Memory in the Small: Combining Collective Memory and Task Support for a Scientific Community

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

Eric Mandel
Harvard-Smithsonian Center for Astrophysics

short title	Memory in the Small
pages	39 (including 8 figures on separate pages)
tables	none
contact	Mark Ackerman Computer Science 444 University of California, Irvine Irvine, CA 92697 714/824-7355 ackerman@uci.edu

Memory in the Small: Combining Collective Memory and Task Support for a Scientific Community

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

Eric Mandel
Harvard-Smithsonian Center for Astrophysics

short title Memory in the Small

contact Mark Ackerman
Computer Science 444
University of California, Irvine
Irvine, CA 92697

714/824-7355
ackerman@uci.edu

Abstract

Many forms of memory exist embedded within the processes and tasks of an organization or community. “Memory-in-the small”, or memory utilized in the performance of an institutionally important task, serves as an effective task support mechanism. By basing memory upon tasks (and basing task support upon memory), 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 system, called the ASSIST, that combines 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 architectural and theoretical issues involved when combining memory with task performance.

Keywords: organizational memory, collective memory, scientific community, computer-supported cooperative work, scientific applications, information spaces, task support

1. Introduction

Organizational and community memories exist in many different forms. Many organizational memory systems are large-scale efforts supporting an entire organization or community (e.g., the Lotus Notes systems described in Orlikowski [1]). Yet, smaller endeavors might also be fruitful. For example, imagine the following support system for a relatively simple task:

> You sit down to write a research paper. Available to you are not only your notes and data, but a myriad of components for tables, spelling, grammar, formatting, and editing from many software houses. You can easily use the best component for any particular subtask, since the components interoperate easily. Furthermore, these components are enveloped in a memory system, providing reference materials, publication templates, bibliographic references, best practices, documentation for the software, help in developing your paper, and many other aids to the task.

Such a system would certainly make writing easier and more efficient. Scenarios such as this call out for combining memory support with task-support components. As with paper writing, tasks are embedded in webs of social actors, activities, and history. Performing a task as a professional requires knowledge of what is possible, what is effective, and indeed, what others' best performances have been. If a support system could augment a major task of an organization or community, it could be of substantial benefit, since it would support only activities known to be central to that organization or community.

This paper reports on such a system. The ASSIST supports the world-wide astrophysics community in a major activity, data analysis.[1] It combines memory support with task performance

---

[1]We recognize that there are important differences between organizations and scientific communities. For example, scientific communities are more likely to share information [2], and

support. Based on both a set of underlying theoretical assumptions and some practical concerns (described below), we believed that the ASSIST would prove beneficial to astrophysicists. The system was indeed useful and usable for its community; however, we found that not all forms of memory support were institutionally viable.

In addition to describing the ASSIST, a further goal in this paper is to deepen the understanding of organizational and community memories. By examining the issues in specific systems (e.g., the ASSIST), we hope to begin the construction of Merton-like theories of the middle-range about how such systems interact with the communities and organizations in which they operate. These theories will not be macro-level theories of organizational performance, nor micro-level studies of individuals' information seeking behaviors. They will be, instead, design theories that explain aggregated behavior and guide further research (as has been suggested by Olson and Olson in [4] and [5] as well as Walls, Widmeyer, and El Sawy in [6]). This line of investigation is not new in organizational and collective memory research. Morrison and Weiser [7] began this work for team memory systems, and Schwabe [8] began to develop a theory of the middle for group memory. Schwabe, for example, suggested that storing group memory in smaller chunks than documents, capturing memory with little effort, and paying attention to privacy measures are all important features of group memory systems. This paper, however, makes the goal of such a theory explicit; it also views constructing systems and deploying them into everyday use as one avenue for constructing these theories.

Accordingly, this paper begins by explaining the rationale for combining memory and task support. To distinguish this form of memory from larger-scale or larger-scope trials, we call our form memory-in-the-small; this will be described in detail below. We describe why memory-in-the-small is theoretically

---

organizations can coerce their members into adopting software (as in [3]). However, the issues described here are general, and we will be careful to distinguish when organizational memory is different from community memory.

interesting, and then we present practical reasons why the astrophysics community needed this type of memory support. Next, the paper describes the ASSIST, the system that combines organizational memory with task performance for the astrophysics community. After a description of the architectural and theoretical issues involved when combining organizational memory with task performance, the paper concludes with lessons learned from the use of the system by astrophysicists.

2. Memory in the small

Collective memory systems come in a number of flavors and shapes.[2] One way to distinguish among such systems is according to their scale (group, organization, community, society) and scope (from tightly focused to general). By doing so, we can construct a design space for collective memory systems.

Representative examples of small-scale, task-based memory systems include Team Memory [9], GO [8], GroupWord [10], and CM/1 [11]. Morrison's Team Memory system examines the task issues in supporting groups, Schwabe's GO examines the general issues of group support systems, Mandviwalla and colleagues' GroupWord examines issues in collaborative writing as a group memory, and Conklin's CM/1 system augments design rationale activities for a group or small organization. All of these systems attempt to augment the memory for a relatively small social grouping.

Larger-scale, larger-scope systems (Figure 1) include Lotus Notes implementations (e.g., as studied in [1]), which try to incorporate the intellectual assets of an entire company, and Schatz's Worm Community System (WCS) [12], which attempts to store all knowledge for a scientific community. WCS is particularly interesting because it incorporates a number of different memory types, including

[2] We will use the term *collective memory* to denote the common attributes of organizational, institutional, and community memory. The term has a related, but slightly different meaning in the historiographical and critical literatures, but unfortunately there is no better term to denote memory in a range of collectivities.

documents, scientific data, and even chat sessions. Figure 1 also shows that WCS supports a larger range of memory objects than does Notes.

Figure 1 about here.

In this research, we were interested in the potential utility of other points in this design space for collective memory systems. As noted, we wished to consider a system that would combine a task-orientation and collective memory facilities to augment individual performance of a specific task within a larger community or organizational context. This echoes the call in Morrison and Vogel's [13] analysis of group support based on a task orientation. Such a system would be tightly focused (i.e., small scope) because of its emphasis on a particular task, yet would support critical activities occurring within a large collectivity (i.e., large scale). As such it would support a range of tasks in the same manner that DSS supported decision-making ([14], [15], [16], [17]). (Memory-in-the-small systems lack DSS's emphasis on modelling, replacing it with an emphasis on memory.) Within a memory-in-the-small system, the collectivity's memory could help individuals to learn common tasks, and at the same time, further learning could accrue to the collectivity as well as the individual. Memory systems that combine both task and memory orientations, we call *memory-in-the-small* to emphasize their restricted scope.

We expected an examination of memory-in-the-small would be fruitful for two theoretical reasons. First, technical problems exist for large-scale organizational memories, resulting from the need for sufficient context to reconstruct any necessary meaning [18]. Indexing (or otherwise classifying) information pertinent to extraordinarily complex or open-ended problems is likely to be difficult, perhaps even impossible. Retrieval becomes similarly difficult. However, tying memory to specific tasks would allow the decomposition of these complex problems and situations into their requisite tasks, and this could be a more feasible candidate for computational support. Tying memory to tasks, then, had the distinct possibility of ameliorating one of the major obstacles to organizational memory, namely, adequately storing and then retrieving information [19].

Second, the classic March and Simon [20] model argues that organizations have limited resources and attention. (Although they speak of organizations, their conclusions carry to larger collectivities as well.) Thus, collective memory is most likely to be useful when it is tied to the short-term needs of a collectivity. Long-term and large-scope collective memories may not offer the immediate returns required by the short-term needs of an organization or community. These short-term needs revolve around tasks.

Memory-in-the-small support is, perhaps, less exciting than general attempts to collect the entire memory of an organization or community. In a large-scale organization, a one or two percent improvement in using the organization's intellectual resources easily justifies an effort to augment its memory, because the waste of intellectual resources is readily apparent. (See, for example, [21] for a popular-media exposition of this theme.) Yet, one might need substantial improvement in task performance (well above one or two percent) to provide a sufficient return on investment or effort for task-based memory systems to be viable. Nonetheless, task-based memories could have the greater chance of success, since they may be more immediately valuable and understandable to users in their everyday behavior ([19], [22]).

Figure 2 about here.

Because of these theoretical concerns, we believed that there could be substantial benefit to examining memory issues in terms of specific tasks. The ASSIST was constructed as an example of memory-in-the-small; its task support was conceived in terms of collective memory. Figure 2 shows the ASSIST's placement in terms of the theoretical space; it is a small-scope, large-scale memory system with many types of information objects. The ASSIST includes many familiar categories of information, such as documents and tutorials. As well, active computational resources, such as software components and dynamic help, are reconceptualized as memory components.

In addition to these theoretical reasons, we also had practical reasons to believe that basing the ASSIST in both a critical task and in collective memory would be useful. We chose the task of data analysis, after investigating the scientific tasks of astrophysicists as well as the software previously constructed to support those tasks. We therefore next describe both the centrality of data analysis for astrophysics and the complexity of the previous generation of software. The combination led to astrophysicists' spending too much time learning software and not enough on their data analysis, a situation that cried out for additional support.

3. Doing science: The centrality of data analysis in astrophysics

Data analysis is central in astrophysics ([23], [24], [25]). One astrophysicist described his work to us:

> An astrophysicist tries to explain how things work. We have instruments. I look at a particular location and I see something unusual, I don't know what it is, but I try to explain the meaning in the context of some particular model. In analysis, you extract particular objects from the data. I want to illustrate the characteristics of whatever I've found -- for example, the x-ray spectrum from a source. I want to confirm it's a supernova or whatever I'm interested in. A simple analysis might be, for example, to fit different models to the spectrum, to show it fits a certain theoretical form.

Astrophysicists readily identify the data analysis task as crucial to their "doing science". (See [26] or [27] for a textbook explanation of astrophysicists' work.) Said one astrophysicist:

> Data analysis is the central reason for existence. Observing has, to my mind, taken second place.

Yet for many astrophysicists, the data analysis task has become increasingly complex and a source of frustration. Part of this has been due to the huge expansion of available data, the result of numerous space-based observatory missions. While additional data is to be celebrated, this profusion of data carries a substantial increase in complexity, both in the data themselves and in the data analysis programs necessary to use the data. This is often a source of frustration for astrophysicists:

> I think for most of us, analysis is a pleasure. However, it becomes frustrating when things don't work or systems are too complex.

Data analysis, then, was a task ready for support. To understand the specifics of providing support, it is important to understand how difficult the astrophysicists' analysis task was before the ASSIST. First, the data available to astrophysicists were themselves complex. The data come from different wavelength "regimes" (e.g., radio, optical, x-ray, and gamma ray), where each regime might require a different analysis strategy (e.g., one might use Poisson statistics for sparse x-ray data, but Gaussian statistics for optical data). Moreover, the astronomical data, especially those taken by space-based detectors, provide increasingly rich and complex content. For example, the increased spatial, spectral, and timing resolution often comes with increasingly complicated instrument-dependent signatures that must be removed before the data can be used.

Additional difficulty resulted from the complexity of the astrophysicist's software environment. No astrophysicist can analyze present-day data without the use of sophisticated software, software that has grown increasingly complex and difficult to master and use. Because of the idiosyncrasies and complexities of astrophysics data, analysis software has increasingly been developed by institutions involved with the development, launch, and operation of space-based observation platforms. Therefore, separate institutions, such as the Smithsonian Astrophysical Observatory (SAO), the Space Telescope Science Institute, the National Optical Astronomy Observatories, the Goddard Space Flight Center, and the European Southern Observatory, 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 has begun to change). For example, one analysis program might assume that the response function already has been folded into the input spectrum, while another might perform this folding as its first step. Thus, seemingly similar but scientifically very different data are needed as input to each institution's programs.

The institutional context, combined with the complexity of the data and the inherent difficulty of doing scientific analysis, led to quite a difficult situation for the average astrophysicist. As mentioned, a large number of data analysis packages are available, and it became 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., the SAOimage program from the SAO) in conjunction with a specific data analysis routine (e.g., IRAF's QPSPEC routine) to generate intermediate data 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.

An astrophysicist must somehow make each of these different systems work together. This is a daunting requirement for successfully conducting analysis. The typical astrophysicist does not know -- or care -- whether the output files from one are compatible with the input files of another, and whether special commands are needed to transfer data and information from one program to another. Furthermore, to return to the earlier example, if the XSPEC program reports a problem with its input file, the astrophysical user will probably not know where the error actually occurred, since the data may have passed through several different analysis programs. This type of computing overhead is extraneous for most astrophysicists; they are far more interested in performing their scientific work than knowing the data processing details.

To make their task worse, this difficult situation is compounded by the difficulty of obtaining help. While all of the activity occurs within an X Window System environment, it is carried out using different programs with differing graphical and command-line interfaces. Just obtaining help for the different systems requires learning about a variety of facilities, including the Unix "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. Moreover, astrophysicists often cannot ask one another; astrophysics is a highly distributed scientific community. There has been little ability to share information about the analysis programs.

The situation, then, that confronted astronomical researchers at the beginning of our research was well summarized by an astrophysicist:

> The scientific aim drives the technological means. You fit a model to a given data spectrum, and you have to do this with computers. But then you find out the software is complex -- there are many UIs [user interfaces] to similar pieces of software, plus no software environment is complete. You have to switch midstream to complete a task. It pushes you off. You don't do what you should do.

In summary, an astrophysicist, attempting to perform astronomical analysis before the ASSIST, faced a seemingly huge number of confusing and conflicting data formats and software packages. Numerous separate analysis packages, with over 1000 analysis components, are available. Their confusing and conflicting user interfaces, data formats, and the like prevented scientists from being able to master them or to easily use them. Their documentation was scattered in different help systems, and the collective knowledge about how to use these tools had not been effectively collected and shared. As a result, many astrophysicists viewed their powerful computer systems as being unnecessarily complex and opaque, and felt frustrated in their efforts to do their research work.

This situation called for a novel support mechanism for the data analysis task. This need resulted in the design of the ASSIST system.

4. The ASSIST

The ASSIST has evolved over the past five years to provide support for astrophysicists in their analysis task. The ASSIST is designed around two central assumptions:

- ❏ Data, software, and other information used by astronomical researchers for data analysis should be organized as a part of an evolving memory system. New software, new documentation, and new folklore should be part of that collective memory. A critical design assumption was that having all of the necessary memory present in one environment would facilitate task performance.

- Furthermore, to be useful to astrophysicists (as relatively naive users who are, in the words of one astrophysicist, “in the tail of being computer literate”), there should be a uniform, flexible, and extensible interface for collecting and accessing this collective memory. Different kinds of memory -- software, documentation, informal knowledge, and other types of information -- should be available through the same interface.

The ASSIST implements just such a uniform graphical interface to analysis modules, documentation, and other types of collective memory. The ASSIST can communicate with different analysis systems and with different documentation mechanisms, tying many sources into a common collective memory artifact.

At the surface, the ASSIST provides a unifying graphical, multi-dimensional view of an astrophysicist’s task environment. 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 for the application code [28].

The ASSIST system 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. It has also been used on several NASA projects, such as the Compton Gamma Ray Observatory and the Advanced X-ray Astrophysics Facility [29].

The ASSIST is built upon the Answer Garden Substrate (AGS), a system specially designed to facilitate the collection and dissemination of organizational memory [30]. In addition to AGS’ standard components, the ASSIST adds many domain-specific components and functionalities that are about equal in size to the AGS. Designing both the ASSIST and AGS has been a joint venture between the research groups at SAO and the University of California, Irvine.

4.1. A tour of the ASSIST

Figure 3 about here.

The ASSIST is an X Window System application 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. The following section describes the ASSIST system from the user perspective.

Figure 4 about here.

Parts of the ASSIST are shown in Figures 3 through 7. Figure 3 shows the first “nodes” that a user sees. The various ASSIST components are arranged into a hyper-network, but the network is presented to the user as a series of hierarchical “trees” that organize and display the various types of information, software, and data. Figure 4 shows the “core” of the ASSIST, the interface for locating and using the approximately 1000 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 figure shows some routines that an astrophysicist might commonly use (upper left) and recently used routines (upper right). Figure 4 (center) also shows a parameter editor for a specific routine, in this example, the “imstatistics” routine. In general, these parameter editors are the graphical representation of the system routines. Before the ASSIST, astrophysicists 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 allows the users to forget low-level details just as any high-level memory should [31]. Thus, the ASSIST serves as a community artifact removing a level of complexity unnecessary to their task. The window for inspecting the analysis results is also shown in the lower right. Although not shown here, data files can also be represented as ASSIST nodes.

Figure 5 about here.

An important point is that information is seamless to the ASSIST user. The system was designed to tie together the components necessary for accomplishing a scientific task and the information required to understand how to use those components. Retrieval is the same for either; users do not need to distinguish among them. Figures 4 through 7 show these various components. As shown in Figure 4, the information objects within the ASSIST consist 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 memory components. Some of these 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 5 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.

Figure 6 shows the interface to part of a document collection about the SAO Spectrum X Gamma (SXG) modules. Note that these nodes are HTML pages. The ability to include HTML pages provides authors with the capability of simultaneously reaching both the general public and ASSIST users (i.e., astrophysicists). The ASSIST provides an override capability so users can view an ASSIST node if one is present, and an HTML page if one is not. This allows the general public to see an overview document, while also allowing astrophysicists to access the software or internal documentation.

Figure 6 about here.

One of the most interesting memory components 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 7 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 the ASSIST to follow the recipes allows the scientists to ignore the details of the data analysis software and to concentrate on the actual analysis, since the software components and the recipes are now in the same environment. Sharing the recipes allows a common method of acquiring and retrieving data analysis methods, something the community did not have before the ASSIST.

Figure 7 about here.

The ASSIST, then, collects all of various components necessary to the astrophysicist's analysis task. As described, this is a very narrowly focused community memory for astrophysicists: it centers around a particular activity instead of many possible activities.

4.2. Usefulness and usability

A critical question was whether the targeted community would find the ASSIST useful and usable. If they did not, one might suspect the basic design assumptions. Astrophysicists found the ASSIST useful and, for the most part, usable.[3]

---

[3]Although the ASSIST is used world-wide by astrophysicists, privacy rules precluded us from knowing exactly what users did with the system. The data for this section, therefore, is from a small study of self-identified astrophysicists. We formally interviewed six self-identified ASSIST users about the system and questioned over twenty others. The interviews were semi-structured and were obtained between six and twelve months after the ASSIST was first made available. The formal interviews were all done within a two-week period, and were a half-hour to an hour long. Questions centered on the usability of the ASSIST as well as the usefulness of the ASSIST for the data analysis task.

Self-identified users are generally more positive than randomly selected users. In general, we have tried to be cautious in our interpretation of the interview data.

In interviews, we found that the users appreciated the system removing or lessening problems with incompatible and inconsistent data formats, interfaces, storage mechanisms, and manipulation techniques. One user reported:

> We have sets of tools that are not particularly friendly, so I use the ASSIST instead.

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, especially for beginning astrophysicists:

> ...the ASSIST is used a lot by people just getting into the field.

There were many astrophysicists who did not adopt the ASSIST. We found that the ASSIST 's usefulness was less for established astrophysicists, because they tended to use a fixed set of data analysis components that they knew well. Some astrophysicists noted that this use was largely a matter of habit. Those who were learning the data analysis packages or who used them infrequently found the ASSIST more valuable.

In general, the usability of the ASSIST was acceptable. (A system need not be "user-friendly" to be useful, although usability is important to continued discretionary use.) Since the ASSIST was an X Window System application, astrophysicists were already used to the general user interface. The functionality of the system was designed to be largely transparent to the users, and it was. In all of the interviews, there was little complaint about basic usability.

There were, however, two usability problems that must be addressed in future research. The first was the number of windows. The ASSIST could generate a large number of windows, and astrophysical users found the number confusing:

> Problems? Lots of windows. I can never find anything. How do you make them go away?

Users tended to become confused between iconifying windows and deleting windows. We attempted to ameliorate this problem through window "stacks" (as with deck-of-card document interfaces), which could be set to automatically delete windows as they sunk in the stack. The application could also be tailored to put different types of information in different stacks or even to tile only the most recent windows. Unfortunately, many users found all of these configurations confusing as well and for some, this hampered their use of the system. This problem does not appear to be limited to the ASSIST, however. The number and positioning of windows has also been found to be a problem in the UARC collaboratory [32] which can also generate large numbers of windows. Olson and Finholt [33] have argued that such systems will need to find new types of data interfaces in order to reduce window complexity.

The second usability problem, which was more minor, was that the lists of analysis components were too terse for some users. Since the lists were composed dynamically from the underlying analysis packages (in order to dynamically add or remove components or even entire analysis packages), the ASSIST used the names of software routines rather than more helpful descriptions. The effect of this usability problem, however, tended to be small, since users could request pop-up help on the package names.

Nonetheless, users found the ASSIST to be usable in general. Most of its functionality was transparent to the users, which became its major value as was intended in the design. Users could find everything they needed in one place in a transparent manner, and they were able to concentrate on their scientific tasks. While the ASSIST could not reduce the inherent complexity of the scientific task, the interviews suggest that the ASSIST noticeably reduced the amount of artificially induced complexity in astrophysical data analysis.

This section provided an overview of the ASSIST's functionality and showed that the system worked as intended for the astrophysicists. The system functioned transparently in the manner intended in the design. The next section discusses what issues the ASSIST raised architecturally and socially for memory-in-the-small and general memory system design.

5. Findings: Supporting task-based memory

As discussed in the introduction, our goal in this research was to deepen our understanding of memory-in-the-small and collective memory in general. Olson and Olson [4], crediting Card, argue for four stages in systems-oriented research areas such as human-computer interaction. These stages fit collective memory research, if slightly amended:

1. Constructing illustrative "point systems" that serve as examples of what can be done.

2. Evaluating and comparing systems to gain an understanding of the proper dimensions by which systems and their social context vary.

3. Analyzing these dimensions to determine relationships.

4. Articulating models and laws that govern behavior.

It is worth noting Olson and Olson's subsequent observation about CSCW systems, since collective memory systems can be considered a form of CSCW systems:

> The field of CSCW has mostly been at the stage of building point systems. Many more systems have been built than have been evaluated. Some attempts at understanding the dimensions by which systems and impacts vary have been proposed.... ([4], p. 2).

It is our belief that this statement is also true for collective memory systems and especially for memory-in-the-small systems. The ASSIST is a "point" in the design space of collective memory. However, we also sought to evaluate the design goals and use of the system in order to further understand what dimensions by which to consider subsequent design and further research. This should contribute to a Merton-like theory-of-the-middle about collective memory.

In this section, therefore, we consider what implications the ASSIST application had for conceptions of organizational and community memories. There were three basic implications -- how collective memory needed to be reconceptualized to support tasks, how the system architecture had to be designed to support this reconceptualized memory, and a set of considerations that were partly technical and partly institutional. We cover each of these implications in turn.

5.1. Reconceptualizing collective memory

As we progressed in our study, the implications of the central design assumptions behind the ASSIST -- conceptualizing task support as a memory problem and providing a uniform interface to many different types of memory components -- became more and more considerable. The task-orientation of the ASSIST required a reconceptualization of a collective memory system away from a pre-defined set of information-repository functionalities.

In a memory-in-the-small, collective memory is whatever aids the performance of the task. This use of memory components as resources in a task follows the conceptualization of shared physical artifacts in distributed cognition theory ([34], [35]). Distributed cognition theory observes that people use whatever resources are at hand in the environment to accomplish their task. The artifacts included may vary considerably.

Within the ASSIST, the task is data analysis. The basis of the application as a collective memory requires the inclusion of any information or other artifact that will aid this task of data analysis. As noted above, the ASSIST includes research data, data analysis recipes, tutorials, the software routines that can operate on that data, documentation, system help, and access to human help.

5.2. Architectural issues in the ASSIST

One important research goal in the ASSIST effort was to understand any architectural issues. Since the ASSIST required the above reconceptualization, the ASSIST also required changes to the underlying collective memory support. As mentioned above, this memory support is provided by the AGS layer within the ASSIST. Originally, the AGS was conceived of as supporting a small range of memory components, consisting largely of display views of text and image objects. In this, it was similar to the current capabilities of the World Wide Web, and could get by with a simple repository model for the information store. The ASSIST, with its substantially more elaborate view of collective memory support, required additional capabilities.

It should be noted that currently AGS is built directly upon the X Window System; thus, the ASSIST is not currently based upon the Web, although it can include Web pages. The ASSIST requires functionality that cannot be supported entirely by the combination of current Web browsers, their Java engines, HTTP, and HTML. This required functionality will be detailed in this section, since we believe that this functionality will be important to memories-in-the-small regardless of the underlying system. While the ASSIST was implemented using a particular system (i.e., AGS), we believe its memory issues generalize.

In order to construct the ASSIST, the underlying system needed to include an extensive heterogeneity of information types, as previously discussed. A given task may require many different types of collective memory for completion; the data analysis tasks supported by the ASSIST exhibit this behavior.

Therefore, the ASSIST required a set of information objects that manipulate, process, and display information. These information objects are not only static documentation or data records; the ASSIST's architecture also had to include heterogeneous objects to support software routines, active documentation, system help, access to human help, and so forth. While the ASSIST needed to provide access to static documentation and other paper surrogates, this functionality is straightforward, and almost any underlying memory system could support it. However, specialized support needed to be constructed for access to software routines and human help. Furthermore, the ASSIST needed to include dynamic and active components, such as dynamic or active documents, software components, and agents, which construct their results and displays. For example, the ASSIST needed to dynamically build its documentation, system help, and bug reports from multiple sources. AGS grew to support these dynamic and active services; the Web is beginning to support these services, although specialized programming is still required.

Figure 8 about here.

Because the ASSIST incorporates many types of active, distributed services and information, the ASSIST's software architecture had to consist of a federation of services, most of which are distributed. Figure 8 shows a simplified overview of the ASSIST's software architecture . (Omitted are SAO's XPA and Xkibbitizer servers and some of the AGS services that make it possible to interact with the operating system and other applications. See [30] for more details.) Not only is each type of service extensible, the system as a whole is extensible and expandable. Although the effort to construct the ASSIST was initially more complex, such an architecture makes it much easier to subsequently include external helper services and diverse information types.[4]

5.3. Social-technical issues in the ASSIST

The major goal of the ASSIST project, as discussed in the Introduction, was to contribute to a deepening of our conception of collective memory. As we considered how task support was changed by viewing it as a collective memory problem, we began to see a number of issues emerge. Most involved considering technical issues within their context of use, a distinguishing characteristic of collective memory research ([18], [31], [36]).

In this section, accordingly, we examine some of the social-technical issues in the ASSIST. Some of these collective memory issues will exist in other systems, but they were made quite apparent (and thus open to investigation and analysis) in the ASSIST.

---

[4]A similar level of customized and integrating code would be required with Web browsers and Java applets. It would be difficult, as far as we know, to achieve this architecture in Lotus Notes.

Two issues concerned what appeared at first to be almost trivial technical considerations. A standard retrieval or task support system might have minimized or ignored them. However, these issues were really socially based. Conceptualizing the task support as a collective memory enabled us to understand the important community and institutional implications. The specific issues, and their generalizations, were:

- *Local and distributed store, as an example of recognizing institutional boundaries within collective memories.* One apparently straightforward technical consideration concerned where to store the data. This issue arose because of limitations in Web browsers and Java applets -- they cannot easily handle local store (i.e., on the user's hard disk or file partition). Indeed, many astrophysical data sets are in remote databases, so local storage would appear to be unnecessary. However, users want to store their data (especially transformed data sets) locally to maintain control over it. Moreover, specific institutions, and not the community as a whole, are responsible for reliable storage. As a collective memory system, the ASSIST needed to respect institutional boundaries, without unduly limiting memory access. Within the ASSIST, astrophysicists needed to be able to store their data files locally, but wanted to access both distributed and local data in the same ways. This social consideration has become one of the major issues in making a Web-based version of the ASSIST.

- *Different "levels" of information, as an example of needing to differentiate and support institutional roles within collective memories.* As a memory system within a scientific community and within institutional settings, the ASSIST needed to provide support for different kinds of community and institutional actors. As mentioned, the ASSIST makes it possible to overload a Web page (accessible to the public) with the interface to a software module (accessible by only community members). The ASSIST also recognizes a range of authoring privileges and support functions. For example, the ASSIST can be tailored to allow users to seek assistance from a local expert (e.g., the hallway “guru”), intermediate experts (e.g., a help desk), or external experts (e.g., the programmers of an analysis package).

The following issues were, in fact, largely technical. However, understanding how the system would be used within a context of work enabled us to reconsider the appropriate technologies. These issues were:

- *Categorization by task-specific features rather than by information characteristics.* In a memory-in-the-small, the task is more important than the details of the data formats, information display formats, and other data-centric attributes. Storage and retrieval, then, should be based upon the information’s use in the task. Our view of this issue builds on that of Morrison and Weiser ([7], [9], [22]), who argue for task-based storage and retrieval.

  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, their data is automatically reworked into the correct format. The astrophysicist sees only her or his data analysis task, not the particulars of information retrieval and display. In general terms, the ASSIST allows 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 (or subtask) and not by the characteristics of the information or data itself. Thus, the ASSIST provides for contextualized storage and retrieval.

- *Multiple methods of memory access and information retrieval.* Of course, storage and retrieval must be based upon the information’s use in the task -- for the particular user. 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 many mechanisms to help astrophysicists find the right data analysis module or other information. The addition of information retrieval capabilities within the ASSIST, as well as hypertext links, allows 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, informal notes, or full-text searches. The emphasis, again, is on the access and retrieval being task-based.

- *Providing help by merging 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 while trying to complete an analysis task. To allow astrophysicists to access help from fellow users or from experts, the ASSIST supports 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.

These technical features -- categorization by social use and practice, inclusion of many retrieval mechanisms, and considering the social network as an information source -- could only be derived by examining the social context of the astrophysical analysis task. The resulting technical features may be useful, however, to other collective memories.

The last set of issues involved substantial technical support, but was essential to the long-term function of a collective memory in the astrophysical community. There has been relatively little work on the long-term issues in organizational and community memories, and in this study, several issues were found to be critical to the ASSIST's continued use.

- *Official, semi-official, and unofficial information.* In the ASSIST's range of information objects, some were considered "official" by the astrophysics community. For example, the code routines and analysis documentation were considered official because they were "released" by an institution. Some types of information were considered "unofficial". Examples of this included personal trees of favorite or useful analysis tools (which could be swapped among users) and user recipes. These were not released officially by any given institution and were considered personal information.

  Between these two extremes were what we termed "semi-official" information. These included bug reports, some tutorials, and even some forms of software. For example, some software was not maintained or supported by any official institution; yet, the software was considered to be useful enough to include. Similarly, "official" bug reports were collected by the programmers of a software package, but the reports were not "officially" available. This semi-official information inhabited a twilight area of respectability.

  While integrating official information into the ASSIST was difficult, providing support for semi-official and unofficial information was even more arduous. The application-program interfaces were not standard, information was not formatted similarly, and records existed in multiple places. However, semi-official information provided a sense of completeness for users. Since users knew that the information was available, they wanted to have it. The secondary inclusion provided a sense that the "system was real" (in the words of one interviewee), since it was often insider information. Unofficial information provided the work-arounds and folklore that make real use possible, but was difficult to gather, disseminate, and maintain, as will be discussed next.

- *Maintenance of the information content*. For communities, this is a critical issue. Just as with software, other types of information, especially active information, will have a much longer maintenance phase than initial development phase [37]. This was very obviously true with the ASSIST. New versions of analysis packages and other types of "official" information were released on institutionally sanctioned schedules (which varied greatly) and needed to be included in the ASSIST. "Unofficial" information was never officially released, by definition, and knowing when to disseminate it was a continuing issue. (Inclusion of Web pages in the ASSIST solved the mechanics of the problem, but did not solve the need to know what the materials were and when to include them. Without a mediating institution, there is no clear way to know.)

  Semi-official information occupied the twilight area of maintenance as well as development and dissemination. In some cases (e.g., bug reports for some analysis packages), programming groups did not ensure the veracity or consistency of the report database. Even in situations where it was maintained (e.g., software routines), it was not officially released by any institution or other agency. The ASSIST rebuilt bug reports and some help documentation dynamically for each session. This may not have been optimal in terms of system performance, but it ensured the most recent information for users, an important social consideration.

To summarize, this section identified a number of issues that were important to the ASSIST as a memory-in-the-small. We have omitted several other issues that were idiosyncratic to the ASSIST (e.g., the specifics of its interactions with other astrophysical tools). The above issues, then, are hopefully a first step towards building a theory of the middle-range about how such memory mechanisms will be shaped by and will interact with their task context. In the case of the ASSIST, the social-technical considerations were paramount, as one might expect with collective memory systems.

6. Future work

The ASSIST attempted to ameliorate the complexity of the problem space for astrophysical end-users in their data analysis task. We were largely successful, but users pointed out that further work is still required. We are grappling with three issues. First, while removing many or most of the existing artificial complexities, we introduced a new complexity for the ASSIST user. As mentioned, the user interface of the ASSIST, while relatively simple, can place a large number of windows on the screen, and some users had difficulties with the number of windows. This was particularly acute with the help nodes.

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. The unofficial recipes were still treated as though they were “official” publications, and the institutional notions of publishing required levels of correctness and authoritativeness that were difficult to obtain. (This not only involved issues of organizational "face” [18], 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, the ASSIST allows 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.

7. Conclusions

If the ASSIST is an example of memory-in-the-small, what does it say about collective 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 memory-in-the-small systems cannot replace large-scale institutional memories, task-based memory should provide an important and useful supplement to existing organizational and community memory capabilities.

Furthermore, a task-based memory, such as the ASSIST, argues for a reconceptualization of memory capabilities and components. In a large-scale memory system (such as the Worm Community System [12]), the tasks possible through the system are limited by the information types and capabilities given. In a memory-in-the-small, however, collective memory is whatever aids the performance of the organizational task. We believe that through task-based memory systems, like the ASSIST, communities and organizations can effectively merge support for their performance and their memory.

This study also uncovered a number of important issues concerning the ASSIST and its context of use, the astrophysical data analysis task. By examining the technology in use and in a specific community, we hope that this paper has helped to map out theoretical issues in designing similar systems.

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, Ralph Swick, Steve Murray, and Roger Brissenden for their support and assistance. Leysia Palen, Lorne Olfman, Kent Sandoe, Christine Halverson, Cathy Marshall, and Frank Shipman, as well as the anonymous reviewers, contributed to this paper.

9. References

1. W. J. Orlikowski, "Learning from Notes: Organizational Issues in Groupware Implementation," in *Proceedings of Computer Supported Cooperative Work (CSCW'92)*, 1992, pp. 362-369.
2. T. Allen, *Managing the Flow of Technology*. Cambridge, MA: MIT Press, 1977.
3. M. L. Markus, "Power, Politics, and MIS Implementation." *Communications of the ACM*, vol. 26 no. 6, pp. 430-444, 1983.
4. G. M. Olson and J. S. Olson, "Research on Computer Supported Cooperative Work," in *Handbook of Human Computer Interaction*, M. Helander, Ed. Amsterdam: Elsever, 1997, in press.
5. G. M. Olson and J. R. Olson, "User-centered design of collaboration technology." *Journal of Organizational Computing*, vol. 1, pp. 61-83, 1991.
6. J. G. Walls, G. R. Widmeyer, and O. A. El Sawy, "Building an Information System Design Theory for Vigilant EIS." *Information Systems Research*, vol. 3, no. 1, pp. 36-60, 1992.
7. J. Morrison and M. Weiser, "A Research Framework for Empirical Studies in Organizational Memory," in *Proceedings of 29th Annual Hawaii International Conference on System Sciences*, 1996, pp. 178-187.
8. G. Schwabe, "Providing for Organizational Memory in Computer-Supported Meetings," in *Proceedings of 27th Annual Hawaii International Conference on System Sciences*, 1994, pp. 171-180.
9. J. Morrison, "Team Memory: Information Management for Business Teams," in *Proceedings of 26th Hawaii International Conference on System Sciences*, 1993, pp. 122-131.
10. M. Mandviwalla, S. D. Clark, II, and K. Sandoe, "Collaborative Writing as a Process of Formalizing Group Memory," in *Proceedings of 28th Annual Hawaii International Conference on System Sciences*, 1995, pp. 342-350.
11. J. Conklin, "Corporate Memory," in *Proceedings of Groupware '92*, 1992, pp. 131-137.
12. B. R. Schatz, "Building an Electronic Community System," in *Readings in Groupware and Computer-Supported Cooperative Work*, R. M. Baecker, Ed. San Mateo, CA: Morgan-Kaufmann, 1993, pp. 550-560.

13. J. Morrison and D. R. Vogel, "Group Support From a Task-Based Perspective," in *Support Functionality in the Office Environment: Proceedings of the IFIP TC8/WG8.3/WG8.4 Working Conference*, A. A. Verrijn-Stuart, H. G. Sol, and P. Hammersley, Ed. New York: North-Holland, 1991, pp. 3-23.

14. S. L. Alter, *Decision Support Systems: Current Practice and Continuing Challenges.* Reading, MA: Addison-Wesley, 1980.

15. M. Treacy, "Future Directions in DSS Technology," in *The Rise of Managerial Computing*, J. F. Rockart and C. V. Bullen, Ed. Homewood, IL: Dow Jones-Irvine, 1986, pp. 111-130.

16. J. Sprague, Ralph H. and H. J. Watson, Ed., *Decision Support Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

17. R. W. Blanning and D. R. King, *Current Research in Decision Support Technology*. Los Alamitos: IEEE Press, 1992.

18. M. S. Ackerman, "Definitional and Contextual Issues in Organizational and Group Memories." *Information Technology and People*, vol. 9, no. 1, pp. 10-24, 1996.

19. C. C. Marshall, F. M. Shipman, and R. J. McCall, "Putting Digital Libraries to Work: Issues from Experience with Community Memories," in *Proceedings of Digital Libraries '94*, 1994, pp. 126-133.

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

21. T. A. Stewart, "BrainPower", Fortune, pp. 44-60, June 3, 1991.

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

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

24. P. A. Charles and F. D. Steward, *Exploring the X-Ray Universe*. New York: Cambridge University Press, 1995.

25. G. W. Fraser, *X-Ray Detectors in Astronomy*. New York: Cambridge University Press, 1989.

26. A. Sanqvist and T. P. Ray, Ed., *Central Activity in Galaxies: From Observational Data to Astrophysical Diagnostics*. New York: Springer-Verlag, 1993.

27. M. A. Barstow, Ed., *White Dwarfs: Advances in Observation and Theory*. Boston: Kluwer Academic, 1992.

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

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

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

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

32. J. S. Olson and S. Teasley, "Groupware in the Wild: Lessons Learned from a Year of Virtual Collocation," in *Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW'96)*, 1996, pp. 419-427.

33. G. Olson and T. Finholt, "Upper Atmospheric Research Collaboratory." Human-Computer Interaction Consortium workshop, presentation, 1997.

34. E. Hutchins, "The Technology of Team Navigation," in *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, J. Galegher and R. Kraut, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1990, pp. 191-220.

35. E. Hutchins, *Cognition in the Wild.* Cambridge, MA: MIT Press, 1995.

36. E. W. Stein and V. Zwass, "Actualizing Organizational Memory with Information Systems." *Information Systems Research*, vol. 6, no. 2, pp. 85-117, 1995.

37. S. R. Schach, *Software Engineering.* Homewood, IL: Irwin, 1990.

replaced

Figure 1: Location of Worm Community System (WCS) and Orlikowski's Notes site in the collective memory space

replaced

Figure 2: Location of the ASSIST system in the collective memory space

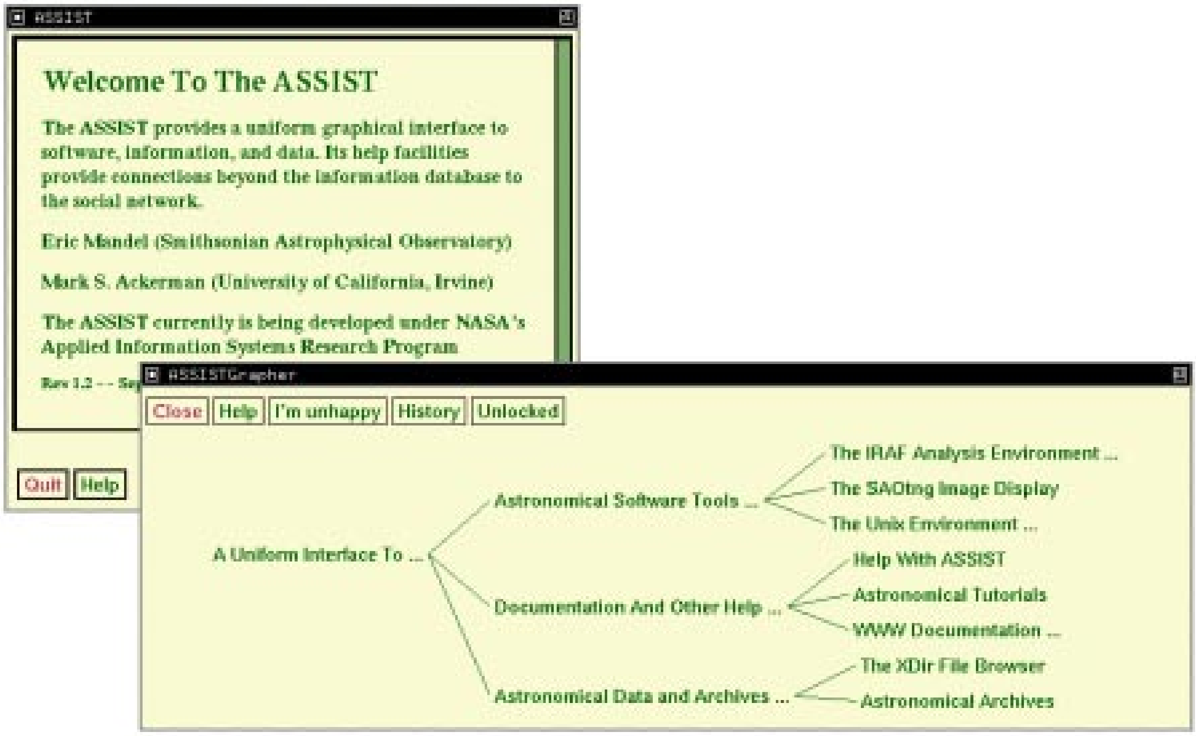

Figure 3: Opening "nodes" in the ASSIST

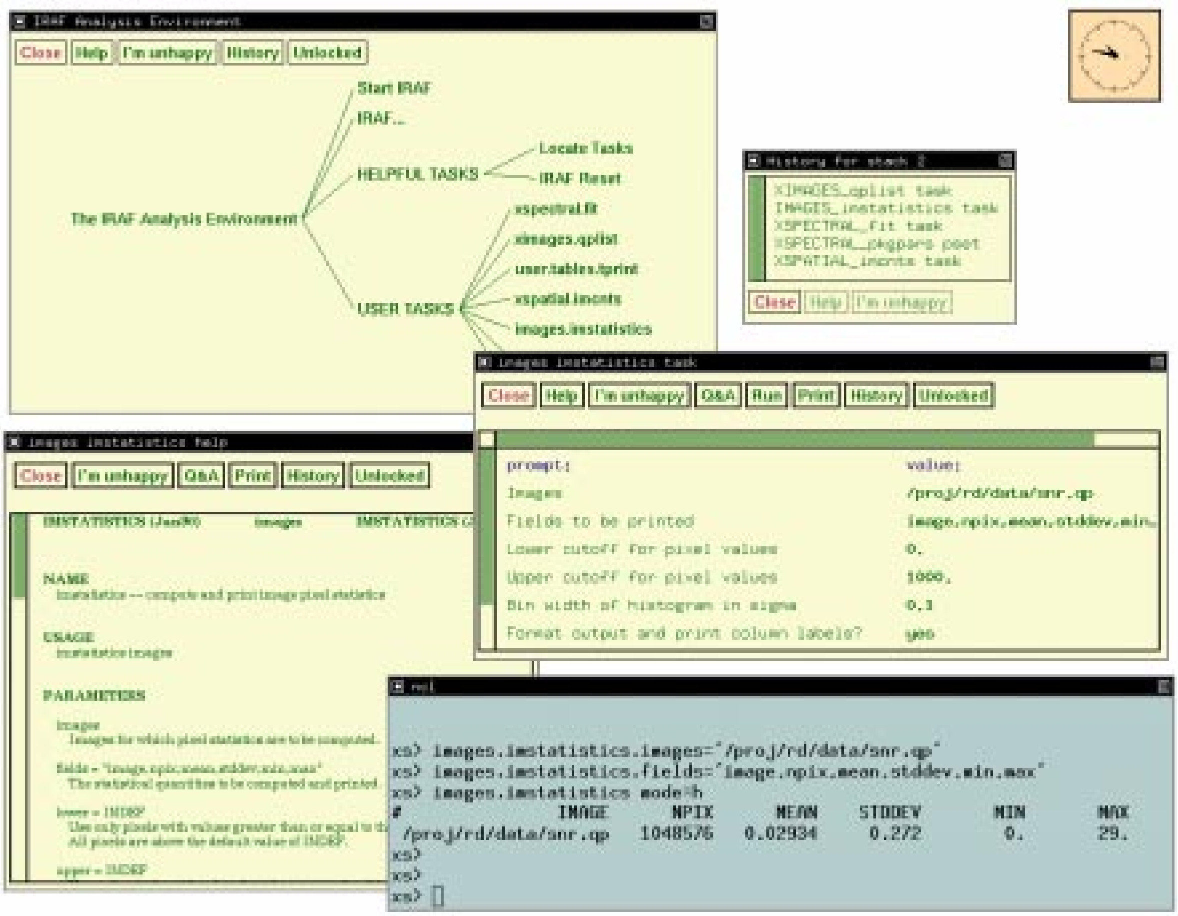

Figure 4: The ASSIST components for accessing software routines

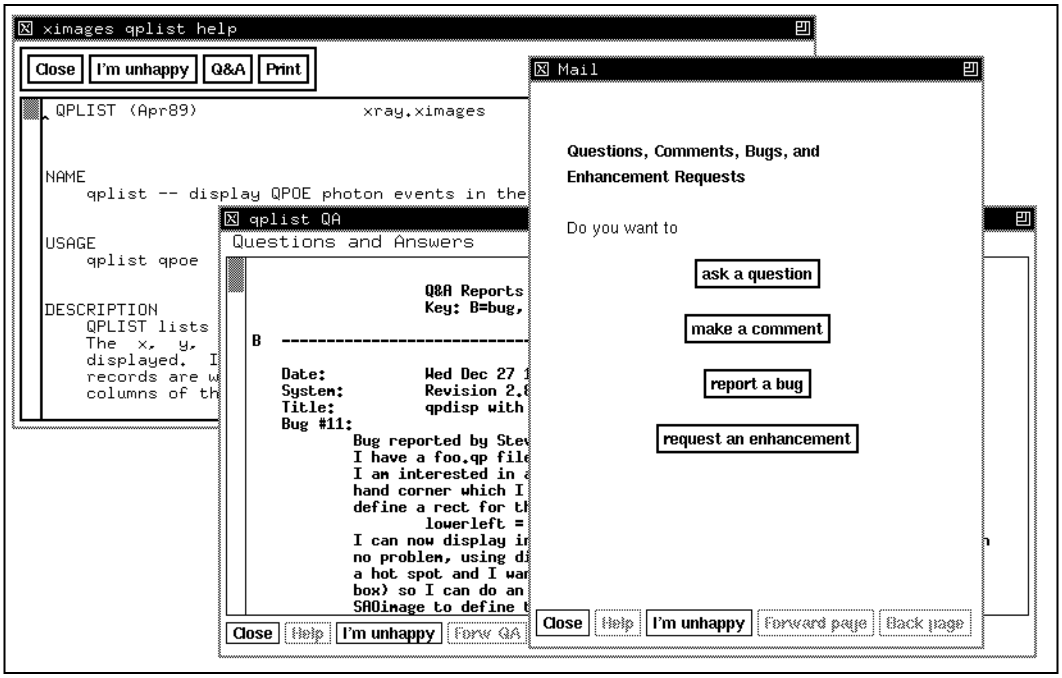

Figure 5: Various types of help available in the ASSIST

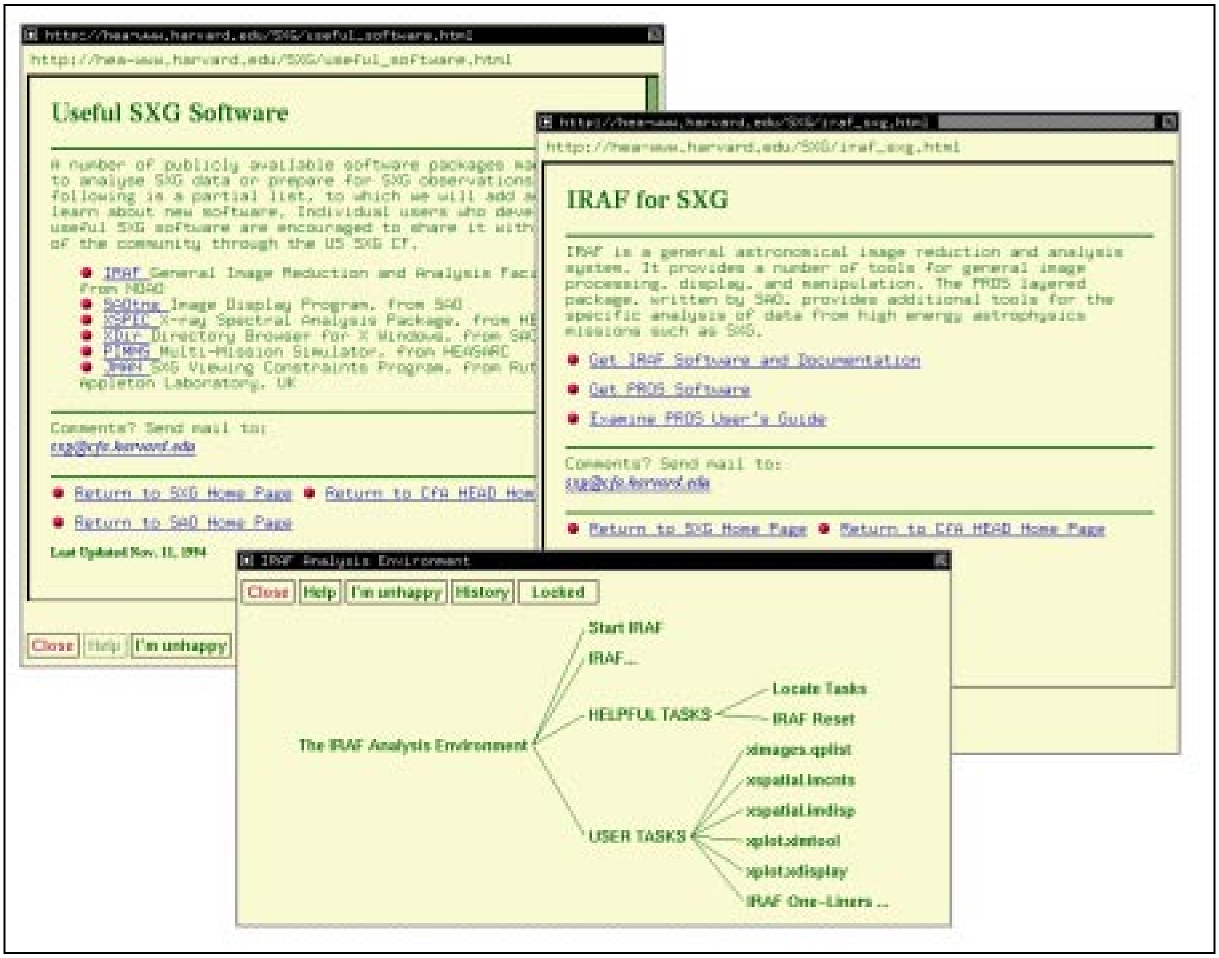

Figure 6: An example of the ASSIST's WWW support.

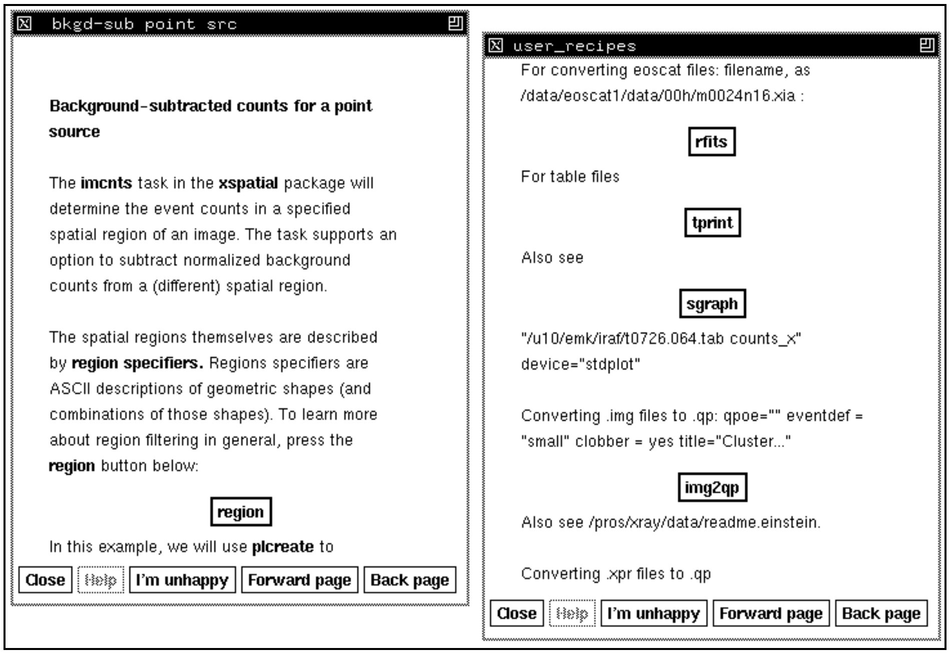

Figure 7: User recipes and tutorials in the ASSIST

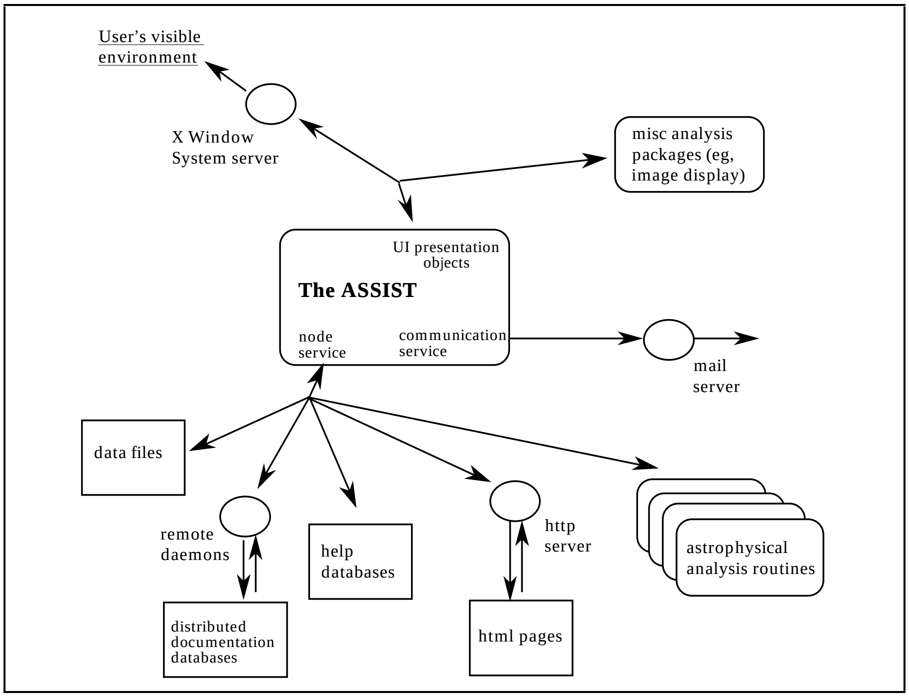

Figure 8: ASSIST's distributed architecture