Effective Knowledge Navigation For Problem Solving Using Heterogeneous Content Types

Ralph Barletta and Philip Klahr
Inference Corporation
100 Rowland Way
Novato CA, 94945
ralph.barletta@inference.com
phil.klahr@inference.com

Introduction

Over the past few years many companies have begun emphasizing and implementing processes to capture and share knowledge within their organizations [Davenport & Prusak, 1997; Edvinsson & Malone, 1997; Stewart, 1997]. Corporate infrastructures are being developed to facilitate knowledge creation and reuse, and a variety of software tools are currently available to support the knowledge management (creation, storage, maintenance and retrieval) process. The main goal of these efforts is to shorten and improve the problem solving process by providing access to relevant knowledge. Businesses want to build knowledge management systems to solve some of these important problems:

- "How do I do X?"
- "Has X been done before? If so, how?"
- "What did we do to handle this problem the last time something like it happened?"

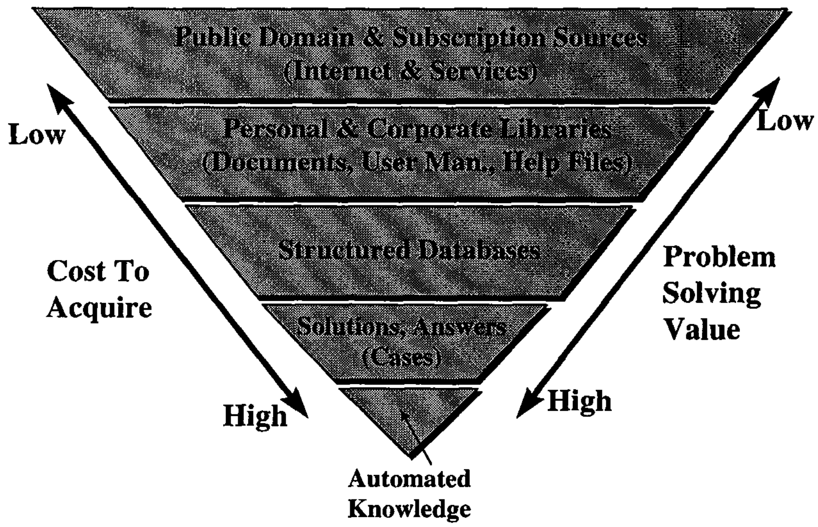
Problem solving knowledge comes in a variety of forms. Much of the information we currently collect for purposes of problem solving reuse is unstructured: documents, technical notes, policy manuals, e-mails, presentations, etc., contain a wealth of corporate information that can be useful for problem solving. A smaller, but still significant fraction of corporate information that can be useful for problem solving is in a highly structured form (typically stored in a database): Customer and demographic data, sales transactions, financial data, etc.. The rapid growth of data warehouses for decision support is a clear indication of the perceived value of structured data for problem solving. A smaller still, but rapidly growing form of information that companies are collecting is in the form of semi-structured, problem/solution pairs, or cases as they are commonly called. A case [Kolodner, 1993] describes a particular problem or scenario that was encountered and the solution to that problem/scenario. For example, in a customer support domain, a case might consist of a software failure a user was having, the diagnostic information that was collected to isolate the cause of the problem and a recommended fix for the problem [Klahr, 1997]. The collection and use of cases is of growing interest in the knowledge management area because this type of knowledge is more explicitly geared towards reuse in a problem solving context.

Given the desire to collect these different forms of information, an organization needs software tools for storing, and more importantly, retrieving the information. Groupware tools like Notes have been useful in providing a framework for collecting unstructured and semi-structured information, while relational database vendors have long provided tools for collecting structured (and semi-structured) information. With the recent emergence of "Universal Servers", we are starting to see the potential for a single storage repository for all types of information, structured semi-structured and unstructured. Technologically, information storage is not the major hurdle in implementing knowledge management systems. The major technological hurdle in implementing effective knowledge management systems is in providing effective navigation and search tools for finding contextually relevant information to a given problem at hand.

This paper proposes an architecture for knowledge navigation based on our experience in implementing knowledge systems for business users, predominantly through the use of cases as the main knowledge source. This architecture is geared toward searching among heterogeneous sources of knowledge using an interactive, question driven approach that we have found to be highly successful in case-based retrieval systems. Our framework addresses the needs of business users who are trying to solve business problems, such as diagnosing customer problems, matching available products to the needs of customers, clarifying company policies and procedures for employees, and even designing new products. These are all knowledge activities that can benefit from corporate knowledge captured and effectively searched and reused.

Problem Solving In The "Knowledge Triangle"

The figure below shows the various different types of knowledge that are used for problem solving. As the figure indicates, knowledge closer to the bottom is more precious for problem solving and harder to come by than knowledge in higher layers.



Automated Knowledge

At the bottom of the triangle is "Automated Knowledge". Automated knowledge is essentially knowledge that can either automatically recognize a problem and/or automatically apply a fix to the problem. In the case of computer hardware, automated knowledge might recognize a configuration problem and automatically apply an adjustment to the configuration. In a marketing application, automated knowledge might recognize an opportunity to put a product on sale and automatically mark down the appropriate items or send out a targeted mailing. Automated knowledge is extremely high value but is also very costly to create and maintain. That is why it tends to be available only when it is cost justified to encode knowledge in such a detailed way. Automated knowledge is often represented as a rule-based or case-based expert system that has been enhanced with procedural code (scripts) for collecting data and activating the appropriate action.

Cases

Above Automated Knowledge are non-automated, Problem/Solution pairs or cases. Cases are used to represent a specific set of problem scenarios and their associated resolutions. Intuitively, any time a problem is encountered, the most ideal form of information (beyond automated knowledge) is to provide the user with a specific past example that mirrors their current problem. Any time a new problem is solved it presents us with an opportunity to create a new case. By learning new cases in this way, the knowledge navigation process gets smarter over time.

The main difference between automated knowledge and cases is in the lack of automation scripting. In many domains, it is impossible to automate the gathering of problem information or the application of a solution. This certainly does not diminish the value of collecting non-automated, case knowledge.

The main difficulty with knowledge in the form of cases is that cases can be expensive to build and maintain, relative to things like documents or databases. Given this, cases are most often used to handle commonly occurring, well understood problems or scenarios that are worth spending the time to structure into case form. For many organizations, case knowledge is often made available to less knowledgeable personnel or customers for self-help applications due to the fact that a case can provide an easily interpretable solution to a commonly occurring problem. More complex problems are better handled by domain experts who can combine their expertise with harder to interpret, document oriented knowledge available at higher levels of the triangle.

Databases

Certainly, databases have been a mainstay in businesses for a long time. However, it is only recently that they have been pushed as a key element in building decision support systems, and it is in this context that databases are relevant to knowledge management. The advent of data warehouses, multi-dimensional databases, data mining, and on-line analytical processing (OLAP) have signaled the move from viewing data and databases in the domain of the "back office" to viewing data and databases as important knowledge sources for "front office" problem solving.

While structured data exists in much greater abundance than automated knowledge or cases, it is not typically organized in a way that facilities its use for problem solving. Problem solving takes place by mapping a general question like "How is product X doing relative to its competitors" into complex SQL statements that produce graphs or tables that must then be interpreted by the user. This makes databases somewhat less attractive for problem solving than cases or automated knowledge because it requires the user to know more about the data and how to formulate a good query than is necessary for cases or automated knowledge. Nevertheless, databases contain complementary knowledge that must be taken into account went building a knowledge navigation architecture.

Corporate Repositories & Public Internet Sites

The top levels of the knowledge triangle are predominantly represented as freeform document collections. This information is abundant and contains a wealth of potentially valuable problem solving information. The difference between corporate repositories and public internet sites is that one is under organizational control and the other isn't. Control over the repository effects the quality and form of the information and its relevance to the organizations users. For this reason, despite the fact that both sources of content are searched using text retrieval, they are represented in different layers in the triangle. We generally would prefer corporate information to public information in the context of solving most problems of corporate interest.

What makes text storage and retrieval attractive to many companies as their main knowledge management paradigm is that it is relatively easy to write up a document in any form and on any subject that will be accessible via the search engine. While, this certainly makes life easy for the knowledge author, unfortunately, it makes life very difficult for the knowledge searcher. Text search typically suffers from lack of precision and recall, particularly when used in a specific problem solving context. Usually, too many documents are retrieved or they are not the best ones because the user doesn't know how to phrase their query relative to the problem they are trying to solve.

In general, after forming an initial keyword or natural language query to a text retrieval engine, the user has one of two unattractive options:

1. Play the "guessing game": What keywords can I add that will reduce the hit set to a more manageable size?
2. Play the "next 20 hits game": Can I find a document (by examining the titles) in the first several hundred that are close enough to what I want to use relevance feedback to refine the query.

Anyone interacting with Internet or local text search engines understands the tremendous frustration involved in playing either (or a combination) of these time consuming "games" when trying to do real work (as opposed to just surfing). What is needed is a better approach for assisting the user in formulating and refining text retrieval queries so that they can either quickly find what they are looking for or at least quickly determine that what they are looking for doesn't exist.

The Knowledge Factory & Knowledge Navigation

In our view, a complete knowledge management solution needs to address two major components:

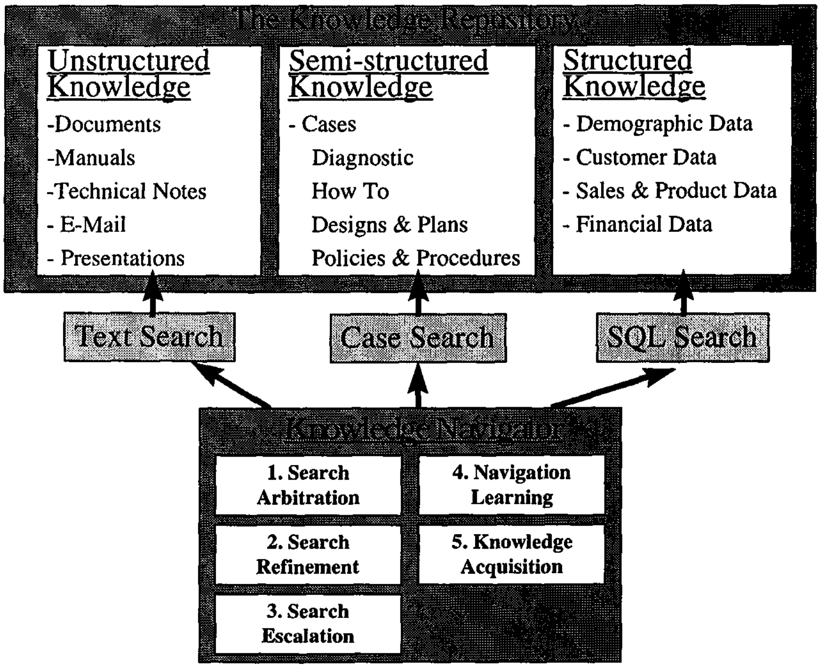
1. A Factory component for establishing the processes to create, monitor, maintain and extend knowledge.
2. A Navigation component for effectively accessing and retrieving knowledge.

Both of these components are critical to successful knowledge management. The first deals with creating useful knowledge content, and avoiding the "garbage in, garbage out" syndrome. It is this area that has been given the most emphasis in knowledge management efforts, i.e., allowing users to capture their knowledge, typically in an unstructured form, and make that knowledge available to other members of the organization or directly to customers (e.g., in electronic manuals, on-line notes, etc.). We will not elaborate on this knowledge management element other than to say that our approach to the addressing it is to establish a *Knowledge Factory* for creating, maintaining and extending knowledge. The Knowledge Factory defines the processes (similar to the steps in an manufacturing assembly line), the tasks to be accomplished in each process (the deliverables), tools to be used (the machines), and the skills required (job definitions). (See, for example, [Borron, Morales & Klahr, 1996] and [Thomas, Foil & Dacus, 1997].) The Knowledge Factory needs to be a permanent and evolving structure for it to be effective on an on-going basis.

The second component of effective knowledge management, and the main focus of this paper, is called *Knowledge Navigation.* Knowledge Navigation is the process of finding relevant problem solving knowledge among all the available knowledge in the repository. This navigation process is just as important as the Knowledge Factory, but up to now has been predominantly viewed and implemented as an open ended, user driven, text retrieval task. This current approach to navigation is unacceptable for several important reasons:

1. As we have claimed, problem solving knowledge comes in a variety of forms, including cases and databases. These types of knowledge can play a vital role in effective problem solving and neither are well served by text retrieval approaches.
2. Companies that take a pure text retrieval approach to knowledge navigation are finding that casual or uninformed users of the knowledge repository are not capable of formulating effective queries to find relevant documents. Current interactions with Internet search engines provide clear examples of the difficulty in quickly finding relevant knowledge to suit a user needs based on keyword search.
3. Even when users do find useful unstructured information, they often find it difficult to interpret the resulting knowledge in the context of the particular problem they are trying to address.
4. There is no learning mechanism in place that will improve access to frequently used documents and convert them over time to more structured forms (like cases) that are more easily searched and interpreted.

Our approach to addressing the current shortcomings in knowledge search involves the creation of an intelligent Knowledge Navigator that is able to provide a consistent, user friendly, navigational metaphor across heterogeneous sources of content (text, cases, and databases) using the appropriate search engine technology for each source (see figure below). The navigational metaphor we are proposing is a dialog-based one in which the system prompts the user with questions that help them narrow the search for relevant information. Question generation is intelligent in the sense that the questions are posed relative to the current context of the search rather than as a static tree.



The major architectural components of the knowledge navigator we are proposing are:

1. Search Arbitration: This is the process of determining which source is the most likely to have content of relevance in solving the problem.
2. Search Refinement: Once a particular content source is selected as being potentially relevant, search refinement is the process of quickly either finding the relevant information or quickly determining that the relevant information is NOT in this content source.
3. Search Escalation: If the information is not in the selected content source, search escalation is the process of transitioning the search from one source to another. The key to good search escalation is in building a good query for the escalated search based on the information collected at the previous content source.
4. Navigational Learning: If useful information is found it is important to improve access to that information for future users. Navigational learning involves incrementally improving the search arbitration and refinement process based on user success and failure in previous search.
5. Knowledge Acquisition: As knowledge is frequently accessed in a less structured content source (i.e., documents or the internet), knowledge acquisition is the process of moving knowledge down the knowledge triangle towards more structured forms (like cases) that are more easily accessible by less informed users of the knowledge in subsequent searches.

Discussion

The Knowledge Factory and Knowledge Navigation are clearly dependent on one another. While the knowledge repository could contain a wealth of well formed knowledge, if a user cannot effectively navigate through the knowledge quickly and efficiently in the context of their problem, the knowledge is effectively useless. Similarly, allowing users to arbitrarily add knowledge to the repository without establishing an effective knowledge factory for all the different forms of knowledge content will defeat the ability of any navigational software to be effective in helping users solve problems.

The major point of this paper has been to emphasize the fact that problem solving knowledge comes in a variety of forms, not all of them text. Given this, it is critical that any knowledge management system that proposes to help people solve problems must figure out a way to create and navigate through this heterogeneous knowledge in a way that goes beyond using a word processor and a text retrieval engine, as is often the case today. We have proposed an architecture that enables the navigation through this heterogeneous content by using a common, dialog based search approach, with questions posed to the user in the context of their current problem solving needs.

References

Borron, J., Morales, D., and Klahr, P. Developing and Deploying Knowledge on a Global Scale. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference (IAAI-96)*, AAAI Press, Menlo Park, California, 1996, 1443-1454.

Davenport, T. H., and Prusak, L. *Working Knowledge: Managing What Your Organization Knows.* Harvard Business School Press, Cambridge, Massachusetts, 1997 (forthcoming).

Edvinsson, L., and Malone, M. S. *Intellectual Capital, Realizing Your Company's True Value by Finding Its Hidden Brainpower.* HarperCollins Publishers, New York, 1997.

Klahr, P. Getting Down to Cases: Seven Principles for an Effective CBR Strategy. *PCAI, V. 11, N. 1*, January/February 1997, 16-19.

Koldoner, J. *Case-Based Reasoning.* Morgan Kaufman Publishers, San Mateo, California, 1993.

Stewart, T. A. *Intellectual Capital, The New Wealth of Organizations.* Doubleday, New York, 1997.

Thomas, H., Foil, R., and Dacus, J. New Technology Bliss and Pain in a Large Customer Service Center. In *Proceedings of the 2nd International Conference on Case-Based Reasoning*, Springer, 1997 (forthcoming).