Chapter XVIII
Managing Knowledge in Organizational Memory Using Topic Maps

Les Miller
Iowa State University, USA

Sree Nilakanta
Iowa State University, USA

Yunan Song
Iowa State University, USA

Lei Zhu
Iowa State University, USA

Ming Hua
Iowa State University, USA

ABSTRACT
Organizational memories play a significant role in knowledge management, but several challenges confront their use. Artifacts of OM are many and varied. Access and use of the stored artifact are influenced by the user’s understanding of these information objects as well as their context. Theories of distributed cognition and the notion of community of practice are used to develop a model of the knowledge management system. In the present work we look at a model for managing organizational memory knowledge. Topic maps are used in the model to represent user cognition of contextualized information. A visual approach to topic maps proposed in the model also allows for access and analysis of stored memory artifacts. The design and implementation of a prototype to test the feasibility of the model is briefly examined.

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.
INTRODUCTION

The late Dr. Peter Drucker had declared that we are in an era of knowledge economy. Understandably, any efficient functioning of the knowledge economy must rely on managing knowledge well. Individuals, organizations, countries, and economies must manage their knowledge through appropriate mechanisms and policies. Knowledge management requires the capture, storage, and use of several types of information and knowledge. Researchers have noted that organizational knowledge may be accumulated and retained using several organizational memory systems (Haseman et al., 2005; Nemati et al., 2002; Nevo & Wand, 2005). Advances in information technologies have helped in accumulating knowledge, but the paradigms applicable to collecting and storing the knowledge do little to motivate its use by managers and decision makers. Accessing and using organizational memories have been a challenge because of the multifaceted nature of memories and knowledge sources. Moreover, characteristics of the decision tasks where the knowledge and memory support are used pose interesting challenges in designing and developing knowledge management systems. In this article, we describe a model that makes use of topic maps to navigate the lattices of organizational memory and employ appropriate knowledge management and analysis tools.

ORGANIZATIONAL MEMORY AND KNOWLEDGE GENERATION

Organizational memory (OM) is a stored collection of organizational history reflected among the many parts (Walsh & Ungson, 1991). It includes both stored records and tacit knowledge and covers the various facets of organizational tasks, employees, and their task environments (Argote et al., 2003; Choy et al., 2005; Lee et al., 1999; Nonaka & Konno, 1998). Because it can be a large and valuable repository of information and knowledge, several researchers have recognized the import of organizational memory in effecting organizational performance (Akgun et al., 2006; Brockman & Morgan 2003; Jennex & Olfman 2002; Ji & Salvendy, 2004; Lesser & Storck, 2001).

Ackerman and Halverson (2004), however, take a critical view of prior research on OM and argue for a theoretical base to properly define and empirically validate future research. They state that as socio-technical systems, organizations and their memories conform to social structures and norms while employing technical models. They use the theory of distributed cognition to develop a theoretical foundation for organizational memory. The basic tenets of this theory are that knowledge evolves from a community of practice and that cognition and inferences result from the shared meaning among the participants (hence the distribution) (Hollan et al., 2000). Communities of practice fulfill a number of functions with respect to the creation, accumulation, and diffusion of knowledge in an organization through exchange and interpretation of information, by retaining knowledge, by stewarding competencies, and providing homes for identities (Wenger, 1998). Collective thinking creates knowledge that otherwise would not be evident. Additionally, changes in state of the memory, as in change from internal to external representation via artifact changes or through the movement of information among the participants (trajectory of information), are necessary to fully utilize an OM. A cycle of changes comprising contextualization to decontextualization and again to recontextualization of the information object takes place as organizational members relive their experience through the stored information object or artifact. An essential feature of knowledge management systems is this capability to change the state of the information object.

Using empirical data and qualitative methods, Ackerman and Halverson (2004) illustrate ap-
plication of the theory of distributed cognition to validate the use of organizational memory in decision making. In order to build and test an organizational memory system to support knowledge management it is not sufficient to have a theory to test the effect of such systems on organizational performance. It is also necessary to have a design theory that would support the need for defining appropriate components upon which other theories can be tested. For example, a design component should be able to support the cycle of changes a stored information object might go through.

Markus et al. (2002) proposed the use of design theory (Walls, Widmeyer, & Sawy, 1992) to describe emergent knowledge processes, which are often found in situations similar to the use of OM and distributed cognition. Design theory brings together elements of user requirements, system features, and development guidelines into one package. Moreover, this set corresponds to an underlying theory (kernel theory), such as distributed cognition that may be tested empirically by examining the outcomes of the systems development for fit with the requirements. Design research proponents identify six nominal process steps, namely problem identification and motivation, objectives of a solution, design, and development, demonstration, evaluation, and communication (Peffers et al., 2006). The first four process steps are considered suitable for doing design research. In addition, when systems are built for testing and validating theory, the appropriate entry point for design research is indeed the design and development process. At this stage an artifact is built and theory is tested using the artifact. The main purpose is to yield a proof of concept of the theory and the system that supports it. The contribution of the present work is in the development of an organizational memory knowledge management model that supports the distributed cognition theory. The design and implementation are presented. We see the proposed model as providing an early step toward realizing the model proposed in Nilakanta et al. (2006).

As social and linguistic dimensions of knowledge become pertinent, users need appropriate tools to support the search and analysis of knowledge. Topic maps are one such tool.

**Topic Maps**

Smolnik and Erdman (2003) state that topic maps provide strong paradigms and concepts for the semantic structuring of link networks and therefore, they are a considerable solution for organizing and navigating large and, continuously growing organizational memories. The ISO standard ISO/IEC 13250 provides a standardized notation for interchangeably representing information about the structure of information resources used to define topics, and the relationships between topics. A set of one or more interrelated documents that employs the notation defined by this International Standard is called a topic map. The standard further states that a topic map defines a multidimensional topic space, a space in which the locations are topics, and in which the distances between topics are measurable in terms of the number of intervening topics which must be visited in order to get from one topic to another, and the kinds of relationships that define the path from one topic to another, if any, through the intervening topics, if any.

Topic maps provide a subject-based classification of resources where each resource represents a real world object. In topic maps, three constructs are provided for describing the subjects represented by the topics: names, occurrences, and associations. These describe the names, properties, and relationships of subjects, respectively. A name may be assigned to more than one topic and a topic may have more than one name. In addition by defining scope and types a name can become rich and complex. An occurrence links to one or more real knowledge sources. An occurrence, however, is not part of the topic map. Associations describe
Managing Knowledge in Organizational Memory Using Topic Maps

relationships among topics and are independent of the real knowledge objects. Associations add value through the relationships.

The Topic Map architecture is designed to facilitate merging topic maps without requiring the merged topic maps to be copied or modified. This feature makes it quite suitable for use in organizational memory systems. As Steiner, Es-smayr, and Wagner (2001) state, the construction of large, complex organizational memories takes place over extended periods of time, growing incrementally both in size and structure. The ISO standard describes how topic maps may be used in situations such as that may be found in OM systems. For example, a topic map can be employed “to structure unstructured information objects, or to facilitate the creation of topic-oriented user interfaces that provide the effect of merging unstructured information bases with structured ones. The overlay mechanism of topic maps can be considered as a kind of external markup mechanism, in the sense that an arbitrary structure is imposed on the information without altering its original form.” Knowledge management applications can also benefit from topic maps, especially when combined with ontologies. Korthaus, Henke, Aleksy, and Schader (2006) describe a topic grid architecture that enables a client application to view distributed topic maps of organizational knowledge as a combined virtual topic map. Each distributed topic map represents a different view of the underlying information.

OVERVIEW

We see the knowledge space of an organization’s memory as consisting of data (transaction database data, warehouse data, and memory artifacts), knowledge (employee knowledge as well as knowledge inferred from the organization’s memory), and tools (retrieval, analysis, and learning tools). Moreover, the knowledge space is distributed throughout the organization. Any model capable of managing the knowledge in such a space has to be flexible, scalable, and distributed. To meet the needs of such a space, we propose a model with the same characteristics. A block diagram of the proposed model is shown in Figure 1. It has been designed to provide support for accessing and analyzing the collective memory of an organization.

Topic maps that support access to an organization’s knowledge via a combination of search terms and tools serve as the user’s entry point. Note that the same knowledge space can take on different views based on the content of the topic map a user uses to navigate the space. To this end, we assume that the topic map framework in Figure 1 supports multiple topic maps. We see the choice of topic map as acting essentially like a view in that as the user logs on, the topic map framework presents the user with an initial topic map level that acts as a root to the topic maps that he or she is authorized to access. This approach was adopted rather than having a standard first
level to simplify the user’s access and restrict the topic map to the task at hand.

The framework that supports the topic map converts the user’s request into a Request object and passes it to the User Interface Controller. The Request Processor converts the request into a set of queries (if required) and a framework query. If the requested information is not currently cached in the Internal Object Data Store, the set of queries are passed to the Data Source Access Module to be transmitted to the appropriate data source. Results from these queries are cached in the Internal Object Data Store as the queries are completed. When all the required data has been cached, the framework query is activated by calling the appropriate tools and data types. The results are returned to user.

Objects are used as the internal unit for storing and communicating artifacts due to the artifact’s natural object structure. For example, the object structure of an e-mail is shown in Figure 2. The primary components make use of traditional wrappers and object views (Yen & Miller, 1995) that are used to support communication within the model.

**Object Views**

The object view type (Yen & Miller, 1995) is defined as being an extension of the object model. The views have a traditional object structure.
Managing Knowledge in Organizational Memory Using Topic Maps

Local Interface Views

The individual data sources are expected to have local control. The local interface view (Yen & Miller, 1995) is a view object type that is used by the local data administrator to provide a mechanism to make the local data accessible to the proposed model in object form. The local interface view allows distribution transparency and representation transparency, while hiding or converting (mapping) some of the data from the data source. The local interface view belongs to the data source. It interacts directly with the data source and passes the result to the wrapper which controls communication with other components of the proposed model. A given data source and its wrapper can support multiple local interface views in order to present its data in different ways to different applications or users.

Topic Map

A topic map is used in the model to provide users with visual access to both search and analysis. It is this interesting blend of semantic search and analysis that motivates its use in the model.

Here we define a topic map to be the directed acyclic graph \( T = (N,E) \), where \( N \) is the set of topic map terms, data types, or search/analysis tools. The directed edges in \( E \) are of the type \((n1, n2)\), where \( n1 \) is a topic map term or a data type and \( n2 \) is a topic map term, a data type, or a search/analysis tool. An implicit node, called the root, points to the topic map terms, data types, or tools that make up the first level of the topic map. Nodes with out degree zero are said to be leaves of the topic map. Leaves point to the information in the organization’s memory and carry any search terms or data types accumulated on the path from the root of the topic map to the leaf. Note that leaves that are either search terms or data types make use of an implicit search tool. In addition each nonleaf level of the topic map has a search tool that can be used to initiate a search based on the topic map terms that have been traversed to get to the topic level or arbitrary keywords entered by the user.

Data Types

The data types supported will depend to some extent on the nature of the organization. Typically, the data types will include the people that the organization relies on for expertise, transaction data, warehouse data, the artifacts that make up the organization’s memory, and knowledge inferred from these data. The typical artifacts are e-mails, meeting minutes, reports, tutorials, news releases, Web pages, and presentations. Extending the supported data types is simply a matter of ensuring that the appropriate object types and tools for search, analysis, and learning are available in the set of supported tools.

Data Sources

A block diagram of the layout of a data source is shown in Figure 3. The query string for a data

```java
public class E-mail{
    private Person sender;
    private Set<Person> receiverList;
    private String subject;
    private String body;
    private Collection<Attachment> attachments;
    ...
}
```

(attributes and methods) with the restriction that they support a derivation method. The derivation method is used to generate the public and private attributes of each object instance created through a local interface view.
source from the request processor is translated to
the appropriate request string and passed to the
local interface view (LIV) via the data source
wrapper. The LIV interprets the request string
and passes it to the data source and converts the
response into the objects defined by the LIV view
type. Note that the LIV may make use of local
analysis and learning tools to process the data
returned from the data source before returning
it to the Internal Object Data Store. The set of
result view objects are then passed back to the
Data Source Access Module and cached in the
Internal Object Data Store for analysis.

Internal Object Data Store

The Internal Object Data Store is based on our
earlier work on object-oriented data warehouses
(Miller, Honavar, Wong, & Nilakanta, 1998;
Miller, Lu, Zhou, & Hurson, 2000). The Internal
Object Data Store combines tools and cached
data to allow in depth analysis of the data in the
knowledge space. The tool set consists of a set of
built in (kernel) tools for managing tools and data,
as well as a set of user defined tools for analyzing
or learning from data (Miller et al., 1998).

User Interface Controller and Topic
Map Framework

The division between these two components in
an implementation will depend on the nature of
the tools used to set up the system. For example,
in the case that the topic maps and results are
delivered to the user via an Internet browser, the
Topic Map Framework represents the HTML and
JSP pages. The User Interface Controller would
be implemented using a Web framework package
such as Goodwill and Hightower (2003). For an
implementation using a Java client as the user
interface, the separate roles of the two components
will be handled by different classes.

Request Processor

The request processor receives a Request object
from the User Interface Controller that defines
the required search criteria and the result format

Figure 3. Data source node layout and request/data flow for retrieval

Figure 4. Polymorphic class hierarchy for two tools that extend the Request class
required to complete the request. The tool type and action is built into the class structure by employing the Polymorphic design pattern. The Request, shown in Figure 4 as the super class, provides the basic request attributes and behavior. Each tool that is added either implicitly or explicitly to the topic map requires the definition of a class that extends the Request class. These new classes add any additional information/actions required by the tool. Figure 4 shows the class hierarchy for two tools.

**IMPLEMENTATION**

To illustrate the functionality of the current implementation of the proposed model, we walk through a sample session and point out some details as the session unfolds. The current implementation has been implemented in Java with everything but the data sources being on a single machine. Our next version will expand this view to include the use of a Web browser as a user’s client for accessing the organization’s memory.

Figure 5 illustrates the initial view of the organizational memory topic map seen by a user as he or she accesses the system. The shape of a symbol on the screen provides an indication of how the topic or tool is used. Tools that provide more than basic information retrieval are represented by rectangles, while ovals indicate that the topic points to another level in the topic map or that the default information retrieval tool can be reached from the oval. The default information retrieval tool uses the path label to determine the domain (e.g., artifact type) or the search criteria (e.g., any nontool topics in the path label). In Figure 5, the *Find Expert* rectangle indicates the existence of a tool that maps areas of expertise to employees with the desired expertise. The user also has access to the default information retrieval tool by clicking on *Search By Arbitrary Terms*, *Committee Search*, or *Tutorial Search*. The difference between the three nodes is the domain that will be searched. Clicking on the *Search By Arbitrary Terms* oval will provide a keyword search of all artifacts stored in the organizational memory, while clicking on either of the other two ovals will provide a keyword based search on the domains indicated within the oval topic name, that is, committee information or tutorials, respectively. The *Search By Topics*...
Managing Knowledge in Organizational Memory Using Topic Maps

A oval represents the root of a subtopic map based on the general topic terms.

The Search Memory Collections oval points to the set of supported organizational memory data types. Figure 6 shows the result of clicking on it. Selecting one of the memory type ovals (e.g., e-mail, minutes, reports, etc.) provides access to the next level of the topic map that focuses on the options available for the artifact chosen. The Search By Arbitrary Terms oval at the bottom of the screen shown in Figure 6 gives access to the same search tool that was available on the first level. It gives the user the ability to select relevant artifacts on the basis of any terms from the artifact types identified on the screen. In our current implementation this tool makes use of a frequency based search as described in Hu, Miller, and Nilakanta (1998).

Figure 7 shows the next level in the organizational memory that the user moves to by clicking on

---

**Figure 6. Artifacts available by selecting the Search Memory Collections topic on the first level (Figure 5)**

**Topic Map Organizational Memory**

---

**Figure 7. Topic map level reached by clicking on the Email oval in Figure 6.**

**Topic Map Organizational Memory**
the E-mail oval from Figure 6. Note that the path label at this point consists of the Search Memory Collections and E-mail topics and is used to restrict the domain of any of the tools to the e-mail domain. Because the only topics in the path label are designed to restrict the domain, the tools will have to provide their own search criteria.

A user can always return to the top level of the topic map by clicking on the home icon at the top of the page. Navigation between pages that have already been traversed can be done by using the forward and backward arrows at the top of the page.

A key feature of any system designed to provide search and analysis of artifacts is that the tools have to be constructed so that the user can easily grasp the results of the search and analysis. To accomplish this it is useful for the tools to be interactive and make use of visual displays of results. We conclude this section by examining a simple e-mail tool that gives the user the ability to choose a set of search terms, a start and stop date, and the number of intervals. Figure 8 shows the screen segment for the results of this tool. The bar graph indicates the relative number of e-mails that matched the search condition for the intervals. The use of the tool is instructive in terms of what occurs within the prototype of the proposed model. The tool represented by Figure 8 is obtained by clicking on the Frequency Analysis rectangle at the bottom of the screen shown in Figure 7. Once the user clicked on the rectangle for the tool, the Topic Map Framework (Figure 1) would send the request to the Request Processor via the request class that extends the Request class for the e-mail tool. The system prompts (not shown) the user for the search terms, start and stop dates, and number of intervals. Submission of this request generates a search of the data sources that contain the relevant e-mails would take place and the results would be sent back to the Internal Object Data Store to be held there. The framework query for the tool's request creates the bar graph and presents it to the user. Our current implementation of the tool allows the user to click on any of the bars to see a list of e-mails that fall into the time interval.

Figure 8. Sample results for number of emails over fixed size intervals using days as the time unit
represented by the bar. The user is then able to choose e-mails from the list to see the full text of the e-mail. Note that each time that the user operates on the data represented by the bar graph, the system only needs to go back to the Internal Object Data Store for the cached data.

**CONCLUSION AND FUTURE CONSIDERATIONS**

A model of an environment for using topic maps to access knowledge from an organization’s collective memory has been presented. The retrieved e-mail collection is representative of the collective thinking which creates knowledge that otherwise would not be evident. Additionally, changes in the state of the memory, as in change from local data source to the object store illustrated the trajectory of information and are necessary to fully utilize an OM. A cycle of changes comprising contextualization to decontextualization and again to recontextualization of the information object takes place as organizational members relive their experience through the stored information object or artifact. An essential feature of knowledge management systems is this capability to change the state of the information object. An initial version of the model has been implemented using Java as the programming language, Poet to store the objects, and Soap to communicate to data sources external to the system.

Future releases will make use of either Java Data Objects or a tool like Hibernate to replace the Poet database system. In the next version of the system, a Web interface will be developed based on the java versions of the tools supported in the current version.

**REFERENCES**


Managing Knowledge in Organizational Memory Using Topic Maps


**ENDNOTE**

1 An earlier version of this article was presented at HICSS-40.

*This work was previously published in the International Journal of Knowledge Management, Vol. 4, Issue 1, edited by M. Jennex, pp. 17-28, copyright 2008 by IGI Publishing (an imprint of IGI Global).*