals [[37]]. Web portals are a common web-based entity that attracts users from a specific community whose interests and ontologies are common by definition. It is then obvious to link to such portals the explicit representation and definitions of the consensual knowledge of a community. The direct benefit is to allow intelligent retrieval through querying and an inference engine incorporated to the ontology.

Sometimes ontologies are needed to implement knowledge sharing among individuals with different views of the world. To enable the browsing and querying for users in a broad audience, it is necessary to build different views of information. Another use of ontologies to provide semantic infrastructure to web-based applications is myPlanet [[38]]. This application makes use of personal views of an ontology to facilitate access to specific services. This is a way to keep a consensual ontology and also allow for personal interests to be contemplated in a search for specific news items. They use an interest-driven profiling tool and deductive retrieval techniques to redirect e-stories sent to a news server. The profiling tool captures user's preferences to determine which information is likely to be useful to which users. This is a form of active casting [[6]] dissemination that attempts to automatically detect potential audience of information. MyPlanet illustrates the common use of information extraction from user's resources to update the ontology. The knowledge obtained by users is used to deploy maintenance methods, one of the most challenging issues in ontology research.

There is intuitive indication that information retrieval can benefit from using ontologies. Aitken and S. Reid [[39]] describe some preliminary experiments supporting this claim.

The use of ontologies to provide commonsense reasoning is sometimes extended in the integration with other knowledge-based methodologies. The idea is that once one adopts an ontology to represent all concepts, actions, and axioms, then it seems reasonable that you conduct all the reasoning within the same representation paradigm. For this purpose, Díaz and González [[40]] have formalized the case-based reasoning methodology in an ontology. Therefore, actions such as retrieval and adaptation are properly encoded in a CBR-specific ontology that can be used in combination with domain-specific ontologies to develop knowledge-rich CBR applications.

Today, there are several ontology editors that can make ontology development and reuse a relatively simple task. A pragmatic analysis of these tools is found in [[41]], where one can find the essential description to guide an effective choice. Another pragmatic perspective on deploying, incorporating, and maintaining ontologies describes ontologies within object-oriented analysis [[42]].

The last decade has witnessed a growing occurrence of research scientists and practitioners looking at ontology-related issues. This growing rate is likely to increase because of what is considered to be the most important application of ontologies, the semantic web. Today, the challenge is to find a standard representation language to make the semantic web operational. One of the promising languages is DAML+OIL[3]. Then, the need of methodologies for reusing and merging ontologies will have a pragmatic use whose results will benefit all of us users of the web.

---

[3] http://www.daml.org/2000/12/daml+oil-index.html.

Case Study – Expert Systems and Business Rules

In surveying the existing literature, examples of explicit KBKM – where as part of the project it was well understood about the relationship between KM and KBS – are difficult to find. In fact in some cases selling a KM implementation based on any knowledge-based technology may be impractical given the stigma of these systems over the years. In addition, corporations tend to be protective over successful KM initiatives that represent competitive advantage. Our case study describes a framework for KBKM efforts on which these systems can be built.

In large corporations, the way a business operates can be articulated in the form of business rules. A business rule, stated simply, is a rule by which a business operates. Business rules pervade all aspects of business from large to small. Business expertise is also articulated in terms of business rules – in fact, over time, if a business is successful, the business rules should represent the experienced view of running a particular business. Oftentimes it is the case, just as it is the case with the traditional expert system development effort, that business rules are both implicit – the business operates with a set of rules that are not formally stated, and tacit – the employees maintain the rules in their heads. So it would seem the paradigm that businesses use to operate (business rules) is the exactly same as is used in expert systems.

For a moment, let's take a step back and consider the relevance of rules to business and in particular to the management process. In an article entitled *Strategy as Simple Rules* [[43]] the authors attempt to articulate the rules of strategy for business organizations. They identify five types of rules used by managers in business processes. These rule types are *how-to* rules, *boundary* rules, *priority* rules, *timing* rules, and *exit* rules. Eisenhardt and Sull [[43]] define and give an example of each of these rules. For example, a boundary rule is defined as *specifying how managers will focus on which opportunities can be pursued and which are outside the pale*. For example, Cisco's early acquisitions rule is as follows: *Companies to be acquired must have no more than 75 employees, 75% of whom are engineers*.

Although this example of a simple strategy rule is very high level one can see that such rules could be easily codified into an expert system type of representation:

If you are deciding to acquire a company,
    And the type of acquisition is <u>early</u>,
    And the company has no more than 75 employees,
    And 75% of the employees are engineers, THEN
This company represents a viable candidate for acquisition.

We could imagine that associated with this rule are others that guide the acquisition process and collectively could be used as a reference, a training device, or a simulation tool for acquisitions.

A primary issue with business rules being tacit and implicit is that for the organization this knowledge will not be permanent. If an employee leaves the organization, then the tacit knowledge leaves with them. Expert systems afford organizations an opportunity to prevent this from happening by retaining the corporate memory of business rules within the knowledge base of the expert system. In addition to the retention aspect of expert systems, there are several others as delineated by Hoplin and Erdman [[44]]. These researchers offer that the advantages of expert systems to business organizations include retention, ease of transfer, easy to document, and more consistent decision making (the computer is not likely to be distracted). Based on their data (albeit from 1990) they also suggest that initially although the cost for development is high, *expert systems cost less than their human counterparts in the long run*.

We suggested earlier that expert systems offer organizations some unique opportunities. Hoplin and Erdman [[44]] point out that an expert system offers the corporation expertise on demand thereby making it unnecessary for the human expert or experts to be physically present. Second, the expert system can be used for predictive modeling. Take for example, a knowledge base of rules built around the Cisco rule of acquisition. Given a perspective acquisition, Cisco could run the acquisition through its expert system to determine the impact on its existing business. A comprehensive expert system for acquisition would have many business rules for determining whether an acquisition should be made.

Due to the dynamic of real world businesses, it is likely that these rules evolve over time. To cope with an organization's growing we suggest incorporating a lessons-learned module [[5]]. This case-based module (see Section 0) would capture lessons either manually (i.e., by acquiring lessons from organization's members) or automatically (i.e., by detecting shifts in the use of rules) and distribute understood knowledge back to its users to ultimately be incorporated into the knowledge base in the form of new rules.

Finally, given the magnitude of the target organization, an ontology can be developed to represent the organization's domain. Properly attached to an abstract ontology, it would be able to capture commonsense knowledge to facilitate validation and verification methods for the expert system and case-based components.

Conclusion

Knowledge-based knowledge management refers to technologies that explicitly manipulate knowledge to perform knowledge tasks of the knowledge process. We have examined knowledge processes in its conceptual form when conceived and performed by humans and some instances of the knowledge processes implemented with technology.

It appears as the human knowledge cycle includes understanding in the knowledge collection; whereas computerized knowledge processes designed to be supported by computers are conceived as having understanding later in the process. From this statement one could identify as genuine knowledge processes only the ones that are able to combine understanding as part of knowledge creation (or as an immediate consequence). Therefore, a genuine knowledge creation would have to be performed either by humans of by intelligent systems that can emulate the required grounds for computer understanding, which include being able to manipulate knowledge and reason with it; identify analogies; and be able to autonomously reorganize its body of knowledge after each new collection [[45]].

We have examined three methodologies that perform knowledge processes that we denote as knowledge-based knowledge management. These methodologies are expert systems, case-based reasoning, and ontologies. We have described recent advances and successful applications of these methodologies and how well they are suited for knowledge management tasks.

KM is gaining increased importance as both practitioners and research scientists acknowledge its relevance by implementing various frameworks to address pragmatic KM problems reusing decades of technology developed for knowledge-based systems. This phenomena itself can be considered a KM task, as knowledge reuse, representation, and retrieval are amongst the most important goals of KM. This also represents strong evidence of the adequacy of using knowledge-based technologies for any KM effort.

The KBKM technologies represent the knowledge engineering body of knowledge and practice of manipulating knowledge-based systems. This body of expertise is the state of the art to support technology-based solutions for knowledge management. Some of the applications here described exemplify how KBKM can be integrated to organizational environments to serve an organization's goals.

We have tried in this chapter to demonstrate the relevance of knowledge-based technologies to knowledge processes and knowledge management tasks. Clearly there is relevance here if and only if technology is not taken as the *fix* for the problem. When an approach is taken where technology is used as the fix for knowledge management, the knowledge management solution is less than satisfactory. Nevertheless, the methods we discuss offer frameworks and tools that can aid in the process of knowledge management. Since many of these technologies have long histories of research it may be possible for an organization to gain insight into knowledge processes by utilizing one or more of the technological methods we discuss.

References

[1]. Nissen ME (1999) Knowledge-based knowledge management in the reengineering domain. Decision Support Systems 27 47–65

[2]. Laudon KC, Laudon JP (1998) Management Information Systems. Prentice-Hall, NJ

[3]. Liebowitz J (2001) Knowledge Management: Learning from Knowledge Engineering. CRC Press, FL

[4]. Staab S, Studer R, Schnurr HP, Sure Y (2001) Knowledge Processes and Ontologies. IEEE Intelligent Systems 16, 1: 26-34

[5]. Weber R, Aha DW (2003) Intelligent Delivery of Military Lessons learned, Decision support systems 34, 3: 287-304

[6]. Weber R, Aha DW, Becerra-Fernandez I (2001) Intelligent lessons learned systems. International Journal of Expert Systems Research & Applications 20:117-34

[7]. Feigenbaum EA (1982) Knowledge Engineering in the 1980's. Stanford University

[8]. Hayes-Roth F, Waterman DA, Lenat DB (1983) Building Expert Systems. Addison-Wesley, Reading, MA
[9]. Talebzadeh T, Mandutianu S, Winner CF (1995) Countrywide Loan-Underwriting Expert System. AI Magazine 16, 1: 51-64
[10]. Reimer U, Margelisch A, Staudt M (2000) EULE: A Knowledge-Based System to Support Business Processes. Knowledge-Based Systems 13: 5261-269
[11]. Aamodt A, Plaza E (1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Communications 7, 1: 39-59
[12]. Aha, D. W. (1998). The Omnipresence of Case-Based Reasoning in Science and Application. Knowledge-Based Systems 11,5-6: 261-273
[13]. Cheetham W, Cuddihy P, Goebelch K (2001) Applications of Soft CBR at General Electric. In: Pal SK, Dillon, TS, Yeung, DS (eds) Soft Computing in Case-Based Reasoning. Springer Verlag, London, pp 335-365
[14]. Aha DW, Becerra-Fernandez I, Maurer F, Muñoz-Avila H (eds) (1999) Exploring Synergies of Knowledge Management and Case-Based Reasoning. (Papers from the AAAI Workshop, Technical Report WS-99-10 AAAIPress, Menlo Park, CA)
[15]. Watson I (2001) Knowledge Management and Case-Based Reasoning: a perfect match? In Russel I, Kolen J (eds) Proc. of the Fourteenth Annual Conference of the International Florida Artificial Intelligence Research Society. AAAI Press, Menlo Park, CA, pp 118-122
[16]. Kolodner J (1993) Case-Based Reasoning, Morgan Kaufmann, Los Altos, CA
[17]. Leake DB, Sooriamurthi R. (2002) Automatically Selecting Strategies for Multi-Case-Base Reasoning. In: Craw S, Preece A. (eds) Advances in Case-Based Reasoning 2416. Springer-Verlag, Berlin, pp 204-218
[18]. Aha DW, Breslow LA, Muñoz-Avila H (2001) Conversational case-based reasoning. Applied Intelligence 14: 9-32
[19]. Watson I (1997) Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufmann, San Mateo, CA
[20]. Branting LK, Hastings J, Lockwood J (2001) CARMA: A Case-Based Range Management Advisor. In: Proceedings of the Thirteenth Innovative Applications of Artificial Intelligence. AAAI Press, Menlo Park, CA, pp 3-10
[21]. Morgan AP, Cafeo JA, Gibbons DI, Lesperance RM Sengir, GH, Simon, AM (2001) CBR for Dimensional Management in a Manufacturing Plant. In: Aha DW, Watson I (eds) Lecture Notes in Artificial Intelligence 2080. Springer Verlag, New York, pp 597-610
[22]. Watson I (2000) A Case-Based Reasoning Application for Engineering Sales Support using Introspective Reasoning. In: Proc. 12th Innovative Applications of Artificial Intelligence Conference. AAAI Press, Menlo Park, CA, pp 1054-1059
[23]. Burke R (2000) A Case-Based Reasoning Approach to Collaborative Filtering. In: Blanzieri E., Portinale L (eds) Advances in Case-Based Reasoning 1898. Springer Verlag, New York, pp 370-379
[24]. Hurley G, Wilson DC (2001) DubLet: An Online CBR System for Rental Property Recommendation. In: Aha DW, Watson I (eds) Lecture Notes in Artificial Intelligence 2080. Springer Verlag, New York, pp 660-674
[25]. Shimazu H, Kusui D (2001) Real-Time Creation of Frequently Asked Questions. In: Aha DW, Watson I (eds) Lecture Notes in Artificial Intelligence 2080. Springer Verlag, New York, pp 611-621

[26]. Sorenson C, Snis U (2000) Codify or Collaborate. From Expert Systems to Systems of Experts for Knowledge Creation in Manufacturing. In: Proc. of IRIS 23. Online: http://iris23.htu.se/proceedings/PDF/109final.PDF
[27]. Lenat DB (1976) AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search. PhD thesis, Stanford University
[28]. Gruber TR (1993) A translation approach to portable ontologies. Knowledge Acquisition 5: 2199-220
[29]. O'Leary DE (1998) Using AI in Knowledge Management: Knowledge Bases and Ontologies. IEEE Intelligent Systems 13: 334-39
[30]. Maedche A (2001) Tutorial on Development and Application of Ontologies. In: Proc. of ECML/PKDD. Online: http://wim.fzi.de/wim/presens/entries/1010241341.ppt
[31]. Valente A, Russ T, MacGregor R, Swartout W (1999) Building and (Re)Using an Ontology of Air Campaign Planning. IEEE Intelligent Systems 14, 1: 27-36
[32]. Chandrasekaran, B, Josephson, JR, & Benjamins, VR (1999)What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems 14, 1: 20-26
[33]. Nyberg E H, Mitamura T (1992) The KANT system: fast, accurate, high-quality translation in practical domains. In: Proc. of the 14th International Conference on Computational Linguistics. University of Nantes, vol 4, pp 1254-1258
[34]. van Zyl J, Corbett DA (2001) Framework for Comparing the use of Linguistic Ontology in an Application. In: Proc. of the Fourteenth European Conference on Artificial Intelligence. Workshop on Applications of Ontologies and Problem-Solving Methods. Online: http://www.cis.unisa.edu.au/~cisjvz/publications.html
[35]. Poibeau T (2001) Deriving a multi-domain information extraction system from a rough ontology. In: Proc. of the Seventeenth International Joint Conference on Artificial Intelligence. Morgan Kauffmann, San Francisco, CA, pp 1264-1272
[36]. Stumme G, Maedche A (2001) FCA-MERGE: Bottom-Up Merging of Ontologies. In: Proc. of the Seventeenth International Joint Conference on Artificial Intelligence. Morgan Kauffmann, San Francisco, CA, pp 225-230
[37]. Staab S, Jürgen A, Decker S, Erdmann E, Hotho A, Maedche A, Schnurr HP, Studer R, Sure Y (2000) AI for the Web - Ontology-based Community Web Portals. In: Proc. of the Twelth Innovative Applications of Artificial Intelligence Conference. AAAI Press/MIT Press, Menlo Park, CA, Cambridge, MA, pp 1034-1039
[38]. Kalfoglou Y, Domingue J, Motta E, Vargas-Vera M, Shum SB (2001) myPlanet: An Ontology-Driven Web-Based Personalized News Service. Workshop on Ontologies and Information Sharing in the Seventeenth International Joint Conference on Artificial Intelligence. Online: http://kmi.open.ac.uk/projects/kmi-planet/
[39]. Aitken S, Reid S (2000) Evaluation of an Ontology-Based Information Retrieval Tool. In: Workshop on the Applications of Ontologies and Problem-Solving Methods in the European Conference on Artificial Intelligence. Online: http://delicias.dia.fi.upm.es/WORKSHOP/ECAI00/15.ps
[40]. Díaz-Agudo B, González-Calero PA (2000) An architecture for knowledge intensive CBR systems. In: Blanzieri E., Portinale L (eds) Advances in Case-Based Reasoning 1898. Springer Verlag, New York, pp 37-48
[41]. Duineveld AJ, Stoter R, Weiden MR, Kenepa B and Benjamins VR (2000) WonderTools? A comparative study of ontological engineering tools. International Journal of Human-Computer Studies 52, 6: 1111-1133
[42]. Knublauch H, Rose T (2000) Round-Trip Engineering of Ontologies for Knowledge-based Systems. In: Proc. of the Twelfth International Conference on Software Engi-

neering and Knowledge Engineering. Knowledge Systems Institute, Chicago, IL, pp 239-247
[43]. Eisenhardt K M, Sull, D N (2001) Strategy as Simple Rules. Harvard Business Review Jan-Feb: 106 – 116
[44]. Hoplin HP, Erdman S J (1990) Expert Systems: An Expanded Field of Vision for Business. In: Awad EM (ed) Conference on Trends and Directions in Expert Systems. ACM Press, New York, NY, pp 1-16
[45]. Jackson P (1999) Introduction to Expert systems. Addison-Wesley, Harlow, England