The Lotus Knowledge Discovery System: Tools and experiences

by W. Pohs
G. Pinder
C. Dougherty
M. White

The Lotus Knowledge Discovery System™ from IBM uses several leading-edge technologies to systematically create associations between corporate expertise and information resources, personalize and organize knowledge for individuals and communities, and provide a place for teams to work, make decisions, and act. It also creates a searchable index, computes document values, and provides a search-and-browse user interface. This paper describes the technology behind the knowledge management components of the product and describes the experience of the United States Joint Forces Command, one of the product's earliest adopters. The Joint Forces Command combined a new knowledge management methodology with the software tools, and the Lotus team gained valuable deployment experience in the process.

Over the past four years, a team of developers and product managers at Lotus Development Corporation has been working on a knowledge management suite of products. Based on its experience and on the existing large body of knowledge management literature, the team decided that knowledge management software should provide virtual "places" where users can organize information, services, and tools to support their particular needs, while at the same time maintaining and updating information in a more general context.

To support this concept, the team defined two distinct components of knowledge management: knowledge aggregation and information discovery. The team determined two principles that drive knowledge aggregation: (1) Individuals and teams need virtual places to work, make decisions, and act. (2) Virtual places should include applications, collaboration services, and personal services.

Individuals or teams define the requirements for these virtual places, but individuals do not always know *what* they do not know. Sometimes, just talking to an expert is the best way to learn. Information discovery is a way to provide access to all the information that is relevant in a corporate environment without prior knowledge of its existence.

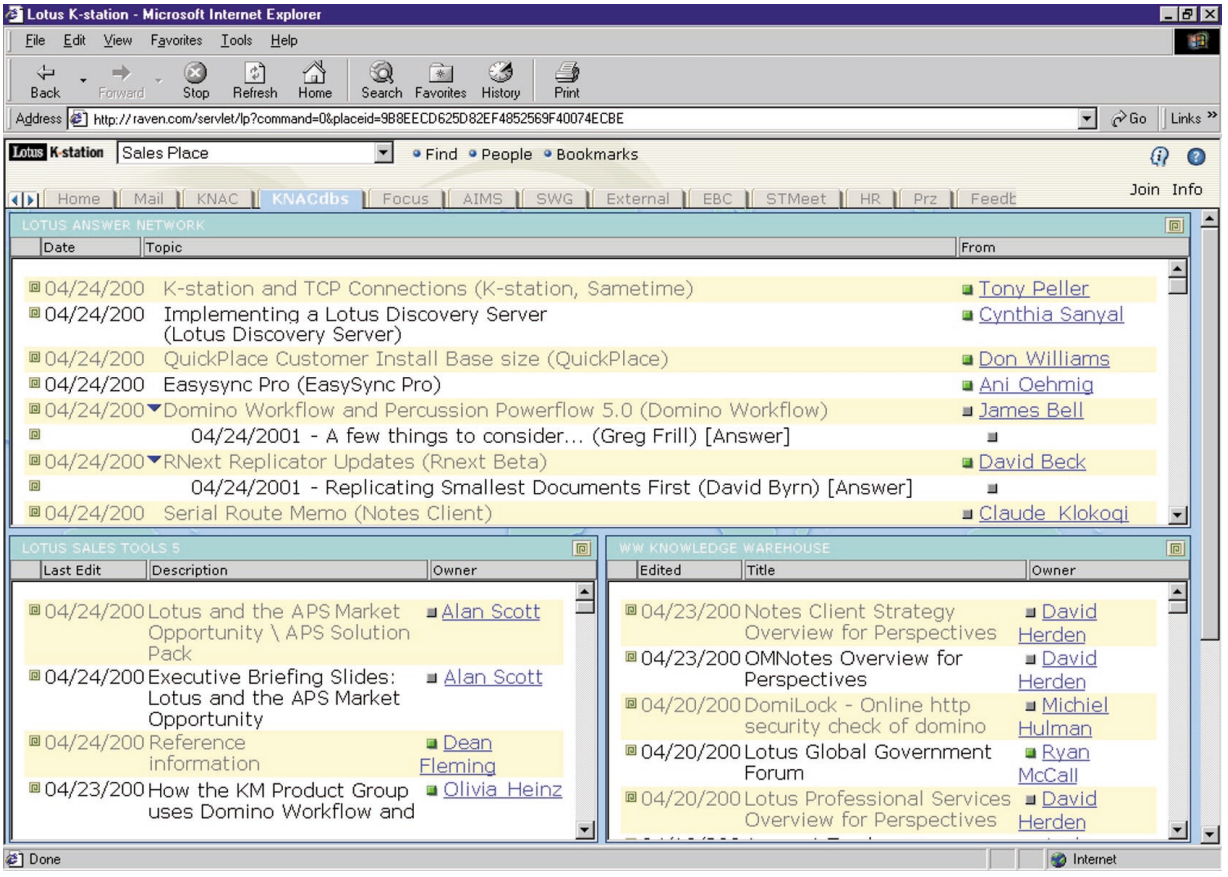
The development team decided that an information discovery server should:

- Automatically find, organize, and map disparate content
- Build a network to locate subject matter experts
- Add value to content by maintaining its context and by incorporating the opinions and judgments of individuals

Using knowledge gained from software development and research groups at Iris Associates, Lotus, and IBM, the team developed a set of components that could be used to convert these theoretical knowledge management tenets into practical corporate applications. The Lotus K-station* portal and the Lotus Discovery Server* are the results of this work.

0018-8670/01/$5.00 © 2001 IBM

Figure 1 A K-station can include portlets that point to useful applications.



The K-station portal

The K-station portal organizes all of a user's information, applications, and contacts by community, interest, task, or job. Users create a personal place by selecting from a list of preconfigured "portlets" (e.g., mail, calendar, discussions, to-do items, team rooms, custom applications, and Web sites). Portlets can support any Domino* or non-Domino applications. Each user's place provides access to a list of other public places that users can join. Figure 1 illustrates a K-station place.

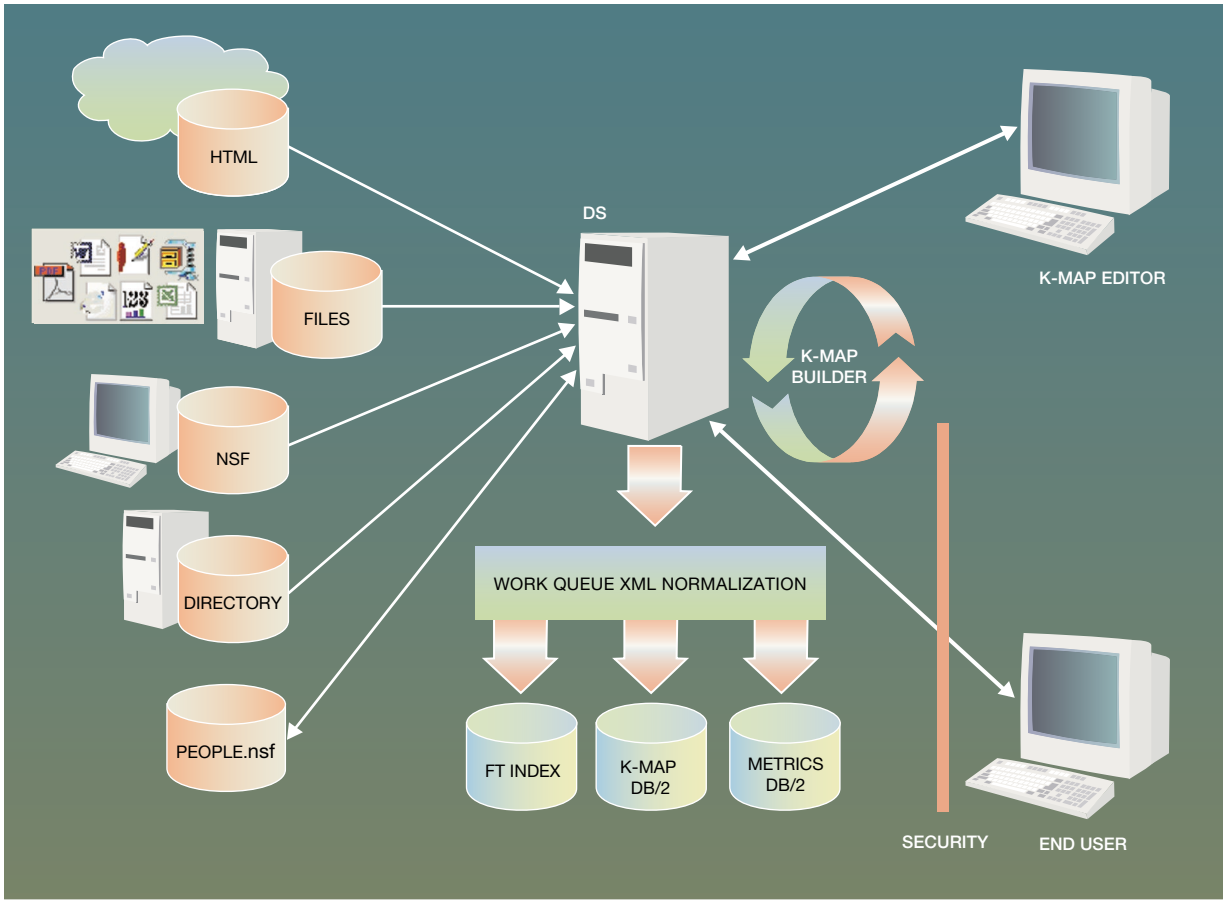
The K-station portal includes multiple places that can be defined by users, created by departmental or enterprise IT (information technology) departments, or developed and shared by colleagues. Community places (e.g., a "personnel review place" or a "new product brainstorming place") are activity-based. In these places, users monitor project status and participate in decision-making. For example, a sales place might include a sales-results reporting application, an information retrieval application, a list of sales tools, and a list of marketing staff available for consultation.

The K-station portal saves time by introducing users to the persons, applications, and information assets they need to be productive in their jobs.

The Discovery Server

The Discovery Server comprises a large, active content catalog and a set of services that collect, collate, and maintain information in the catalog. The catalog is an index to the written information and expertise that exist within an organization. The server regularly refreshes the catalog by tracking content, user interests, and usage activity. It delivers a great

Figure 2 The Discovery Server architecture includes both data and process components.



deal of information about an organization in terms of where things are, who knows what, what is important, and what subjects generate the most interest and activity. Primarily, it gathers existing corporate documents (in Notes* databases, external Web sites, and files on the intranet), creates several representations of this information in XML (extensible markup language) in a full-text index, in document clusters, and in a user profile database), and provides several user interfaces to display and maintain the information in various ways. The information is stored in a DB2* (DATABASE 2*) database, which is accessed by the various services whenever they need to display or maintain the data.

Figure 2 illustrates the components and processes used by the Discovery Server. Each of these components is explained in the sections that follow.

Spiders. The Discovery Server spiders are the "worker" tasks of the system. They run on a schedule set by the administrator. They either gather documents from the selected sources or monitor changes and deletions to the sources. Each supported data type (Notes databases, file system files, external Web sites) has its own customized spider. Discovery Server administrators complete on-line forms that describe each source to be included in the catalog. They provide information about where to find the source, and in the case of Web sites and file system files, about how many levels to retrieve based on links or subdirectories within the source.

The spiders adhere to source-level security and are good "net citizens." They report back, via a Discovery Server user interface, if they cannot access certain sources because of security restrictions. The spi-

ders create XML representations of the information they collect, and either add this information directly to the DB2 database or store it in work queues to be used later by other services. The spiders extract author, usage, content, security, and source location information from each document. They extract Unicode[1] settings to determine the native language of the source documents. The spiders register each document in the DB2 database. As each XML document is identified, DB2 returns a unique 16-character identifier for the document and all associated information obtained through subsequent Discovery Server processes. After registration, the XML documents are passed back to the spiders, which then transfer the XML output to three Discovery Server work queues: metrics, K-map builder, and K-map indexing.

Specialized versions of the Discovery Server spiders, called profile source spiders, gather information about individuals from Domino directory databases and LDAP (Lightweight Directory Access Protocol) server-compliant directories. Another spider examines Domino e-mail content but does not participate fully in the Discovery Server processes. E-mail content provides relationships between individuals and subject matter (affinities), but this content can never be published through the other Discovery Server services.

When scheduled, the e-mail spider connects to the specified Domino mail database and examines all sent and saved mail. The author, from, to, copy to, subject, and body fields are extracted and converted to XML using the process just described. The Discovery Server K-map indexing and K-map builder services do not directly process the e-mail content. Instead, the e-mail XML is forwarded to the metrics queue. The metrics process evaluates e-mail content to determine relationships to existing K-map category areas. E-mail content with no relationship to K-map category areas is ignored.

The K-map. The K-map is the backbone of the Discovery Server's search-and-browse user interface. It is accessed by users to locate content from many disparate sources, by "drilling down" through subject clusters, by using a full-text search, or by using a combination of both search strategies. Additional information about the relationships between individuals and document activity adds value and context to the user's search and retrieval experience. The K-map supports dynamic access to document security information when content is searched, to ensure that users view only the documents they are authorized to access. The K-map user interface (Figure 3) shows people awareness, affinities, and document values.

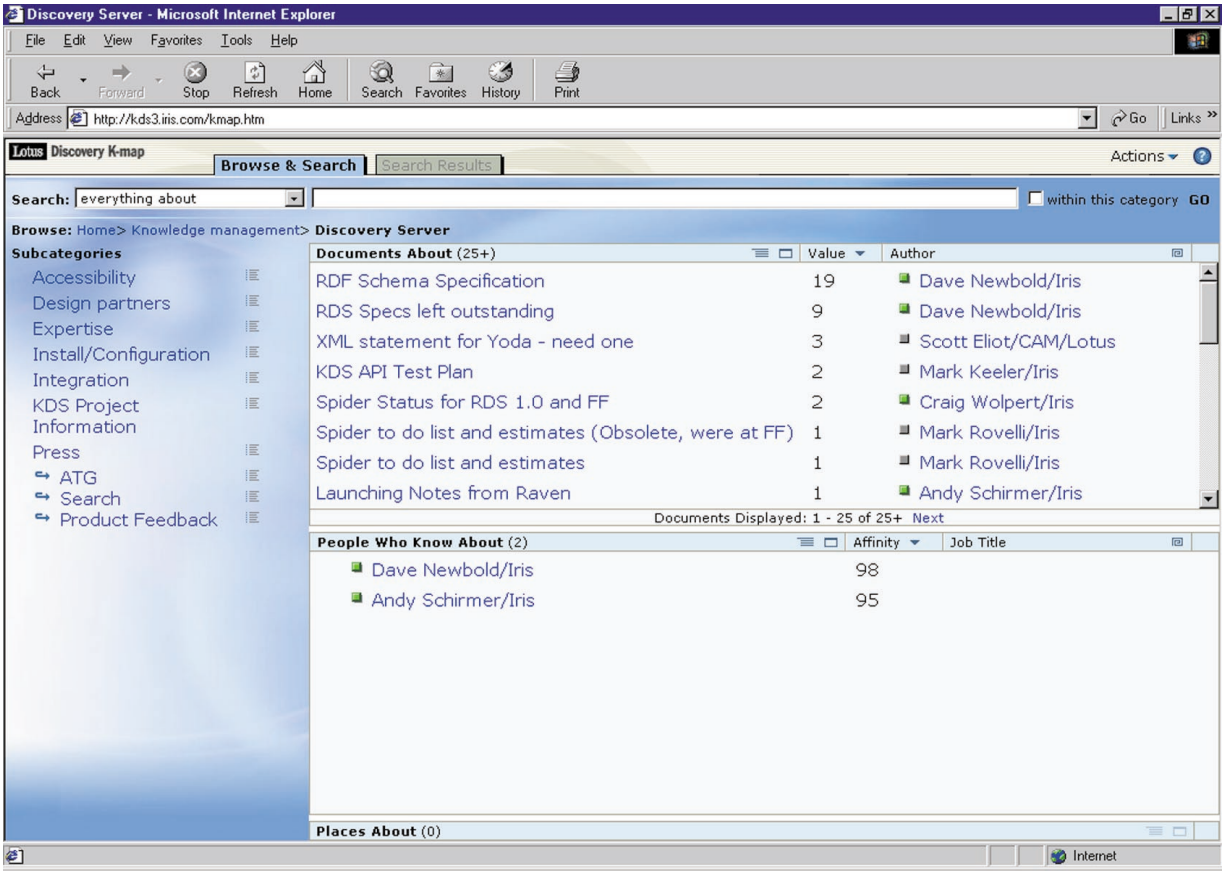
The K-map builder. The Discovery Server's K-map builder software statistically analyzes the words in documents to create groups of similar documents called clusters. Based on the Sabio software developed at the IBM Research Center at Almaden, California, the K-map builder treats words and phrases in documents as points in a large, multidimensional space. Each dimension corresponds to a single word or phrase and the number of times it appears. When two documents share many of the same words and phrases, they will be relatively close together in this space, and will appear in the same document cluster.[2] The K-map builder builds document clusters, creates labels for these clusters, and classifies new documents into existing clusters. It also identifies documents that do not fit into any existing clusters.

The K-Map is the backbone of the Discovery Server's search-and-browse user interface.

The K-map builder uses a combination of EM (expectation-maximization) and K-means clustering techniques to build the initial clusters, and the SVM (Support Vector Machines) classifier for categorization. These techniques are good at finding general themes in collections of documents, but almost always require some manual reorganization of the clusters. The K-map builder divides information into eight to ten clusters, and then subdivides these clusters into subclusters, generally four levels deep. Once the initial set of clusters has been created, the K-map builder's classifier compares the words in new documents to the words in the documents in the clusters it has already created.

The K-map builder makes two passes through the data: one to create the clusters, and another to create labels for the clusters. The labels are derived directly from words in the text and titles. These may not be the same words that will be used to ask for

Figure 3 The K-map user interface shows people awareness, affinities, and document values.



information. Human editors rearrange and relabel the clusters to reflect more commonly used search terms that may not appear in the actual text. The labels are also used to define subject affinities between individuals and documents in the system.

The K-map builder adds new documents to the K-map as new documents are added to the sources searched by its spiders. It classifies new documents by comparing them to documents in the existing clusters. When editors move documents to different clusters, new documents with words that are statistically similar will be classified into these different clusters.

Full-text indexing. The Discovery Server search engine is an implementation of IBM's global text retrieval (GTR) engine. The GTR engine uses n-gram technology, which creates an index by breaking words down into "grams," or strings having a uniform number of characters, to facilitate quick and efficient searching. The optimum number of characters in the string (the "n" in n-gram) varies in different languages: for English and other Latin-derived languages, it is three characters; for Asian languages, it is two. GTR supports many languages and has the ability to index documents that contain text in more than one language by adhering to the standard known as Unicode, thus making it a good choice for a product intended for international use.

"Fuzzy" search and stemming[3] capabilities are features of n-gram engines. Because the engine is simply looking for matches between strings of characters rather than trying to make sense of the words that make up the query, keeping track of partial matches is easily accomplished.

The K-map user interface displays a search score (a number between 1 and 100) next to the title of each found document, and lists the documents in descending order. A document with a score of 100 has the most relevance to the search terms. The search engine uses several methods of ranking, some traditional and others somewhat novel. In general, the ranking depends on the size of the document being searched, the number of matches, and the location of the matches within the document. For example, a large document with two matches, one in the beginning and one in the middle of the document, is assigned a lower score than a smaller document with two matches at the beginning of the document. The search engine also uses statistical data on word usage frequency to make sure that words like "a," "the," or "of" have less weight than those that are less frequently used.

Document usage patterns are then factored into the scores using information collected by the spiders on how many users have accessed each document and on how many links there are to and from the document. All of these measures are adjusted each time the spiders are run and the index updated.

Gathering information about individuals. The Discovery Server builds and maintains user profiles in a repository that can be queried directly to locate experts by skill, experience, project, education, and job type. The profiles are created either by drawing demographic data from any LDAP server or Domino public directory, or by mapping fields from other, specific applications such as team rooms, discussions, and project tracking. The Discovery Server then uses the metrics processing tool to determine relationships between known clusters and user activities. Affinities are relationships between individuals and clusters in the K-map. The relationships are based on user actions such as reading a document, authoring a document, responding to a document, editing a document, and creating a link to a document. When a user's interaction with documents reaches a predefined threshold, the metrics processing service sends an e-mail notification, including a link to the proposed affinity information, to the user. The e-mail notifies the user of the affinity proposed by the Discovery Server, and requests confirmation and approval to publish the affinity in the searchable user profile. In this way users control the information that is publicly available about them. Figure 4 shows a user profile with affinities.

Making it work

The K-station and Discovery Server provide ways for organizations to develop customized solutions to specific knowledge management problems. Users and communities can aggregate important information and customize their workspaces using K-station, and then drill down more deeply when necessary by using the search and browse capabilities of the Discovery Server. The Discovery Server automatically collects the judgments of individuals by analyzing their actions, and presents these judgments in context at search time.

Changes are gathered and tracked by the system and the K-map is dynamically revised. Implementation of these knowledge management components requires some up-front analysis by content managers, but the result is a continuously updated information aggregation and discovery system.

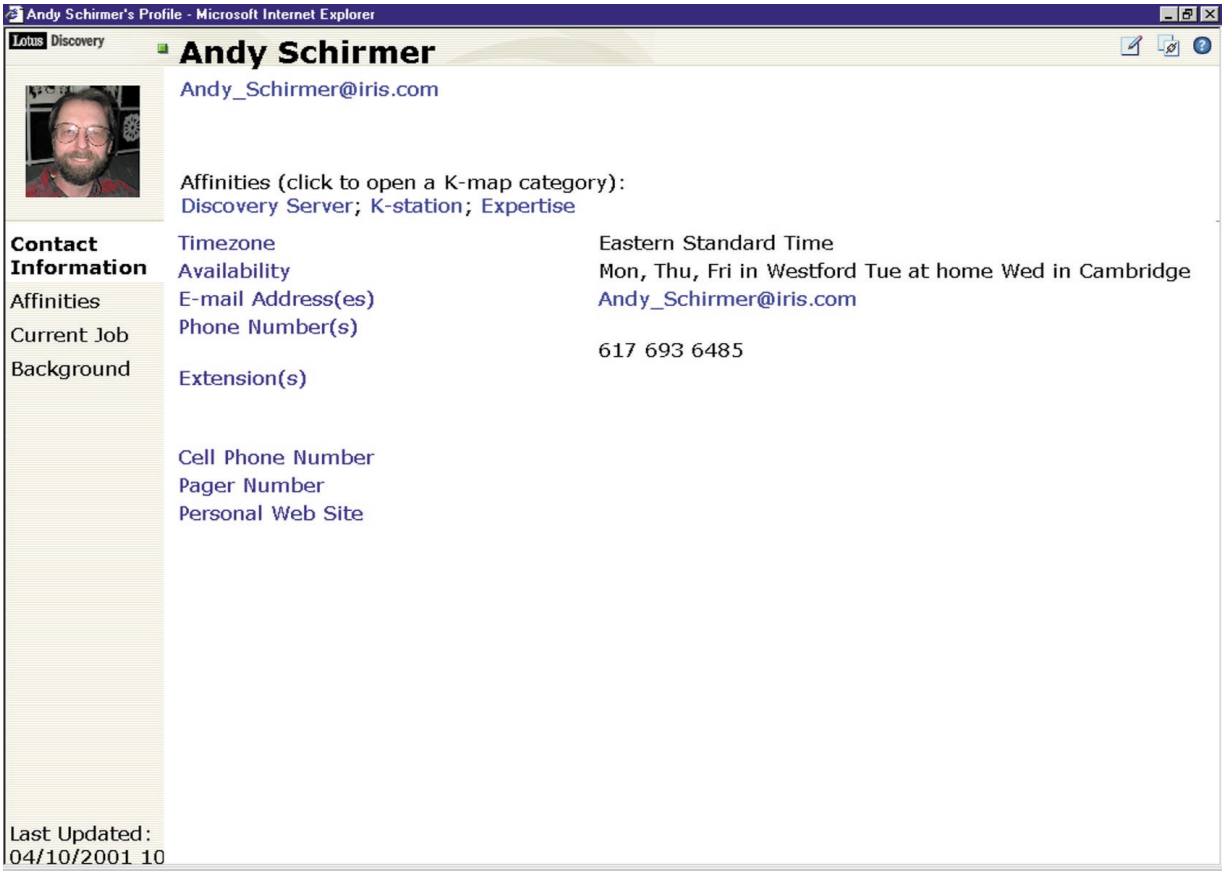
But how does all this work in practice? While the Knowledge Discovery System was being developed, Lotus worked with a small set of design partners who tested early versions of the software in real-life environments to solve real organizational problems. One of these design partners was the U.S. Joint Forces Command.

The Joint Forces Command case study

In the movie *Saving Private Ryan*,[4] Captain John Miller neutralizes enemy gunfire on a Normandy beach on D-Day (June 6, 1944) and then reports to his commanding officer. His report states that inaccurate or missing information regarding the arrangement of enemy forces along the beach led to the loss of many lives. Captain Miller was able to overcome the enemy only by utilizing an established military doctrine, employed by the Allied Forces, that dictated the use of an overwhelming number of troops when the battlefield configuration was unknown. The assumption was that superior numbers would prevail—with the risk of losing a large number of lives.

With the advent of the Cold War and the proliferation of nuclear weapons following World War II, few if any military commanders have had to decide how many soldiers are to be put at risk to prevail against an unknown battlefield configuration. The potential horror of, and the deterrence provided by, nuclear weapons made war on the scale of the Normandy invasion unlikely.

Figure 4 The user profile contains publicly available information about the user's affinities.



In the post-Cold War era, the mission of the U.S. military is to develop its capabilities to fight and win (and thus deter) regional conflicts with minimal loss of life. Rather than face a politically unacceptable war of attrition, the U.S. military is now developing the necessary means to know and understand exact battlefield configuration. With this knowledge, an optimal strategy can be devised, which will minimize the number of troops required and the associated risks.

The United States Joint Forces Command (USJFCOM) was set up by Congress in May, 1998, to lead the transformation of the U.S. military to achieve "full spectrum dominance" in the post-Cold War world. An integral component of USJFCOM's mission is to conduct joint concept development and experimentation to maximize present and future military capabilities. The Joint Experimentation Directorate (JE) fills this role. JE's specific mission is to develop, explore, and assess new joint war fighting concepts using both organizational structures and emerging technology. JE uses a process of discovery, innovation, and experimentation to assess concepts and then passes recommendations for change to DOTMLPF (Doctrine, Organization, Training, Materiel, Leadership, Personnel, and Facilities).

Experimentation team. JE was established with six employees. Today it has over 280 staff members. In April, 1999, JE began the task of determining the type of technology infrastructure it would need to support its mission. A team was formed to carry out an initial requirements and systems analysis assessment. This eight-week phase concluded with a comprehensive set of requirements and a defined list of recommendations that included a Web portal, a digital library, on-line analytical processing (OLAP) search

tools, data mining tools, collaborative tools, semantic modeling, and data storage tools.

An executive team within JE validated these recommendations and assigned each to a priority tier. Tier 1 contained the Web portal, collaboration tools, and data storage. Tier 2 contained the OLAP search tools, data mining tools, the digital library, and semantic modeling tools. Tier 3 contained the remaining recommendations.

To attract the necessary investment to begin implementation, JE began building a business case for the tier 1 recommendations. In the process, JE realized that technology was not going to solve the business problem. Within JE, different organizations had different views of their own and others' responsibilities. Multiple organizations claimed responsibility for the same processes and tasks, while some processes and tasks had no ownership at all. A group could be performing the same work as another group with neither group being aware of the duplication. Considering all the reusable work that was being undertaken, none was actually being reused. Without an organizational transformation that included culture, policies, and procedures to overcome these business-level problems, the technology tools just would not be used effectively.

Knowledge management team. Around this time, key staff within JE began reading *Working Knowledge*,[5] one of the seminal works in the knowledge management field. Other readings and Web searches on the same topic led to the conclusion that the principles embodied by knowledge management (KM) represented the solution needed to implement the necessary organizational transformation. With this impetus, a JE knowledge management (JEKM) team was formed to oversee and lead the project. From this point on, the project continued on two interrelated tracks.

For the first project track, the JEKM team met almost daily to share newly discovered material or personal thoughts on KM. This collaboration continually revised the organization's approach until "knowledge management" was no longer considered a valid expression of their needs. Instead, what was needed was to consider knowledge as an environment, rather than a commodity. By treating knowledge as an environment, the JEKM team believes that the end result is a work environment where knowledge is pervasive in doing business. Thus, when faced with a problem, the first collective thought of an organization is from a knowledge perspective.

For example, suppose a high-ranking military officer conceives of a new military concept and suggests that JE conduct an experiment to test the concept (referred to as "concept X" for the purpose of this case study). In a knowledge environment, JE would first browse and search its existing knowledge base to see if concept X had been investigated in the past. If not, then have any related concepts been investigated? If yes, what was the outcome of that investigation? Was it tested? Were any lessons learned from that experiment? Who are the experts in the fields closely related to concept X? What are their thoughts on the viability of concept X? Using this knowledge environment and the knowledge-pervasive approach that it engenders, much can be learned about concept X without spending large amounts of time and money conducting an investigation "from scratch."

When faced with a problem in an environment where knowledge is pervasive, the first collective thought of an organization is from a knowledge perspective.

This approach fits very well with the latest thought in military doctrine alluded to in the introduction of this case study. When attacking a military position, the first step a military commander would automatically take, in this new military paradigm, would be to attain complete knowledge of the battle space—and then have that knowledge continually updated throughout the military engagement. With such knowledge, problems can be managed and optimized to minimize risk and maximize results.

The second project track continued along the technology path. The JEKM team conducted a full market study in March, 2000, to find products that would meet the goals of the tier 1 recommendations (Web portal, collaboration tools, and data storage).

The major requirements on which each product was measured were:

- Ability to locate subject matter experts

- Ubiquitous access to organization knowledge and competencies
- Access to data based on security level
- Knowledge capture and preservation without user involvement, with distribution across the enterprise
- Individual users allowed to customize information as needed
- Ability to share knowledge with users external to the network without technology or licensing issues

To prepare for Discovery System, organizations should ensure that the title and author name for all documents are accurate.

None of the products initially examined was able to meet all the requirements in an integrated way, though a group of individual products could collectively meet most of the functional requirements. As a result, JE considered taking a "best-of-breed" approach, which would mean committing large funds to integrate the disparate products. However, during this consideration, details about the Lotus Knowledge Discovery System began to surface in the trade press. After further research, and discussions with Lotus, the team determined that the integrated nature of the Discovery Server offered a compelling solution when compared with other nonintegrated product offerings.

Limited objective experiment with Lotus products. Beginning in October, 2000, JE contracted with Lotus Professional Services (LPS) to conduct a short "limited objective experiment" (LOE) to install the Lotus products under consideration. The goal of this experiment, monitored by JEKM and a JE project executive, was to determine whether or not the products functioned as advertised.

To make the experiment as realistic as possible, JE assembled a group of "knowledge experts." Their mission was to define a set of processes within JE to which the products could be applied as a test. The initial list contained more than 60 processes, but it was eventually shortened to 20. From that list, a number of the JE knowledge experts determined which four processes were the most appropriate for the LOE.

All four processes were large, and LPS decided to select a discrete set of activities from each process that could be supported for the experiment. After interviewing JE staff members, LPS compiled a list of suitable activities and scored each activity, based on an algorithm composed of breadth, reuse, capability, innovation, effort for LPS, effort for JE, and risk. From the scored list of activities, a subset was selected for actual implementation during the LOE. The activities ranged from creating a K-map (taxonomy) from a large shared file system to creating a synchronous collaborative portal to support experimentation strategy between internal and external staff.

Once the products were installed and stabilized, several customization efforts took place in parallel. Those efforts centered on creating a K-map using the Lotus Discovery Server and developing a set of K-station "places."

For the K-map, a subset of JE's public shared drive was copied to the test network from the production network. These documents were then explored by Discovery Server spiders. This first K-map produced some interesting observations and results.

Lessons learned. First, it was already well-known that the public shared drive followed no structure or document-naming rules and contained very little consistent document meta-data. The K-map reflected the unmanaged nature of the shared drive by surfacing many documents that had either no title or a meaningless title and by showing that many documents apparently shared the same author, when in reality the original document had been copied multiple times and modified for different uses by many authors. The result was that some document cluster labels in the K-map accurately reflected the concepts used by JE; however, it looked as though the documents in the clusters were wrong. The lesson learned by applying Discovery Server spiders to a shared public file system is that the meta-data contained within the files has a significant impact on the perceived quality of the K-map. To prepare for the Lotus Discovery System, organizations should consider instituting meta-data policies for all documents created to ensure that the author name and the title are accurate.

Second, while the system appeared to have done a good job of identifying similar documents and locating them under the same cluster label, the labels produced were often confusing. Lotus Discovery Server determines the label for a cluster of similar documents by looking at the words within those documents and deriving the three most commonly used

descriptive nouns. By using the K-map editor tool that comes with the product, an editor can rename the clusters, but this is a time-consuming process. The Lotus product team learned from this the importance of enhancing Lotus Discovery Server to provide, without human intervention, more "common-sense" labels that reflect an external taxonomy structure provided by a customer. This enhancement would allow the customer to quickly train the Lotus Discovery Server to meet the needs of the organization. Work in this area is ongoing and includes a team at IBM Research.

While there is a learning curve for creating a K-map, the rest of the customizations were relatively easy once the requirements were determined. The K-station requires only minimal developer intervention.

The effect of the products on JE is already being seen. Instead of relying on a human to act as a broker of information for an experiment, the technology is now the broker. Additionally, JE staffers now have a software environment that can support collaboration and provide quick access across the knowledge in the organization, rather than just the knowledge in a single functional group.

What comes next?

JE is nearing the end of the beginning of its journey toward knowledge pervasiveness and recognizes that the most difficult part lies ahead. Once the technology framework is established, the work of transforming JE into a knowledge environment must begin in earnest. Few—if any—organizations have attempted such a transformation. Some of the pieces of the puzzle are known—training, incentives, high-level support, hard work, and tolerance. Other pieces have yet to be identified and discussed. JE's success will be the first step in demonstrating that a knowledge environment can not only enhance an organization's efficiency but, in this case, also enhance the transformation of national defense.

The JE experience proved that a successful knowledge management implementation requires software tools and a corresponding knowledge management methodology. The limited objective experiment uncovered several interesting issues about K-map construction and customization that have been subsequently addressed by the Lotus development and deployment teams.

Acknowledgments

The entire Lotus Discovery Server development team contributed to this paper, but we would particularly like to thank Gayle Thiel, Lauren Wendel, James Goodwin, Andy Schirmer, Cynthia Regnante, and Kathleen Murphy.

Cited references and notes

1. See http://www.unicode.org.
2. R. Pool, "Natural Selection," *IBM Research Magazine*, No. 2 (2000); see http://domino.research.ibm.com/comm/wwwr_thinkresearch.nsf/pages/selection200.html.
3. "Stemming" reduces a variant word to a common root. For example, "knowing" and "knowledge" have a common root.
4. The movie *Saving Private Ryan* was written by R. Rodak, directed by S. Spielberg, and produced by S. Spielberg, I. Bryce, M. Gordon, and G. Levinsohn. It was released by DreamWorks Pictures on July 24, 1998.
5. T. H. Davenport and L. Prusak, *Working Knowledge: How Organizations Manage What They Know*, Harvard Business School Press, Boston, MA (1998).

Accepted for publication September 6, 2001.

Wendi Pohs *IBM Software Group, 1 Rogers Street, Cambridge, Massachusetts 02142 (electronic mail: Wendi_Pohs@iris.com).* Ms. Pohs works on the Lotus Discovery Server product development team. She concentrates on building and managing taxonomies for Discovery Server deployments. Prior to joining the product team, she was a user assistance manager at Lotus. She also worked on information retrieval systems at Digital Equipment Corporation and at the American Mathematical Society. Ms. Pohs received her B.A. and M.I.L.S. degrees from the University of Michigan.

Gary Pinder *IBM Software Group, 1 Rogers Street, Cambridge, Massachusetts 02142 (electronic mail: gary_pinder@lotus.com).* Mr. Pinder is an information technology architect with the Enterprise Knowledge Services group of Lotus Professional Services. He is responsible for the design and delivery of technology solutions employing the Lotus knowledge management product set. He has nine years of experience in information technology and has worked in multiple industries including investment banking, commercial banking, automotive, telecommunications, insurance, and federal and state government.

Christopher Dougherty *IBM Software Group, 1 Rogers Street, Cambridge, Massachusetts 02142 (electronic mail: christopher_dougherty@lotus.com).* Mr. Dougherty has been working in the technology field since 1988—as technical support, team leader, chief architect, and project manager. Before coming to Lotus Development he was employed by the World Bank as the development manager for an institutionally oriented customer relationship management system. As a senior consultant for IBM Lotus, he is responsible for working directly with Lotus customers, to understand their knowledge management requirements and to devise the plan that will move them toward the ultimate goal of

becoming a knowledge organization. Prior to this role, Mr. Dougherty worked extensively with the United States Joint Forces and the U.S. Navy's Atlantic fleet to deploy knowledge management tools on land and at sea.

Marianne White *IBM Software Group, 1 Rogers Street, Cambridge, Massachusetts 02142 (electronic mail: marianne_white@lotus.com).* Ms. White is a principal technical editor at Lotus Development Corporation. She edits and indexes documentation for Domino and other Lotus products. Prior to joining Lotus in 1998, she was a writer and editor at Aspen Technology and Digital Equipment Corporation. Ms. White received her M.L.S. degree from Simmons College, Boston, MA.