N9517531

NASA Conference Publication 3281

Third International Symposium on Space Mission Operations and Ground Data Systems Part 2

Edited by
James L. Rash
NASA Goddard Space Flight Center
Greenbelt, Maryland

Proceedings of a conference held at
Greenbelt Marriott Hotel
Greenbelt, Maryland, USA
November 15-18, 1994

National Aeronautics
and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

1994

REPRODUCED BY:
U.S. Department of Commerce
National Technical Information Service
Springfield, Virginia 22161
NTIS

Automating Testbed Documentation and Database Access Using World Wide Web (WWW) Tools

Charles Ames
Jet Propulsion Laboratory
California Institute of Technology
Mail stop 179-206
Pasadena, CA 91109
818 354 7098
818 393 6154 (fax)
chuck@tsunami.jpl.nasa.gov

Brent Auernheimer
California State University
Dept. of Computer Science
Fresno, CA 93740-0109
209 278 2573
209 278 4197 (fax)
brent_auernheimer@CSUFresno.edu

Young H. Lee
Jet Propulsion Laboratory
California Institute of Technology
Mail stop 301-340
Pasadena, CA 91109
818 354 1326
818 393 4100 (fax)
young@natashi.jpl.nasa.gov

Abstract

A method for providing uniform transparent access to disparate distributed information systems was demonstrated. A prototype testing interface was developed to access documentation and information using publicly available hypermedia tools. The prototype gives testers a uniform, platform-independent user interface to on-line documentation, user manuals, and mission-specific test and operations data. Mosaic was the common user interface, and HTML (Hypertext Markup Language) provided hypertext capability.

Introduction

The Jet Propulsion Laboratory's Test Engineering Laboratory (TEL) evaluates new technologies for possible use during spacecraft system testing.

Formal test environments are highly structured and information intensive. Information that may be useful for later analysis of failure reports or change requests is not always obvious during system test. Clearly, it is better to err on the side of collecting data that may never be used. Testers also consult numerous reference documents, including test plans, handbooks, acronym lists, and glossaries.

For these reasons, spacecraft system testing is a paper-intensive operation. The project described in this paper addresses this problem using freely-available, multi-platform hypertext interfaces.

Several NASA centers support related work. An inter-center working group, ICED[1] (InterCenter Electronic Documentation workgroup) is informally organized to share information among groups exploring the use of hyper- and multi-media interfaces to testing, operations, and ground data systems.

This paper is organized as follows: the context of the prototype, the JPL system test environment, is described; next, the development of the prototype is outlined; the transition from prototype to product is documented; finally, future work is described.

The JPL System Test Environment

JPL's Advanced Multi-Mission Operations System (AMMOS) is a networked computer system consisting of 28 software and hardware subsystems. Its principle purposes are to sequence and uplink commands to spacecraft and to process downlinked telemetry. Both testers and users provide feedback to AMMOS developers about needed repairs and improvements in the form of Failure Reports (FRs) and Change Requests (CRs) which are stored in the Anomaly Tracking System (ATS) database. Developers and testers refer to this database to prioritize their work.

[1]ICED has regularly scheduled teleconferences and maintains an on-line repository of findings. The contact person for ICED is Anthony Griffith, agriffith@jscprofs.nasa.gov.

Preceding page blank

Preparation for system test occurs in parallel with system development. Test preparations include: writing test plans; organizing test cases, data, and scenarios into test procedures; defining acceptance criteria; and negotiating the test schedule.

System verification and validation includes functional, performance, security, and reliability testing. Test logs are maintained, reports are generated, and FRs are written detailing software, hardware, or configuration failures. Engineers generate CRs in response to FRs. A change board approves or disapproves each CR after impact analysis.

As proof of concept, a variety of physical documents used by testers were converted to hypertext. These documents include:

- References: Test Engineering Handbook, Acronym List, and Glossary
- AMMOS User manuals and guides
- Flight project specific documents: test plans, procedures, and reports
- Articles posted to the Internet about software testing.

More than 4MB of testbed specific documents were converted to hypertext. All of these documents are accessible through a WAIS (Wide Area Information Server) full-text search and retrieval [WAIS]. Figure 1 is the result of a WAIS search of software testing articles.

HTML (HyperText Markup Language) was used to decorate text with hypertext tags (links and anchors), and to make explicit the logical structure of documents [HTML]. A client-server relationship is a fundamental assumption behind the use of markup languages and related presentation clients (viewers). That is, authors embed tags in their documents to make the logical document structure discernible by client viewer programs. For example, an author may wish to organize information as a bulleted list. Figure 2 shows the document as authored, and the document as presented by two client viewers (Mosaic and Lynx [Mosaic, Lynx]).

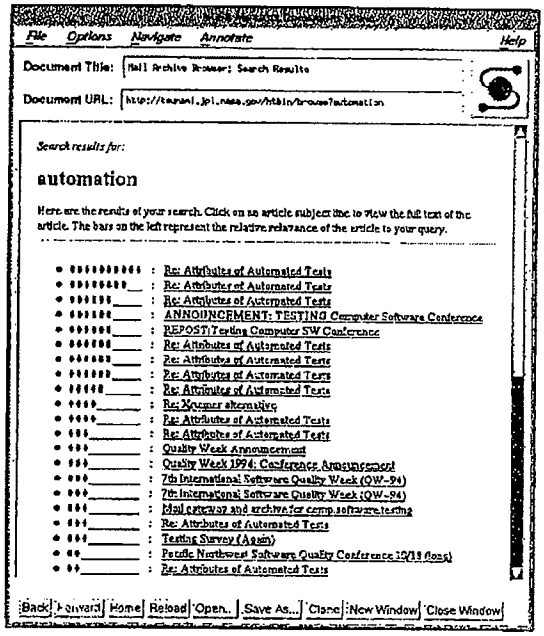


Figure 1. Result of WAIS search

```
<TITLE>Example </TITLE>
<H1>Example</H1>
This is a bulleted list:
<UL>
<LI> Weekdays
   <UL>
   <LI> Monday
   <LI> Tuesday
   <LI> Wednesday
   <LI> Thursday
   <LI> Friday
   </UL>
<LI> Weekend
   <UL>
   <LI> Saturday
   <LI> Sunday
   </UL>
</UL>
```

Figure 2. HTML, Mosaic and Lynx example (cont'd on next page)

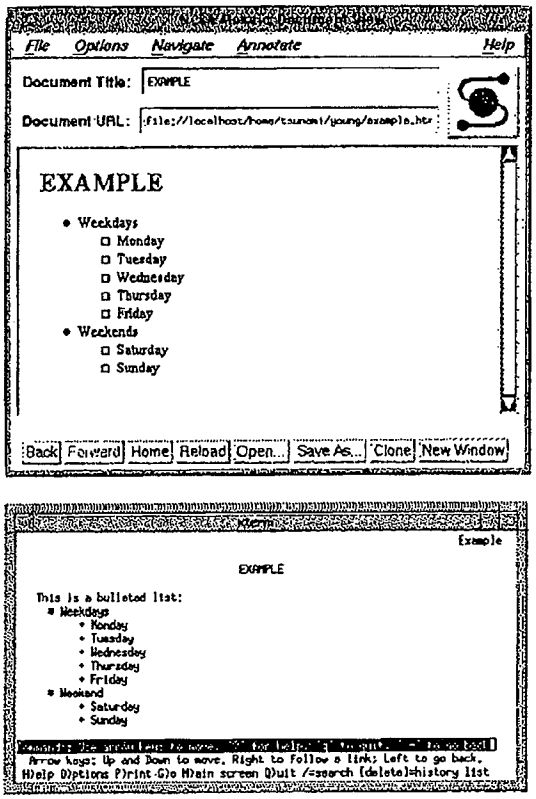


Figure 2 (Cont'd). HTML, Mosaic and Lynx example

It is important to note that the format of the presentation is determined by the client interface. The advantage of this separation of logical structure and format is that HTML clients exist for several platforms. A disadvantage, however, is that authors cannot be sure of exact placement of objects on users' screens. This is unacceptable for certain engineering and operations tasks.

The TEL prototype demonstrates the use of graphical data to resolve this problem. Graphical data can be traditional images or documents requiring a specific display format. Mosaic invokes data-specific viewing applications during the interpretation of an HTML document. For example, mission Sequence of Events (SOE) schedules and Space Flight Operations Schedule (SFOS) timelines are difficult to represent in HTML. The SFOS is a graphical timeline representation of critical information contained in the SOE. The prototype maintained a uniform user interface by launching special viewers for these documents from Mosaic. Figure 3 is the result of a query for an SOE segment.

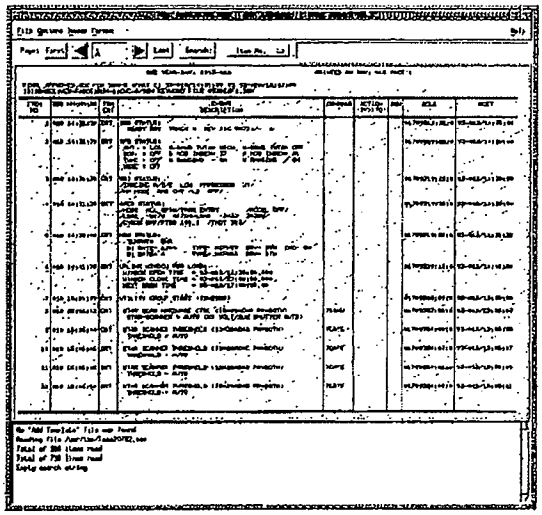


Figure 3 An SOE segment.

Finally, the prototype's most innovative aspect is the access provided to the existing Anomaly Tracking System (ATS) database of failure reports (FRs). The ATS is essential to the daily work of JPL testers. The prototype allows ATS information to be queried in a straightforward way by any combination of spacecraft, subsystem, criticality, date, and other criteria.

Previously, access to an FR database required the use of a commercial relational database interface, or telephone calls to support personnel requesting that a query be submitted. Using the capabilities provided by Mosaic it is possible to significantly simplify query formation and submission. This makes the FR database accessible to users unfamiliar or uncomfortable with relational databases. No modification to the existing ATS system was necessary.

Figure 4 is the search form as it appears using a Mosaic interface. Users compose a query by clicking buttons to choose menu items. The form in Figure 4 has been set up to choose a "listing" format of all open failure reports. The query is submitted by clicking the "generate" button. This new interface provides simple and consistent

access to users from any workstation. Users have reported a reduction in time required to access the ATS and an increase in utility of the ATS system. The result of the query is shown in Figure 5.

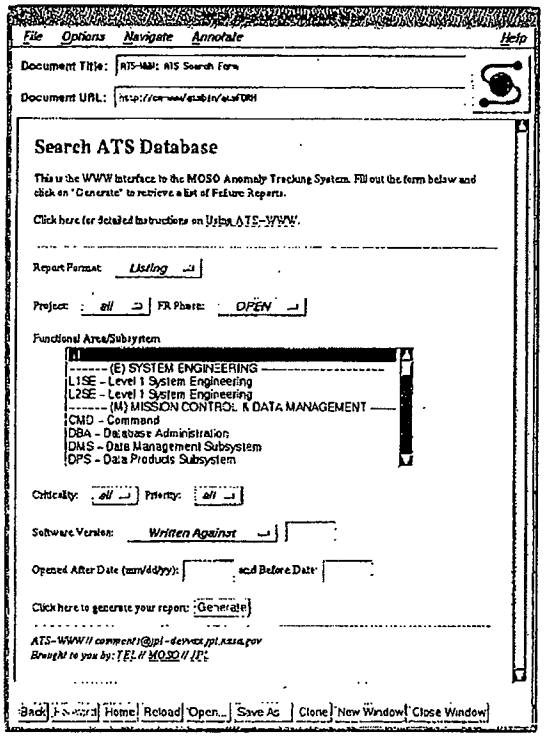


Figure 4. FR database query form

The Mosiac interface to ATS was implemented using a Common Gateway Interface (CGI) extension to a World-Wide-Web (WWW) server[2] [WWW, CGI]. CGI extensions are used to create interactive documents. Figure 6 illustrates how CGI defines the interaction between a WWW server and programs run by the server to carry out special client requests. User inputs are encoded by Mosaic as special Uniform Resource Locators (URL) and passed to the WWW server [URL]. The server invokes the CGI application and passes the user's inputs to it. The CGI application then carries out the user's request (e.g., extracts data from a database) and sends the result back to the WWW server in HTML format. Finally, the WWW server forwards the result back to the client viewer for presentation to the user.

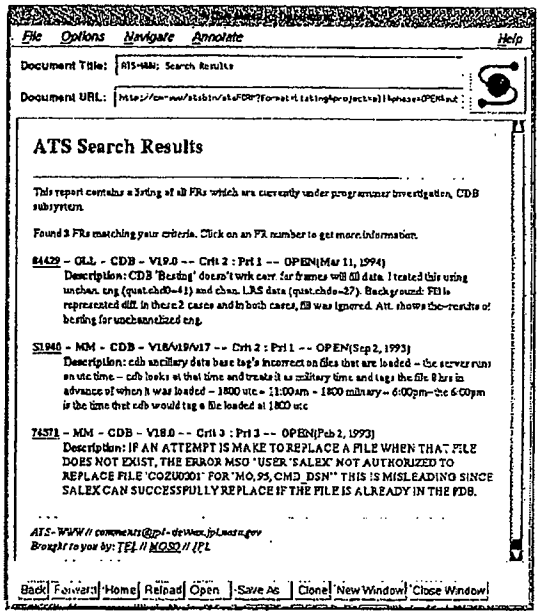


Figure 5. Result of FR database query using Mosaic interface

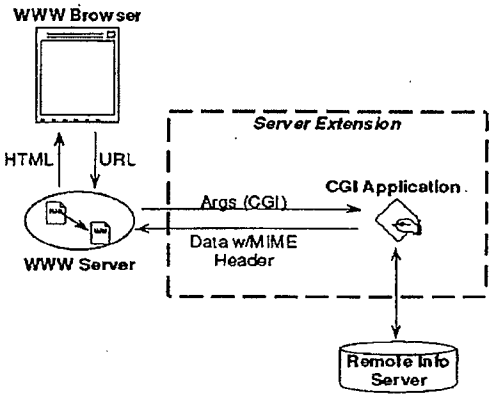


Figure 6. WWW server extensions

[2]NCSA's httpd v1.3 was used for both the prototype and delivered system

Development of the prototype

The prototype system was developed over 12 weeks by three people. It consisted of approximately 2000 pages of hypertexted hard copy documents, and 1500 lines of Perl scripts to interface with the existing ATS database front end [Wall and Schwartz].

One of the advantages of using HTML and Mosaic viewers was that potential users were able to see working prototypes quickly as development continued.

The prototype has provided a foundation for future work by demonstrating user-level integration of separate information systems and providing a uniform view of these systems across workstations.

HTML and Mosaic were chosen over other systems for several reasons. Adobe Acrobat[3] offers excellent cross-platform document browsing capabilities, but provides only rudimentary support for hyperlinks and does not support client-server interaction, making it difficult for one server to support multiple platforms over a wide area. Hyperman [Crues], developed at the Johnson Space Center and based on Adobe's PDS (Page Description Language), allows personal annotations and stronger hypertext capabilities, and will support the client-server model in the future. However, neither of these tools support "on-the-fly" document generation required for access to ATS, nor do they allow integration of user-defined viewers for unanticipated data types.

Current status

The TEL's prototype system has become a product supported by the Multimission Operations Systems Office (MOSO). The production version includes a hypertext form for submission of change request (CR) queries, as well as forms for submission and update of FRs and CRs. A larger effort is under way to convert AMMOS user documentation to HTML format, and the Cassini project is making much of its project documentation available through HTML clients.

[3]Acrobat is a trademark of Adobe Systems Incorporated.

Future work

One problem with using client user interfaces to interpret tagged hypertext documents is that clients may interpret logical organization tags in documents as suggestions rather than commands. Clients are free to display documents in idiosyncratic ways. In practice, the behavior of clients is not as anarchical as it sounds.

Because of the necessity of absolute format control in some engineering and operations documents, the TEL is continuing to evaluate extensions to HTML. In particular, HTML+ [HTML+] promises to provide increased support for mathematical symbols, tables, change bars, and floating panels (sidebars).

Second, future prototypes will allow testers to attach "personal annotations" as well as MIME (Multimedia Internet Mail Extensions) [MIME] format objects (i.e., screen dumps, core files, support documents, etc.) to FRs.

Third, the Deep Space Network (DSN) maintains a similar problem report tracking database accessed by sites worldwide. A system based on the TEL prototype and MOSO ATS product is being developed.

Summary

The TEL prototype demonstrates an integrated, consistent view of existing distributed information systems using low cost tools. In some cases, greater integration is achievable using hypertext (i.e. linking references to FRs in documents to the FRs themselves). Making information available in this way reduces delays due to information not being readily accessible when needed.

Acknowledgments

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Specifically, David Hermsen, manager of the TEL, recognized the potential of this work and supported the effort. John Louie assisted with the prototyping of the Perl interface to the ATS. Tho Le developed the ATS production system interface. Diane Miller and Debbie Tsoi-A-Sue provided logistical support.

Availability

The URL for the TEL's homepage is http://tsunami.jpl.nasa.gov/tel-home.html.

References

[CGI] The CGI specification is `http://hoohoo.ncsa.uiuc.edu/cgi`.

[Crues] E. Z. Crues. HyperMan 2.0 Enhanced Electronic Document Viewing. Presentation to KSC Mini-Workshop on Electronic Documentation, February 1994. Dr. Crues' email address is `ezcrues@gothamcity.jsc.nasa.gov`.

[HTML] The specification is `http://info.cern.ch/hypertext/WWW/MarkUp/HTML.html`. A beginner's guide is `http://www.ncsa.uiuc/demoweb/html-primer.html`.

[HTML+] The draft HTML+ document is `ftp://ds.internic.net/internet-drafts/draft-raggett-www-html-00.txt`.

[Lynx] Lynx is described in `http://www.cs.ukans.edu/about_lynx/about_lynx.html`.

[MIME] MIME is described in RFC 1341, RFC 1343, and RFC 1344, available at `ftp://ftp.internic.net/rfc`.

[Mosaic] The home page for Mosaic is `http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html`.

[URL] An overview of URLs is `http://www.ncsa.uiuc.edu/demoweb/url-primer.html`. The specification is `ftp://info.cern.ch/pub/doc/url-spec.txt`.

[WAIS] A bibliography of WAIS documentation is `ftp://quake.think.com/wais/bibliography.txt`.

[Wall and Schwartz] L. Wall and R. L. Schwartz. *Programming perl*, O'Reilly & Associates Inc., 1990.

[WWW] Links to documentation about the World-Wide-Web, including a bibliography, are at `http://info.cern.ch/hypertext/WWW/TheProject.html`.