December 2001

Organizational Memory: An Implementation Approach

Pankaj Hyde  
*Southern Illinois University at Carbondale*

Micki Hyde  
*Southern Illinois University at Carbondale*

Follow this and additional works at: [http://aisel.aisnet.org/amcis2001](http://aisel.aisnet.org/amcis2001)

Recommended Citation  
[http://aisel.aisnet.org/amcis2001/235](http://aisel.aisnet.org/amcis2001/235)

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2001 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
ORGANIZATIONAL MEMORY:
AN IMPLEMENTATION APPROACH

Pankaj
Southern Illinois University at Carbondale
pankaj@siu.edu

Micki Hyde
Southern Illinois University at Carbondale
mickih@siu.edu

Abstract

Organizational Memory (OM) models have existed for some years now. While these models provide a good conceptual framework, actual implementation is inherently complex. Another apparent drawback of these models is that they remain conceptual in nature and do not attempt to cross the bridge to implementation. This paper models OM as an instance of distributed computing application and uses abstractions to suggest an implementation in a component-wise fashion. Such an implementation approach will enable practitioners to implement the OM components using the best technologies available and suited for the components. Using the argument for a component-wise implementation of OM using the best available technologies, the implementation of an indexing scheme for OM using an enterprise directory service is proposed.

Keywords: Organizational memory, OM model, indexing, directory services, implementation

Introduction

Organizational Memory (OM) has been steadily gaining importance for organizations as they realize the benefits of experience and past learning in staying competitive, reducing transaction cost and making better decisions (Stein and Zwass 1995; Walsh and Ungson 1991). At the same time, the contribution of Information Technology (IT) in organizational functions has grown to an extent that IT is seen as an integral part of any organizational initiative. It is natural to assume then that IT will be providing support for OM. It is not clear if this is happening on an explicit level. The predominant framework for OM implementation found in the literature search is by Stein and Zwass (1995). This framework is more conceptual in nature and it provides a good starting point for implementing an Organizational Memory Information System (OMIS) but it fails to bridge the gap to implementation. A recent case study of the development of information systems (IS) found that the support for the development of OM is one of the requirements not currently satisfied by IS (Soderquist and Nellore 2000). Thus there is a need for an OM model and framework that can aid in the development of IS to support OM. We present in this paper an implementation-focused model and framework for the IT support of OM. Sufficient conceptual grounding for OM and related areas exists currently to warrant an approach that is specifically implementation oriented.

Another motivation for an implementation-oriented model and framework is drawn from the argument that in conceptualizing IT support for a concept it is often the IT that will constrain the application of a concept. This is applicable in the context of OM, which incorporates the complexities of organizations, cognitive psychology, sociology, and information technology. A full-fledged OMIS cannot be implemented using the technologies currently available (the reasons are several but are not explicated here). Making the traversal from the ‘esoteric’ theoretical model or to an implementable design when technology poses constraints can be achieved using components that are well abstracted. A component can be implemented using the technologies available and can evolve as technologies evolve without disrupting the remainder of the system due to well-defined interfaces that it presents to the remaining components.

1A full-fledged OMIS is very broad in scope. It will involve explicit management of the five OM bins (Walsh and Ungson 1991) thorough the use of IT. IT will provide support for support all the mnemonic functions (Stein and Zwass 1995) needed for the management of these bins and their contents.
To make the transition to implementation, this paper presents OM as a distributed computing application running over the enterprise computer network. The discussion of the model uses abstractions that are frequently applied in the area of programming languages and network models (OSI 7 layer model). Consequent to this model the paper discusses the implementation of one component, which is the indexing scheme for OM using currently available enterprise directory services. We discuss some of the technical requirements for such a directory. Due to constraints on the length of this paper, the detailed discussion remains for subsequent work.

### An Organizational Memory Model

OM “is a convenient metaphor that can be used to define the information and knowledge known by the organization and the process by which such information is acquired, stored and retrieved by the organization members” (Anand et al. 1998). Another definition states that, OM relies on knowledge that is spatially distributed throughout the processes, individuals and artifacts of the organization and beyond its boundary (Stein and Zwass 1995). OM is an important part of organizational decision-making due to its contribution in areas such as organizational learning, providing case-based epistemic memory, etc. It is also an important part of Organizational Decision Support Systems (ODSS). It is implicitly and explicitly part of many ODSS definition, models and implementations (Joey 1991; Watson 1990; Swanson 1990).

Synthesis of the concepts from OM definitions introduced here indicates that OM is knowledge/information stored in some form (including cognitive representations) in people or artifacts together with the procedures to maintain the knowledge/information. People capture the shared interpretations/values—a vital part of organizational memory and not stored effectively by artifacts due to the implicit nature of these values/interpretations. We can identify two types of ‘people’ entities and two types of ‘artifact’ entities. People can be individuals or groups. In forming links with group, the link is often through an interface that the group presents. This interface may be a single member of the group, more than one member of the group, or through an artifact (like reports written by the group). Artifacts serve as complements to people and help to reduce the cognitive load by storing information in greater volume and detail; archive information that is no longer current but is needed for historical reasons; store knowledge/information in a pre-emptive mode so that it can be used when a need arises; and aid in new knowledge/information creation. From an IT point of view artifacts can be categorized into those that are stored on computer media and those that are stored on non-computer media. Artifacts include objects like documents, databases, videos and audio records, etc. Artifacts that

---

3We use information and knowledge interchangeably since the focus on this paper is on explicit knowledge.

4Processes are not included explicitly as a process can be viewed as a synthesis of people and artifacts connected in an organizational context together with cause and effect knowledge for activities that are part of the process.

4Even though artifacts on non-computer media are decreasing as a result of advances in IT they still account for some portion of the knowledge/information present in the organization.
reside on computer media may be visualized as a collection of objects (from a programming viewpoint) since they would represent both content and behavior. Non-computer media would include paper documents, videotapes, audiotapes etc. The form and media of artifacts may differ across situations and organizations.

The information/knowledge that resides in people and artifacts aggregates to OM when it is linked seamlessly to the information/knowledge stored in other people and artifacts throughout the enterprise and/or beyond it. Linkages are important for sharing/exchange of information/knowledge, and for creating and supporting shared values/interpretations. Linkages are functions of the organizational context (e.g. how far outside the organizations these linkages go will be influenced by number of partners and the nature of partnerships) in which the people and artifacts exist (Anand et al. 1998). We propose at this stage that the use of linkages in an effective manner requires a directory service of some sort. A graphical representation of the model is in figure 1.

Thus one may visualize OM as a collection of entities (people and artifacts representing the information/knowledge stored in them) distributed both within and outside the organization that are linked through an efficient indexing-mechanism/directory structure for referencing. Acting upon this distributed knowledge/information will be a set of processes for creation, storage, retrieval, dissemination, and maintenance activities. Agents will execute these processes. Agents include people, IT applications, or a combination of both. A person will have his/her own (primarily implicit) processes to retrieve, acquire, store and maintain the knowledge/information contained within him/her. In an IT supported OM, whenever the information/knowledge leaves the boundary of a person it will be first converted into an artifact for purposes of an exchange through the computer network

In an IT implementation, the model in Figure 1 represents a distributed computing application that connects computer artifacts and people (using IT applications to interact with each other and the artifacts) through a computer network. People and artifacts form the repository for storing information/knowledge/cases etc. that form the part of the OM in both explicit (artifact) and implicit (people) form. The agents that execute various processes on this repository include IT applications executed and controlled by people; and intelligent IT applications like expert systems, S/W agents, self-executing scripts, etc.

The model of the OM presented above is presented as a top-down model and is geared toward implementation. Various technical capabilities are needed to implement the model. For example, while a physical structure of links is easiest to implement, other details that are important and necessary are models for content (information/knowledge) representation; and definitions and details for the processes of retrieval, acquisition, storage and maintenance (Anand et al. 1998; Stein and Zwass 1995). While these are important components of the model and the model cannot be complete without their inclusion, implementation can be less painful and more successful if the model is implemented piece-by-piece using the best of the breed technologies available. The abstractions from a distributed computing application can aid in this task and are presented in the next section.

Organizational Memory Components: Implementation Perspective

For implementation purposes the organizational memory can be viewed as a distributed processing/computing application running off a networked computing infrastructure. The networked computing infrastructure has become the predominant medium of information exchange in the organization. The networked computing infrastructure provides links with unique addresses for artifacts and people within and outside the organization (e.g. TCP/IP address of computers, URLs for artifacts and email addresses for people) and can form the physical implementation of the links between the entities (people and artifacts) in an organization. Other layers are added on the top of the physical links to yield an OM implementation model.

Figure 2 shows a simplistic model for OM abstracted into four layers. These layers are Link Infrastructure, Directory Services, Applications, and People. Each of the layers provides some services to the layers above it while hiding the implementation details. A layer may also pose some constraints on the capabilities of what can be achieved by the layer above it, e.g., with the lowering of the cost for storage it may be possible to store all the video artifacts on computer media but the bandwidth capabilities of the network may constrain the delivery of this video to the users.

Link infrastructure is composed of entities and links between these entities. The computer network elements (routers, bridges, wires etc.), hardware, and the operating system and utilities implement the physical link structure for the OM. The link infrastructure provides a set of services to the next layer, which is that of directory services. Directory services provide several features including those postulated in Anand et al. (1998). Directory services implements a universal indexing scheme and one logical directory for the universe of interest. The entities at each end of a link are identified uniquely within and outside the

---

*This is similar to the abstraction in the OSI seven-layer model for networking*
organization through the services provided by the directory services. Directory services maintain names and address of all the artifacts and the people that are accessible through the computer network. Agents discover the existence of entities using the search facilities in the directory services. Directory services provide a link to entities external to the organization and provide a link into these external organizations through the use of a uniform/standardized metadata and use of standards like Light Weight Directory Access Protocol (LDAP 3). Agents (composed of people and IT applications) provide a set of processes that act on the OM entities included in the link infrastructure.

Some Implementation Issues

From an implementation perspective for OM people, entities may be treated as a black box with interfaces defined for transmission and receipt of the stored knowledge/information. From time to time some of the information/knowledge would be transferred from people to artifacts (e.g., when a sales person leaves a company he/she has to file a report detailing clients and other soft information on the clients) for long term retention and/or planned forgetting (Landry 1999). Information/knowledge may also be transferred to an artifact to by people to reduce the cognitive and information overload and/or to have a more permanent storage. When information/knowledge transfers to an artifact some decontextualization of the knowledge/information will occur, and on retrieval of information/knowledge from the artifact some recontextualization and interpretation would need to be done (Ackerman and Christine 1999). Due to limitations of current representational techniques, transfer process, and the cognitive limitations of the people this transfer process cannot be perfect and some loss of context will inevitably occur in the exchange of information/knowledge. This will lead to differences in interpretations from the same artifacts when different people make interpretations at the same time or by the same person over a period of time. The extent of decontextualization (a real problem in building shared interpretations) may vary depending upon the factors like richness of the media (Daft and Lengel 1986) and cognitive attributes of people entities involved. The requirement of transfer of information over the links without decontextualization, retrieval and acquisition without the need to recontextualize, and subsequent interpretation is a significant problem (for reasons not explicated here due to length constraints). Most of these issues are related to the representation of the information/knowledge within the artifacts.

While the issues of representation are being dealt, the inherent complexity of capturing the context in the representation poses constraints for an OM implementation using current IT. This mandates that an implementation may focus attention on other components of the OM that can be implemented efficiently using current technologies. The bottom two layers of link

---

4Issues related to acquisition, retrieval, storage and management of OM would also be significant and would be related to the representation to some extent.
infrastructure and directory services are two such components. Link infrastructure may be implemented using the current networking technologies. The technologies are mature and this link infrastructure is sufficiently robust and capable. The technologies for implementing directories have matured over the last two years and these technologies may be used to implement an indexing/directory service for OM. We next discuss some of the features and technical requirements of such an enterprise directory to implement directory services for OM.

**Directory Service Features for Organizational Memory**

Several directory services like Novell Directory Services, Microsoft Active Directory, Netscape Iplanet Directory and others by IBM are in the offerings today. These directory services are primarily aimed at managing network and infrastructure resources and have the capabilities to manage people, but they also have the capabilities to implement directory services for OM. The OM literature does not discuss the features, requirements, and/or capabilities of a directory service in any detail. We attempt to bridge the gap with discussion of the technical features and requirements. The features or requirements are divided into the broad categories of repository, directory interfaces, performance, security, and support for business/organization. The requirements under each category should be viewed interdependent with requirements in other categories. The following set of requirements borrows from the capabilities of current directory services available in the market as well as additional requirements the authors believe to be necessary.

**Repository:** The repository should be able to define and create entries for different types of entities. These would include hardware/network resources (printers, servers, etc.), and artifacts (documents, applications etc.) and user profiles for people on these resources. The repository should meet the following requirements:

a) **Metadata** for the entries should be well defined, standardized and comprehensive with the capability for future expansion.

b) Capability to **query and retrieve**, create, modify, delete and update an object and its attributes.

c) Object allowed to have a **unique name** within the scope in which the directory is referenced.

d) **Capable and flexible search** function.

e) **Data integrity** insured.

f) **Segmentation** capability (similar to partitioning capabilities in databases) to provide the capability to split directory data into separate portions for ease of directory management and enhanced performance.

**Interfaces:** The directory will provide services to the applications and users. This requires that the interfaces for data exchange and programming should be well defined. Some of the desirable characteristics for interfaces may be:

a) Open standards for **schema and metadata** that are uniform across various directory implementations.

b) A well-defined **application programming interface** (API) that allows for the extension of the applications and functionality provided with the directory service.

c) Standard and well-defined interfaces for the interchange of information like export and import of information using standard interchange formats (e.g. XML).

d) Facilities for **discovery**. People and artifacts should be able to discover new entities. Entities should able to announce updates and modifications to other relevant directories or entities.

e) As artifacts are revised/updated/created/deleted the changes should propagate automatically. Mechanisms for **synchronization** (like event based synchronization) should exist where the repository is distributed.

**Performance:** Performance will cover several aspects like scalability, Quality of Service etc.

a) Ability to **scale** up and provide real time performance as the directory becomes the cornerstone of any access on the network and grows in complexity and scope.

b) Resilience and fault tolerance to hardware/network failures.

c) A capability to define several types of indices apart from the primary **indices** (to assure real time performance). The provision should exist to build indexes in real time.

d) **Quality of Service (QoS)** provision may be provided for entities and agents. Activities initiated by agents may be granted special consideration of bandwidth and processing time as per the according to task-specific requirements.

---

7 An important class of artifacts is documents. An architecture for on-line text with its own directory may work in conjunction with the directory services proposed here (Pankaj 2000).

8 The standard schema, metadata and APIs would mean that client applications based on these could access variety of directories.
Special consideration may be required to overcome the resource constraints posed by technologies. E.g., only the desktop in the executive offices may be able to engage in real time video communication (can be assured through machine name, IP address etc.).

e) **Flexible and extensible schema and metadata** to allow for future expansion and changes while maintaining backward compatibility.

f) A mechanism of **caching** or linking to objects in other directories that are frequently accessed by users and artifacts in a directory. E.g., if a server in the head office is frequently accessed in the regional offices the information on this server may be cached in each regional office directory.

g) **Archiving and retiring** of information to move old and less accessed information to different segment of the directory. This segment may be off-line or on a slow server and hence free up resources for more recent information. Archiving and retiring may be used to implement forgetting of information in a limited fashion. A well-managed archiving function will aid to capture the history of the organization. E.g., deleted profiles and changed profiles may be retained. A person may assume a role and then give it up, if the information on the role is important it may then be advisable to retain the information about the role in case someone in the future needs some information related to the tasks performed by that role.

**Security:** Security is an important requirement given that the directory can be global in scope. Security confirms the identity of entities and agents and controls the subsequent actions that agents can perform. Security features can be used to enforce organizational norms and processes for communication.

a) **Authentication** of the agents prior to access to the repository. Various authentication mechanisms may be used based on the scope of access that is available to the agent. The agent or an artifact may be granted pre-established levels of access to repository and other agents through roles.

b) An agent should be able to **grant and revoke** access and other right for the entities and agents it controls and to other agents. This decentralizes the management of the directory to owners or local administrators responsible for a portion or a segment of the directory.

c) **Encryption** should be available for storing critical information like passwords, etc. For exchange of information/knowledge PKI, SSL, Kerberos or digital certificates may be used.

d) Since the directory is a central repository for all identities, it should have the capabilities for **Key Management** and act as a **Certificate Authority** for the users and artifacts.

e) **Audit trails** should be established to record any change to entities and agents.

**Support for Business/Organization:** Business/organizational structure and environment would require certain capabilities from the directory services.

a) Capability to create and maintain **virtual directories** to support virtual organizations. This would mean a directory that is composed of the segments from other directories.

b) Availability of a set of methods/facilities to enforce organizational norms and processes in the communication. This would include mechanisms like the flow control of messages and the automatic filtering of knowledge/information by applications (also part of security) among others.

c) The directory structure may mimic the organizational structure. This may mean capabilities to define associations between the agents and entities in a flexible manner. Linkages may be structured hierarchically in the form of a tree as a default option, or in other fashions, to mimic an organic or a matrix structure.

d) Ability for an entity to assume multiple identities/addresses (aliases) in a particular domain.

**Current Technologies for Directory Service**

As mentioned above, several directory services are currently available and primarily aimed at managing IT resources and people. The extension to artifacts may be more an issue of scalability and performance in view of large number of artifacts that may exist in the organization. In such a case the artifacts may be built on a separate directory. Using the currently available resource directory products, a directory scheme for the organization memory can be implemented. This will provide the effective implementation of the indexing layer. Using a component wise strategy the bottom two components may be may be implemented and other components may be added as and when the technology evolves. On account of well-abstracted components the implementation of the existing components may also be improved without adversely affecting other components once new technology is available.
Conclusion

Considering OM as a distributed computing application and defining well-abstracted components can provide an implementation approach that is feasible and practical. It can use the best of the technologies available. With respect to the directory services, more and more organizations have been implementing directory services for the management of resources. It would be a simple extension for these organizations to extend the directory service and the network-computing infrastructure to provide a foundation for OM. Further work in this area is needed that will address a detailed specification of the technical requirements. A detail akin to that in a software requirement specification document will also be needed. This should be followed by a prototype implementation using directory services. It is expected that the approach outlined in this paper will provide a way forward in OM implementation.

References


Soderquist, Klas and Rajesh Nellore, “Information systems in fast cycle development: Identifying user needs in integrated automotive component development”, R & D Management; Oxford; Jul 2000;


