Memory in the Small: An Application to Provide Task-Based Organizational Memory for a Scientific Community

Mark S. Ackerman
Department of Information and Computer Science
and Graduate School of Management
University of California, Irvine

Eric Mandel
Harvard-Smithsonian Center for Astrophysics

Abstract

Many forms of organizational memory must exist embedded within the organizational processes and tasks. This paper argues that "memory-in-the small," memory utilized in the performance of an organizational task, can serve as an effective performance support mechanism. By basing organizational memory upon organizational tasks (and basing task support upon organizational memory), organizational memory systems can provide additional and necessary support services for organizations and communities.

As an example of memory-in-the-small, this paper describes a software application, called the ASSIST, that combines organizational memory with task performance for a scientific community. The ASSIST utilizes and stores the collective memory of astrophysicists about data analysis, and is used world-wide by astrophysicists. The paper also considers the theoretical and architectural issues involved when combining organizational memory with task performance.

1. Introduction

Organizational memory must exist at many locations within the organization. It must exist not only within archival forms (such as history and storage records), but as a necessary ingredient embedded within many organizational processes and tasks. Indeed, because of the short-term needs of organizations, task performance support may be the most effective use of organizational memory. In short, this paper argues that "memory-in-the small," memory utilized in the performance of an organizational task, can serve as an effective performance support mechanism. Basing organizational memory upon organizational tasks (and basing task support upon organizational memory) has many notable characteristics, including more immediate utility to organizational participants and more relevant classification of information within the memory.

This paper begins by explaining "memory-in-the-small." It continues by describing a software application, called the ASSIST, that combines organizational memory with task performance for a scientific community.[1] The ASSIST utilizes and stores the collective memory of astrophysicists about data analysis, and is used world-wide by astrophysicists. The paper concludes with a consideration of the architectural issues involved when combining organizational memory with task performance.

2. Memory in the small

Organizational memory systems come in a number of flavors and shapes. They can range from task-based memory systems (e.g., [12], [16], [4]) to organizational memory systems that attempt to capture a substantial portion of a firm's or a scientific community's knowledge (e.g., [15], [13]).

This paper argues that task-based memories will be highly important to organizations and may likely be the most fruitful place for researchers to examine. We argue this for two major reasons. First, as we have argued elsewhere, sufficient context to reconstruct any necessary meaning is required with organizational memory systems [2]. Indexing (or otherwise classifying) information pertinent to extraordinarily complex, open-ended, or

Authors' addresses: ackerman@ics.uci.edu and eric@head-cfa.harvard.edu. A copy of this paper can be obtained at http://www.ics.uci.edu/CORPS/ackerman.html

[1] We use "organizational memory" and "community memory" interchangeably in this paper. There are important differences between organizations and scientific communities. However, the tasks described here are general to both, and the term "organizational memory" is more widely used.

wicked problems will be difficult or impossible. Retrieval becomes similarly difficult.

However, the decomposition of inordinately complex problems and situations into their subcomponent tasks is a much more feasible candidate for organizational memory. It ameliorates one of the major obstacles to organizational memory, namely, the technical *and* social complexities of adequately storing and then retrieving information [11].

Second, the classic March and Simon [10] model of organizations argues that organizations have limited resources and attention. Thus, organizational memory may be most useful when it is tied to the short-term needs of organizations. Again, long-term storage and general organizational memories may not have the immediate returns required by the short-term needs of an organization.

This task-based memory support is *memory-in-the-small*. It is, in some sense, less exciting than general attempts to collect the organizational memory of an organization. In a large-scale organization, a one or two percent improvement in overall effectiveness or efficiency in using the total intellectual resources easily justifies an organizational memory effort. The waste of intellectual resources is readily apparent in most large-scale organizations. (See, for example, [17] for a popular-media exposition of this theme.)

Task-based memory, however, may need a substantially greater level of performance improvement; ironically, in light of the above arguments, it may have the greater chance of success. Task-based memory is more immediately valuable and understandable to the user in his everyday behavior ([18], [11]).

Below is a description of a specific organizational memory application, the ASSIST. The application is used by the astrophysics community, a highly geographically distributed scientific community. The ASSIST is of particular interest because it was constructed as an example of memory-in-the-small and because its task support was conceived in terms of organizational memory. In its conception of organizational memory, computational resources (as well as many familiar categories of information) are reconceptualized as memory components. The basis of the ASSIST in both organizational tasks (or in this case, the scientific tasks involved in data analysis) and in organizational memory argues for this reconceptualization, as will be discussed below.

3. Doing science: Data analysis in astrophysics

Data analysis is central to the scientific mission, and therefore serves as an interesting case of a computer-supported task. It is one readily identified by astrophysicists as central to their vocational performance [3].

For the astrophysics community, computing systems are indispensable to the data analysis task. The data available to astrophysicists comes from different wavelength regimes (e.g., radio, optical, x-ray, and gamma ray), and each might require a different analysis strategy (e.g., use of Poisson statistics for sparse x-ray data versus the use of Gaussian statistics for optical data). Astronomical data, especially those taken by space-based detectors, are of increasingly rich content. The increased spatial, spectral, and timing resolution, however, comes often at the cost of increasingly complicated instrument-dependent signatures that must be removed before the data can be used. Also, the amount of primary science and required auxiliary data can add up to several megabytes for a single observation, requiring special computer techniques to analyze it efficiently. As a result of these and other factors, no astrophysicist can analyze present-day data without the use of sophisticated software.

While the typical astrophysicist does not want to be particularly concerned with his software, the institutional and scientific context of astrophysics require him to have a keen interest in his computing systems. Because of the idiosyncrasies and complexities of astrophysics data, analysis software is increasingly developed by institutions involved with the development, launch, and operation of a space-based observation platform. Groups such as Smithsonian Astrophysical Observatory (SAO), the Space Telescope Science Institute, the National Optical Astronomy Observatories, the Goddard Space Flight Center, the European Southern Observatory, and others provide astronomical analysis software for their different astronomy missions and projects.

The software developed at these different institutions often has been incompatible at the conceptual, detailed, and data format levels (although this is beginning to change). For example, one x-ray spectral analysis program assumes that the telescope point response function already has been folded into the input spectrum, while another performs this folding as its first step. Thus, seemingly similar but scientifically very different spectra are needed as input to each program.

The institutional context, combined with the task requirements, leads to difficulties for the average user. As mentioned, a number of data analysis packages are available, and no one system can satisfy all of a user's analysis needs. It is inevitably necessary to combine the functions of several analysis packages in order to carry out a research task. For example, it might be necessary to use an image display program (e.g., SAOimage) in conjunction with a specific data analysis routine (e.g., IRAF's QPSPEC routine) to generate a spectrum that can then be translated into valid input for spectral fitting routines (e.g., XANADU's XSPEC routines). Results from these spectral fitting routines might then be plotted using another package or further analyzed using a separate statistical package. An astrophysicist might also use standard Unix tools, such as awk, grep, or an editor, to search or edit the data and intermediate results.

The necessity of using several heterogeneous software systems in the act of analysis forces additional and, to the astrophysicist, extraneous overhead. All of the activity occurs within the X Window System environment, but it all occurs using different programs with differing graphical and command-line interfaces. Moreover, just obtaining help for the different systems requires learning about a variety of facilities -- man pages, hardcopy documents, and special help facilities for each analysis program. An astrophysicist must learn about each help facility individually in order to be able to find out about using the packages.

These problems intensify for astrophysicists that combine different analysis programs as part of their overall research strategy; in doing so, the programs' distinctions become blurred in a confusing way. To go back to the previous example, consider the problem of using the SAOimage image display program and the IRAF QPSPEC routine to create a file that will be used in the XSPEC spectral analysis program. One must somehow make each of these different systems work together. How does an astrophysicist know whether the output files from one are compatible with the input files of another, or whether special commands are needed to transfer data and information from one program to another? Furthermore, if the XSPEC program reports a problem with its input file, how does the astrophysical user know whether the problem is centered in SAOimage, IRAF, or XSPEC? This type of computing overhead is extraneous to most astrophysicists, who are interested in obtaining their scientific results rather than becoming computer professionals.

One solution is for astronomy researchers to share their understanding of the different software systems (and the interconnections between these systems) with one another. Intuitively, members of a scientific community should be able to share their expertise in one or more areas of software analysis. Some, for example, will have developed techniques for combining analysis tools in different systems. Since no one can master the complexities of every program needed to perform astronomical research, there is a great need to be able to get help from fellow users and experts concerning a given software system.

However, the use of computing systems within scientific communities is now modeled upon the use of workstations and distributed computational resources. The physical organization within research institutions is no longer one in which researchers sit in a central computer room with many terminals. Most researchers now have a workstation (or even more than one) in their office, connected to other workstations by networks. While this change has made individual access to computational power much more convenient, it has taken also away a critical means of storing and disseminating organizational knowledge; i.e., shared conversation between researchers working in the same room. In fact, the experts for each system are a different set of people who often are situated at different institutions.

The situation that confronts astronomical researchers, then, can be summarized in this way: Astrophysicists, attempting to perform astronomical analysis, are confronted with a large number of software tools. Their confusing and conflicting user interfaces, data formats, and the like prevent researchers from being able to master them individually or to use them in combination. Their documentation is scattered in different help systems. And most importantly, the organizational knowledge about how to use these tools has not been effectively collected and shared. As a result, many researchers view their powerful computer systems as being unnecessarily complex and opaque, and feel frustrated in their efforts to do research without spending all of their time learning about computers.

4. The ASSIST

4.1. Overview

The ASSIST application is a prototype solution, developed over the past three years by the authors as a step toward overcoming these difficult problems. The ASSIST is designed around two central themes:

❏ Data, software, and other information used by astronomical researchers should be organized as a part of a growing organizational memory system.

❏ There should be a uniform, flexible, and extensible interface for collecting and accessing this organizational memory.

The ASSIST implements just such a uniform graphical interface to analysis modules, documentation, and other organizational memory. The ASSIST can communicate with different analysis systems and with different documentation schemes, tying both into a common orga

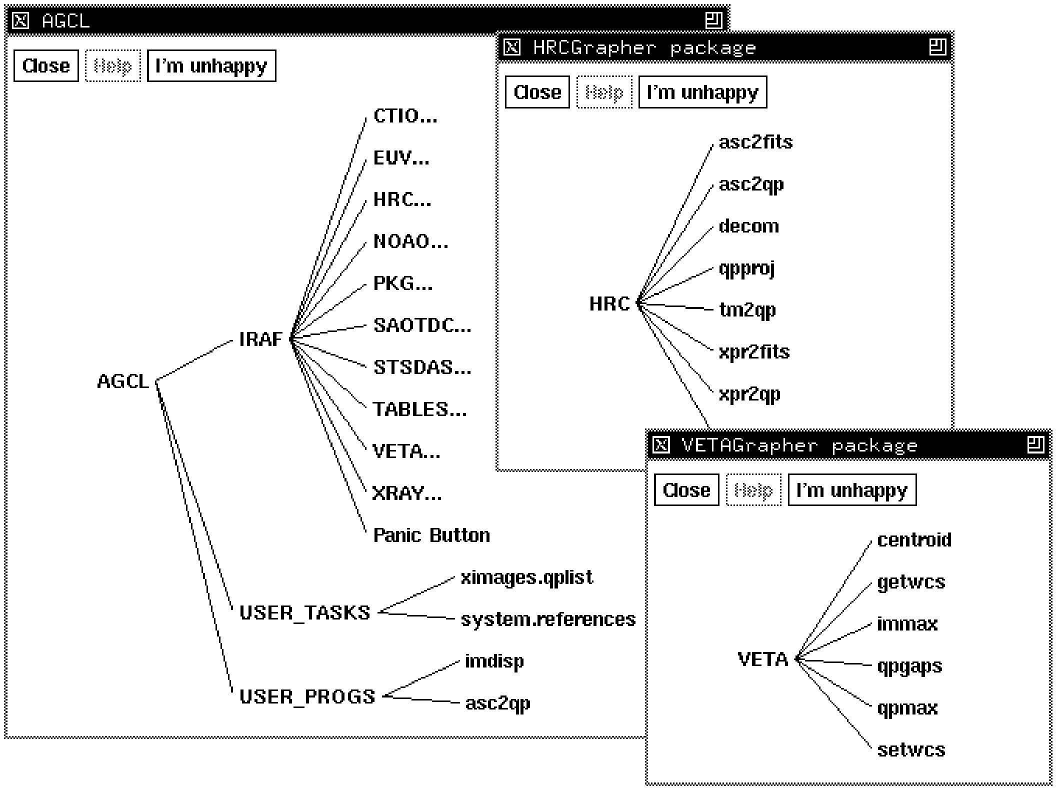

Figure 1: Interface within the ASSIST for using data analysis modules

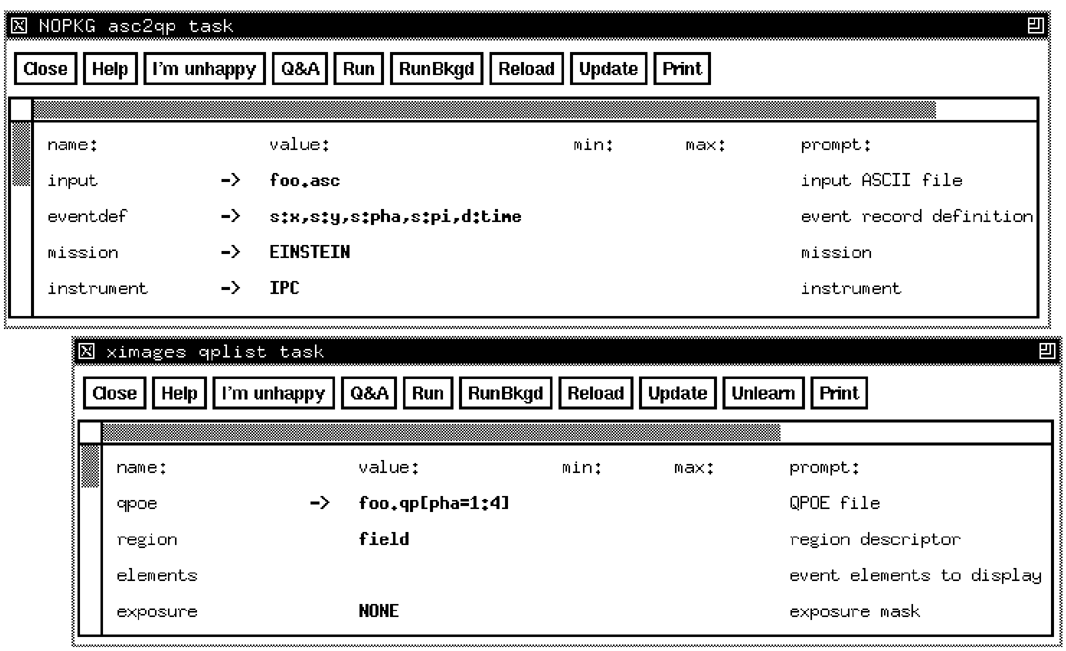

Figure 2: ASSIST's parameter editors for controlling software modules

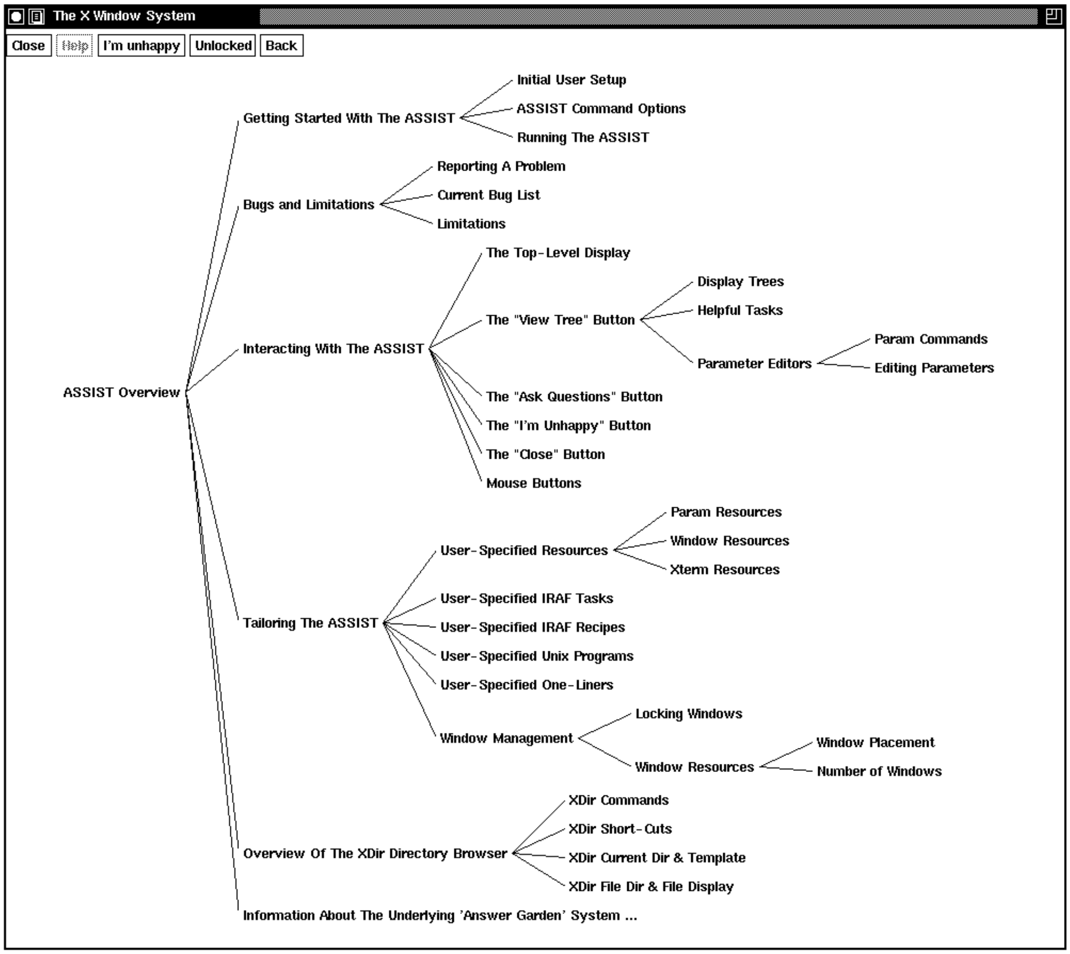

Figure 3: Part of the documentation in the ASSIST

nizational memory artifact. The ASSIST was layered on top of the Answer Garden Substrate (AGS). AGS itself is a system specially designed to facilitate the collection and dissemination of organizational memory [1]; several other applications have been built using AGS.[2] Designing both the ASSIST and AGS has been a joint venture between the research groups at SAO and UCI.

4.2 The details

The ASSIST is an X Window System application; it provides a unifying graphical, multi-dimensional view of the analysis environment, with separate windows for viewing lists of available routines and programs, for browsing help files and tutorials, for setting up and running routines, and for inspecting analysis results. In essence, the ASSIST embeds the data analysis software components within a surrounding help and documentation system. Conceptually, this is parallel to Knuth's WEB system, a system in which Pascal code for an application (e.g., TEX) was embedded within the structured documentation about the application code [8].

The ASSIST was retrieved by more than forty astronomical sites within the first four months of its public release and is in current use by astrophysicists around the world. The ASSIST has also been used on several NASA projects, such as the Compton Gamma Ray Observatory and the Advanced X-ray Astrophysics Facility [9]. The AGS system can be found on most X Window System archive servers; hundreds of sites have built and examined it.

[2] As with any system that serves as a base layer for multiple applications or that can generate multiple applications, care must be taken to distinguish among the applications of that system and the system itself. AGS can generate simple organizational memory applications, and more complex applications can be easily composed using AGS.

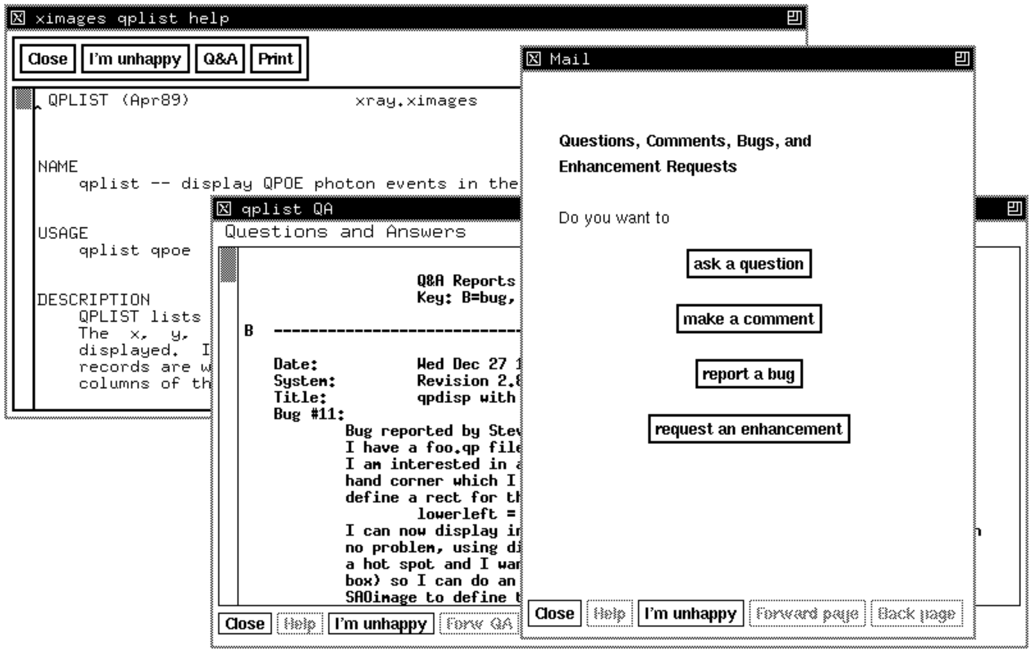

Figure 4: Various types of help available in the ASSIST

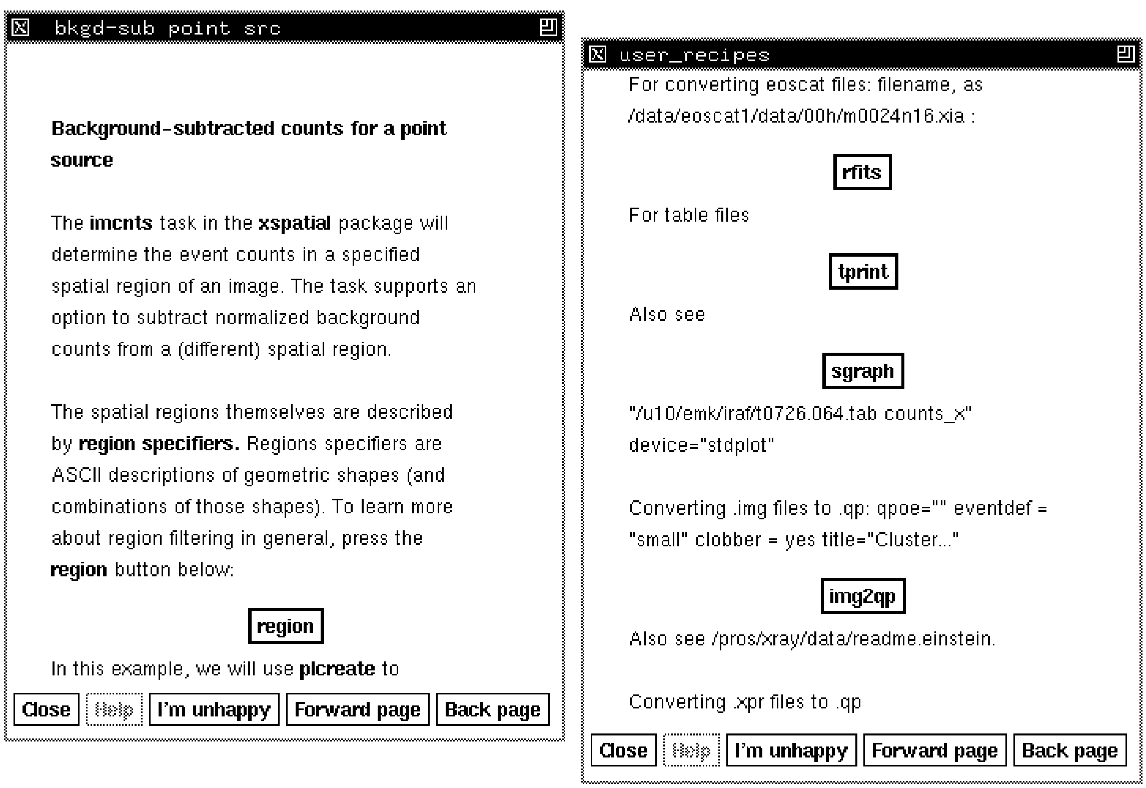

Figure 5: User recipes and tutorials in the ASSIST

Parts of the ASSIST are shown in Figures 1 through 5. Figure 1 shows the "core" of the ASSIST, the interface for locating and using about one thousand software components and their associated documentation. As one might expect, the sheer number of possible components (each a separate analysis or related routine) makes the scientific task more complex. The tree structures shown include some usual routines that an astrophysicist might use. Figure 2 shows the parameter editors for particular routines; these parameter editors are the graphical representation of the system routines. The window for inspecting analysis results is not shown. Note that astrophysicists, before the ASSIST, had to keep the names and parameters for these routines in their heads or use the documentation for each system. In an important sense, the ASSIST serves as a community artifact through which astrophysicists can forget a level of complexity unnecessary to their task; thus, the ASSIST serves not only as a memory application but as an anti-memory (or amnesiac) system [14].

To the ASSIST user, information is seamless. There is relatively little difference between retrieving the components necessary for accomplishing a scientific task and the information required to understand how to use those components. As mentioned, the information objects within the ASSIST consist of consists of data analysis packages (at the subroutine level), parameter editors, and testing software. But, the information objects can also include collections of questions and answers, documentation, help, and other standard organization memory components. Some of these organizational memory components are built dynamically from other databases and systems; the data need not be statically composed. Furthermore, this information within the ASSIST grows over time, as users have questions or need additional information. Figure 3 shows the interface to part of a document collection about these software modules. Figure 4 shows a documentation page, questions and answers about the software or documentation, and the interface to bug reports, feature requests, and help from other users and experts.

One of the most interesting memory component is what astrophysicists have termed a "living cookbook." Scientists can provide their data analysis recipes -- previously shared by pulling out their spiral notebooks -- for others or for their own personal use through the ASSIST application. Figure 5 shows an example of a user recipe as well as a tutorial. Unlike paper recipes, the links between the recipes and software modules are live; pressing a button in the recipe results in a program action. Thus, using ASSIST to follow the recipes allows the scientists to ignore the details of the data analysis software and to concentrate on the actual analysis: Having the software components and the recipes in the same memory application provides much more power to the user. Sharing the recipes allows a common method of acquiring and retrieving data analysis methods, something the community did not have before the ASSIST.

5. Supporting task-based memory

5.1. The ASSIST as organizational memory

The task-based characteristics of the ASSIST required a reconceptualization of an organizational memory system away from a pre-defined set of functionalities. In memory-in-the-small, organizational memory is whatever aids the performance of the organizational task. The artifacts included as part of this organization memory may vary considerably.

This use of organizational memory components as resources in an organizational task follows the conceptualization of shared physical artifacts in distributed cognition theory ([5], [6], [7]). Distributed cognition theory is intrinsically interested, among other concerns, in shared physical artifacts that serve as a form of social memory. Hutchins and Klausen's speed bug, a physical object that is used by cockpit crews to determine lift-off velocity, serves as an example. Speed bugs are little indicators that are placed at the point on the airspeed indicator where the pilot should lift the plane off the ground; the crew sets the value before take-off. As such, speed bugs are both memory and task performance aids. As well, speed bugs allow the crew to shift the cognitive effort required to fly across time and, because they are shared artifacts across social space.

Within the ASSIST, the organizational task is data analysis. The basis of the application as an organizational memory requires the inclusion of any information or other artifacts that will aid this task of data analysis. We have seen in the ASSIST that these memory artifacts must include research data, data analysis recipes, tutorials, the software routines that can operate on that data, documentation, system help, access to human help, and so forth.

5.2. Organizational memory support for the ASSIST

Because the ASSIST application required a reconceptualization of organizational memory, the ASSIST also required a number of changes to the system layer providing the organizational memory components. As mentioned above, the ASSIST is an application layered upon AGS; AGS provides a range of organizational memory support.

Originally, the AGS was conceived of supporting a small range of memory components, largely varying display views of text and image objects. The ASSIST, with its substantially more elaborate view of organizational memory support, required additional capabilities. It is beyond the scope of this paper to discuss AGS fully (see [1], but the required capabilities can be abstracted to several general characteristics. In order to build the ASSIST as required, AGS needed to include:

1. *An extensive heterogeneity of information types*. A given organizational task may require many different types of organizational memory for completion; the data analysis tasks supported by the ASSIST exhibit this behavior. Therefore, the set of information types that manipulate, process, or display information must be heterogeneous, as discussed in the above sections, to support software routines, documentation, system help, access to human help, and so forth. Additionally, users should not need to distinguish between electronic surrogates of paper documents, dynamic or active documents, software components, and agents.

2. *Extensibility and expandability of information types and general services*. The open-ended nature of scientific tasks requires that any task-support application be extensible and expandable. For example, the requirement within the ASSIST to add new types of software routines (e.g., from a new analysis program), as well as new types of documentation (e.g., using a new word-processing language), required AGS to provide facilities to add new information types to an application seamlessly. This included the ability for an application writer to add functionality to all information types, including those that already existed. As well, AGS provides mechanisms for extending the base set of functionalities (including the functionality to extend the system!).

The above two capabilities allowed the ASSIST to become arbitrarily complex in the information and functionalities it incorporated. Of course, the goal was *not* to have application become more complex for the end-user: the goal of the ASSIST was to reduce the amount of apparent complexity to the astrophysicist. However, we found that, almost paradoxically, the inclusion of many types of heterogeneous information types and services reduced the complexity for the user when those functionalities were combined with task-oriented capabilities. That is, to perform their task, users need many types of information and services, but they need them organized with regard to the performance of their task (rather than with regard to the information and services per se). Therefore, additional capabilities required in AGS were:

3. *Categorization by task-specific features instead by information characteristics*. Astrophysicists, when they begin a data analysis task, retrieve a number of memory components without being aware of their type or even their existence. For example, documentation and help in the ASSIST are drawn from many differing locations for a given data analysis task. Additionally, the data is automatically reworked into the correct format. The astrophysicist sees only her data analysis task, not the particulars of information retrieval and display.

   In general terms, the capabilities for heterogeneous information types and heterogeneous retrieval allow the user to access whatever information is necessary to a particular organizational task or subtask. But, to do so effectively, the required information must be categorized by the task and *not* merely by the characteristics of the information itself. Thus, AGS and the ASSIST provide for a contextualized retrieval.

4. *Heterogeneous methods of memory access and information retrieval*. In many situations, especially when solving complex problems, users may not know how to frame their task thoroughly. Moreover, users may need information about other tasks to frame or solve their problem effectively. Allowing multiple methods of access and retrieval is also required if retrieval is to be dependent on the characteristics of a task where each user performs the task differently.

   The ASSIST attempts to relieve these types of situations by including mechanisms to help astrophysicists find the right data analysis module; this required the addition of information retrieval capabilities within the ASSIST. The information retrieval capabilities, as well as hypertext links, allow astrophysicists to traverse a number of paths to find the same modules. For example, users may find the right information or modules by following reference-oriented hypertext views, tutorials, or full-text searches. The emphasis is, again, on the access and retrieval being task-based.

5. *Merging of information retrieval and communications*. Associated with the requirement that information access should be multi-faceted, we did not want to restrict information seeking to only computerized information. Scientists often seek help informally from colleagues and systems staff. To allow astrophysicists to access help from fellow users or from experts, the ASSIST required access to the social network of the community. The tie to the communications system of a community allows people to be information repositories, allowing information seekers to ask colleagues about problems and issues. This is a basic facility of AGS.

The above three characteristics attempted to ameliorate the complexity of the problem space for the end-user. We were largely successful, but more work will be required.

6. Lessons

As mentioned, the ASSIST is used world-wide by astrophysicists; privacy concerns limit us in knowing exactly how many users we have. In follow-up interviews with self-identified ASSIST users, we found that the users appreciated the system removing or lessening problems with incompatible and inconsistent data formats, interfaces, storage mechanisms, and manipulation techniques. In addition, users noted that the system provided them with all required information, data, and software in one place. Additionally, ASSIST users noted that the system simplified the inherent, cognitive complexity of finding hundreds of data analysis components and methods. While we could not reduce the inherent complexity in the scientific task, all of these comments suggest that the ASSIST noticeably reduced the amount of artificially induced complexity in astrophysical data analysis.

However, users also pointed out that further work is still required. We are grappling with three issues that users have described. First, while removing many or most of the existing artificial complexities, we introduced new complexities for the user using the ASSIST. The user interface of the ASSIST, while relatively simple, does put up large numbers of windows on the screen; some users had difficulties with the number of windows. This was particularly acute with the help nodes. Users had difficulty distinguishing among the various types of nodes; having many different visual types of help nodes made the situation more difficult to comprehend.

Second, while individual researchers did develop personal "recipes" as part of a "living cookbook," the groups responsible for the scientific analysis software were hesitant to supply such recipes, even informally. Institutional notions of publishing required levels of correctness and authoritativeness that were difficult to obtain. (This not only involved issues of organizational "face" [2], it also revolved around scientists' definitions of "publishing" versus "authoring".) More investigation is needed in order to elucidate how different kinds of information sharing can be organized and presented within this community.

Finally, our methods of allowing users to ask other scientists for help were not flexible enough. Currently, we allow only two categories of user, novice and expert. We would like to allow a range of expertise, moving from novice through normal users to expert; questions could be asked of any group.

A second version of the ASSIST effort has been funded by NASA, and work is continuing.

7. Conclusions

If the ASSIST is an example of memory-in-the-small, what does it say about organizational memory systems in general? Our experience with the ASSIST demonstrates that basing memory support around tasks and that basing task performance around memory support is effective. While not replacing the need for large-scale institutional memories, task-based memory should provide an important and useful supplement to existing organizational memory capabilities.

Furthermore, a task-based organizational memory, such as the ASSIST, argues for a reconceptualization of memory capabilities and components. In a large-scale memory system (such as that provided by [15], the tasks possible through the system are limited by the information types and capabilities given. In memory-in-the-small, however, organizational memory is whatever aids the performance of the organizational task. We believe that through task-based memory systems, like the ASSIST and AGS, communities and organizations can effectively merge their performance and their memory.

8. Acknowledgments

This research is supported, in part, by research grants from NASA (NRA-93-OSSA-09) and the UCI Committee on Research. This work was also supported under NASA contracts to the IRAF Technical Working Group (NAGW-1921), the AXAF High Resolution Camera (NAS8-38248), and the AXAF Science Center (NAS8-39073). Part of this work was done while the first author was at the MIT Center for Coordination Science under research grants from the X Consortium, Digital Equipment Corporation, the National Science Foundation (IRI-8903034), and the MIT International Financial Services Research Center.

The authors would like to thank John Roll, Steve Murray, and Roger Brissenden for their support and assistance. Conversations with Cynthia Hunt, Leysia Palen, Brian Starr, Lorne Olfman, Kent Sandoe, Cathy Marshall, and Frank Shipman, as well as the comments from the anonymous reviewers, contributed to the theoretical portions of this paper.

9. References

1. Ackerman, M. S. Answer Garden: A Tool for Growing Organizational Memory. Massachusetts Institute of Technology, Ph.D. Thesis, 1993.

2. Ackerman, M. S. Definitional and Contextual Issues in Organizational and Group Memories. *Proceedings of Twenty-seventh IEEE Hawaii International Conference of System Sciences (HICSS 94),* 1994, 191-200.

3. Bowers, R. L. *Astrophysics.* Jones and Bartlett, Boston, 1984.

4. Conklin, J. Corporate Memory. *Proceedings of Groupware '92,* 1992, 131-137.

5. Hutchins, E. The Technology of Team Navigation. *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work.* Galegher, J. and R. Kraut ed. Lawrence Erlbaum, Hillsdale, NJ, 1990.

6. Hutchins, E. How a Cockpit Remembers Its Speeds. Manuscript, 1991.

7. Hutchins, E. and T. Klausen. Distributed Cognition in an Airline Cockpit. Manuscript, 1990.

8. Knuth, D. E. *TEX: The Program.* Addison Wesley, Reading, Massachusetts, 1986.

9. Mandel, E., J. Roll, S. S. Murray and M. S. Ackerman. AXAF User Interfaces for Heterogeneous Analysis Environments. *Proceedings of Conference on Astronomy from Large Databases - II,* 1992, 361-372.

10. March, J. G. and H. A. Simon. *Organizations.* Wiley, New York, 1958.

11. Marshall, C. C., I. Frank M. Shipman and R. J. McCall. Putting Digital Libraries to Work: Issues from Experience with Community Memories. *Proceedings of Digital Libraries '94,* 1994, 126-133.

12. Morrison, J. Team Memory: Information Management for Business Teams. *Proceedings of 26th Hawaii International Conference on System Sciences,* 1993.

13. Orlikowski, W. J. Learning from Notes: Organizational Issues in Groupware Implementation. *Proceedings of Computer Supported Cooperative Work (CSCW) '92,* 1992, 362-369.

14. Sandoe, K. and L. Olfman. Anticipating the Mnemonic Shift: Organizational Remembering and Forgetting in 2001. *Proceedings of International Conference on Information Systems (ICIS),* 1992, 127-137.

15. Schatz, B. R. Building an Electronic Community System. *Readings in Groupware and Computer-Supported Cooperative Work.* R. Baecker ed. Morgan-Kaufmann, San Mateo, CA, 1993.

16. Schwabe, G. Providing for Organizational Memory in Computer-Supported Meetings. *Proceedings of 27th Annual Hawaii International Conference on System Sciences,* 1994, 171-180.

17. Stewart, T. A. *BrainPower.* Fortune. 44-60, 1991.

18. Weiser, M. and J. Morrison. Capturing, Linking and Retrieving Team Project Information. Working paper, Department of Management Sciences, University of Iowa, 1994.