THE ASSIST: BRINGING INFORMATION AND SOFTWARE TOGETHER FOR SCIENTISTS

NASA Grant NAGW-3913

Final Report

For the period 01 April 1994 through 30 September1997

Principal Investigator
Eric Mandel

October 1997

Prepared for:

National Aeronautics and Space Administration
Washington, D.C. 20546

Smithsonian Institution
Astrophysical Observatory
Cambridge, Massachusetts 02138

The Smithsonian Astrophysical Observatory
is a member of the
Harvard-Smithsonian Center for Astrophysics

The NASA Technical Officer for this grant is Glenn Mucklow, Code: ST, National Aeronautics and Space Administration, Washington, D.C. 20546

1 Introduction

The ASSIST was developed as a step toward overcoming the problems faced by researchers when trying to utilize complex and often conflicting astronomical data analysis systems. It implements a uniform graphical interface to analysis systems, documentation, data, and organizational memory. It is layered on top of the Answer Garden Substrate (AGS), a system specially designed to facilitate the collection and dissemination of organizational memory. Under the AISRP program, we further developed the ASSIST to make it even easier for researchers to overcome the difficulties of accessing software and information in a complex computer environment.

2 Enhancing ASSIST

An important goal of this project was to generalize the ASSIST so that it could support and manage a variety of astronomical analysis systems. Prior to our work under the AISRP program, the ASSIST had hard-wired supported for the Image Reduction and Analysis Facility (IRAF) and the Unix shell as target analysis environments. This support was generalized by the development of a new Analysis "information node type" within ASSIST. This node type allows arbitrary external programs (both graphical and command-driven) to be started and managed from the ASSIST. Its functionality has been tested with a variety of astronomical and other programs, such as IRAF, the Unix shell, the XSPEC spectral analysis package, the XDir directory browser, telnet, ftp, and the SAOimage and SAOtng display programs.

Advanced communication management between the ASSIST and external programs was implemented as part of our project. We developed the X Public Access (XPA) mechanism, which allows any X program (such as ASSIST) to define points of public access through which data and commands can be exchanged with external programs. XPA services were added to ASSIST to allow external processes to manipulate information nodes. External processes can send commands to ASSIST to activate information nodes such as documents and analysis systems. External processes also can add new temporary nodes to ASSIST or retrieve information about existing nodes.

The ASSIST was designed to display a variety of information types and formats. Since HTML has become a very widely used documentation format, an HTML node type also was added to ASSIST. The HTML node supports asynchronous retrieval and display of World Wide Web (WWW) documents, with full interrupt capability. This node type takes an HTML file specifier as input and displays it as hypertext. When a hypertext link is activated within a document, the string underlying the link is analyzed. If it is an ASSIST node, that node is displayed. Otherwise, if it is a Universal Resource Locator (URL), the appropriate WWW document is retrieved and a temporary HTML node is created within ASSIST to manage and display that document. Thus, the ASSIST HTML node can be used to display arbitrary ASSIST nodes or HTML documents or any combination of the two. If there is an ASSIST node of the same name as the WWW URL (or an alternate ASSIST node associated with a URL), the user can display either by using different mouse buttons. The significance of this feature is that one can have a standard HTML page that displays a hypertext document when viewed from a WWW browser such as Netscape, but that also activates other functions (such as running an analysis program). This functionality will be extremely valuable in cases where one wants to supply general information via the Web but do something more detailed for scientific users via ASSIST.

As part of ASSIST development work, we also enhanced the underlying IRAF-compatible parameter interface. This parameter interface is one of the foundations for ASSIST's communication with, and management of, external analysis systems. It consists of an application programming interface to manage named parameter files, read parameters, write parameters, and check for the existence of parameters. The interface also contains a set of host programs to support manipulation of parameters at the command line.

From the outset, the parameter interface has supported "indirect parameters", whose values are contained in other parameters. For ASSIST, indirect parameter support was extended to include "dynamic indirect parameters", *i.e.*, parameters whose values are generated by running an external program. Dynamic indirect parameters are used, for example, to retrieve the current set of region of interest markers from the SAOtng image display program.

Experience with indirect and dynamic indirect parameters helped solve an important problem with analysis environments in which separate programs have different parameter names for the same parameterized object (such as the current image file name). It never has been possible to enforce a uniform naming scheme for parameters in programs that are developed by different groups for different purposes. The result is that users continually and manually have to update parameters in different tasks to a single "global" value, a situation that leads to errors, wasted time, and much frustration. To solve this problem, we developed a layered extension to the parameter interface to support the "classing" of indirect parameters. Indirect classing is a mechanism by which arbitrary parameters can be mapped to (classes of) parameters in "global" parameter files. The global parameter files are arbitrary and can be set up by the user as needed. Indirect classing can be turned on and off using environment variables without modification of parameter file values themselves. Thus, users can switch between global parameters and local parameters quickly and easily.

To integrate indirect classing into ASSIST, we extended ASSIST to act as a mini-message bus system. ASSIST can track the parameters in individual programs that have been mapped to global parameters, and update the display of parameter values when global values are changed. It also can ensure that these parameters are only modified globally. In this way, ASSIST can provide a unified view of parameters across different analysis systems and programs.

3 Public Release of ASSIST

Beginning in the spring of 1995, substantial effort was expended to prepare for a public release of ASSIST (as part of the SAO R&D software suite). Testing, documentation, bug fixes, and enhancements were carried out. A simple software build procedure was automated and documented. We made public our first public release of ASSIST in June 1995. Periodic releases were offered throughout our grant period, right up to the end of our project in September 1997. In all, we made eight public releases of ASSIST and related software as part of the SAO R&D software suite. We offered the software on all major platforms used in the astronomical community (SunOS, Sun Solaris, SGI, Dec Alpha, HP, and Linux), and made available both source code and binary distributions. ASSIST will continue to be available through anonymous ftp (ftp://sao-ftp.harvard.edu/pub/rd) and through the Web (http://hea-www.harvard.edu/RD/). The latest release (1.7) has been retrieved by more than 300 unique sites in its first 3 months of availability (comparing favorably with 450 unique sites over a 9 month period for release 1.6).

Because of the heavy use of this software suite within the astronomical community, we recognized early on the need to divide our AISRP project work into two efforts: at SAO, we focused our attention on the support and further development of ASSIST and related software in the SAO R&D package. At UC Irvine, we focused our attention on further research into information systems management. The need to make this division of labor is a measure of the success of SAO R&D software suite: we quickly acquired a substantial user base in the astronomical community and we needed to devote considerable attention to supporting these users.

4 Future Directions for Information Management

While work on the SAO R&D software suite was being performed at SAO, future-directed and experimental work was performed at UC Irvine. This work reflected our continued research into computer-supported cooperative work (CSCW) and consisted of three components:

- the Cafe ConstructionKit (CafeCK)
- the Collaborative Refinery (Co-Refinery)
- the Do-I-Care agent (DICA)

We envision that these three systems eventually will replace parts of the Answer Garden Substrate, which forms the foundation for ASSIST. They are meant to be a next-generation platform using the Web and Java components. They also provide new levels of help and information gathering for astrophysicists.

The Cafe ConstructionKit is a computer-supported cooperative work (collaboration), communication, and information toolkit. Providing building blocks for constructing applications easily, the primary purpose of CafeCK is to construct the necessary components for collaborative help for the ASSIST. Through several cycles of prototyping, we migrated the design toward Java for our components to handle message communication (both synchronous and asynchronous), message filtering, and message retrieval, and to be compatible to standard Web browsers and applications. We completed these efforts by re-designing and re-implementing a more flexible and powerful escalation agent to provide better collaborative help for the user.

The Collaborative Refinery (Co-Refinery) is a set of Web-based tools that make it easy to construct Frequently-Asked Question (FAQ) capabilities. Using Co-Refinery, users can iteratively construct answers to commonly asked questions, allowing software experts to avoid answering the same questions over and over again. We have added to Co-Refinery's editing and authoring tools for distilling information from any type of semi-structured message by creating several new information distillate types. Also, during the last year, we tested Co-Refinery informally with user groups, made the authoring more substantial and robust, refined and simplified the component architecture, and began work on a database back-end.

The Do-I-Care Agent (DICA) deals with the resource re-discovery problem on the Web and in the ASSIST. In general, users may know that Web locations provide interesting information, but they may not know when to revisit the site for new information. This is particularly vexing for sites containing answers to help questions or for data sites. DICA allows the users to "train" the agent (using a machine learning engine) to revisit the site only when something "interesting" and new has been found. Over the last year, DICA was field tested and the system was re-designed and re-implemented to handle collaborative filtering among a group of coordinated uses.

5 Papers

The following papers concerning ASSIST were accepted during our grant work:

M. S. Ackerman and E. Mandel, "Memory in the Small: An Application to Provide Task-Based Organizational Memory for a Scientific Community", Proceedings of the 28th Annual Hawaii International Conference on System Sciences (HICSS 1995), vol. IV, pp. 323–332.

E. Mandel, M. S. Ackerman, "The ASSIST: Unified Management of Public and Private Information", Astronomical Society of the Pacific Conference Series, **101**, 92.

E. Mandel, M. S. Ackerman, "The ASSIST: A Unified Interface to Software and Information", NASA Science Information Systems Newsletter, **v. III, issue 37**, pp. 1–5.

M. S. Ackerman and B. Starr. "Social Activity Indicators for Groupware." IEEE Computer, June, 1996.

B. Starr, M. S. Ackerman, and M. Pazzani, "Do-I-Care: A Collaborative Web Agent." Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'96), short papers, April, 1996, pp. 273–274.

M. S. Ackerman and D. W. McDonald. "Answer Garden 2: Merging Organizational Memory with Collective Help." Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'96).