HyperText Markup Language (HTML) and the Web
Novel New Aids to Testing Automation and Information Access

Jon Hagar Phil Bell Earl Burba Debra Wittekind
Lockheed Martin Astronautics
Mail Stop H0512
P.O. Box 179
Denver, CO 80201
303-977-1625
303-977-1472 (fax)
hagar@den.mmc.com

Keywords**:** Hypertext Markup Language (HTML), the World-Wide Web (WWW), Test Tools, Configuration Management (CM), Test Documentation, Reuse, Accessibility
Audience: Application Oriented and General

Abstract:

In just a few short years, the World-Wide Web has become the most popular vehicle for receiving and distributing global information. The World-Wide Web is easy to use, cost-effective, and it is quickly becoming a marketplace for commercial advertisers and publishers as well as individuals seeking information. At Lockheed Martin we have also been successfully using this new technology to distribute information internally at the corporate and project levels as well as assisting with engineering activities like software testing and quality assurance. Using the proper tools and configuration, the World-Wide Web can lend itself to internal document maintenance, configuration management, software test automation and support of a repeatable test processes. We have successfully implemented tools in these areas, most notably one that generates tests and supports execution of software testing. For most companies in the United States, the tools necessary to implement the World-Wide Web are already in place or readily accessible. The requirements include computers with Internet/Network access and Web support software, which can be obtained without charge in some cases.

1.0 Introduction

The most rapidly expanding area of the Internet has been the World Wide Web (WWW a.k.a. the Web). It is flashy, fun, new, intriguing and “all the rage”, but can it be a tool for engineering and project use? Can the Web, home pages, information

pages, and HTML be used to aid and trap understanding needed to do quality test engineering?

Our product area at Lockheed Martin Astronautics (LMA - formally Martin Marietta) has been developing a variety of support and documentation aids based on the WWW and Hypertext Markup Language (HTML). We find that Web technologies offer information sharing that can be very useful to “real world” projects. Our particular area of application is use of the WWW to support software test engineering activities associated with a long term, on-going verification and validation project.

We have implemented test execution support tools and information access on a local segment of the Web. For our project, an important aspect of testing is the ease and quick access of correct test information. Information that is important to test includes:

- Product configuration control and management;
- Test documentation in the form of plans, procedures, results and reports;
- Accessibility of product versions; and
- Test automation, repeatability, traceability, and process.

In this paper we will describe how the Web approach to information directly supports us in these areas. We have implemented executable test procedures in HTML that were themselves generated by HTML programs. These test procedures have numerous advantages including being all electronic and able to interface with other Web resources. These are resulting in more efficient use of engineering time, parallel reviews, and less paper. Our early excursions in this domain demonstrate that real contributions can be made to production grade testing and engineering projects. The following lists descriptions of tools that our project has implemented based on the Web/HTML:

- Automatic test procedure generation via an easy graphical user interface (GUI)[TE5, see 4];
- Test procedure execution and results retention [TM6, see 4];
- Configuration and information management with easy on-line access to project documentation and software including controlled files, test tools, status, and reference information[TM1, see 4];
- Reuse and asset repository/library; and
- Advertising of project capabilities and information (a common Web theme)

We see this media as a viable tool that is still in its infancy. Our work to date has been internal to the company firewall with only limited access to other parts of the company. This is because of company procedures and the nature of some of our work and/or the information we provide.

This paper is written as an introduction to our work in this area. We define the type of testing and environment to which we are applying Web technologies. We outline our Web related applications and introduce some of how we have developed these tools using WWW technology. A final section looks at the advantages and lessons encountered during this work as well as our plans for continuing efforts.

1.1 Our Project Test Environment

LMA in Denver, Colorado, has been producing and testing critical software systems for several decades. Our test group is responsible for the verification and validation of flight control software. An example mission profile, including mission and vehicle characteristics that the software must support, is depicted in figure 1. The software that we test is: embedded within and interfaces with the hardware components of this system; functions in hard real time; and is responsible for guidance, navigation and control of the system. Our testing shows that the software will accomplish its mission, be reliable, and have good quality. However these goals must be accomplished within cost and schedule constraints. These constraints lead us to look for automation and processes improvements within our test environment

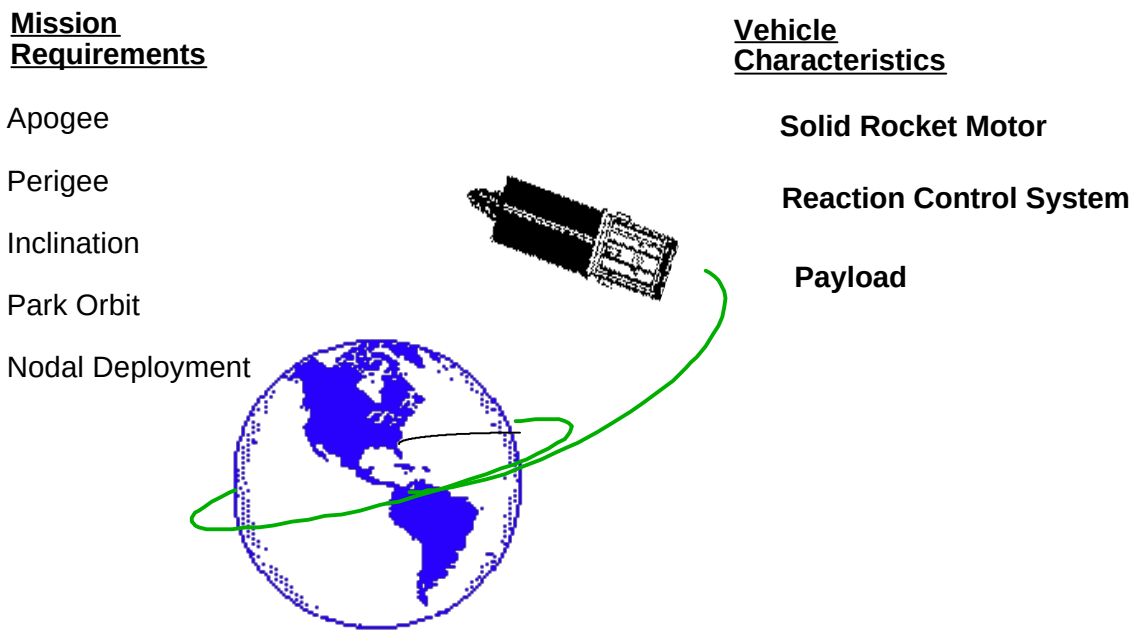

Figure 111 - Complex System-Software Requirements

Our test environment is a laboratory that consists of two different types of computer systems:

1) an avionics test bed system which consists of flight computers and interface hardware, and

2) simulation and test support systems which execute on UNIX workstations.

The lab is connected to a network that supports over twenty engineers in our group. Interconnected with this network are workstations (Suns and Silicon Graphics) and microcomputers (IBMs and Macintoshes) that are accessible to each engineer and management. This system of computers is used to support testing, analysis, and simulation at various levels from unit to system testing.

As with most test areas[1], our testers are faced with a series of problems:

- Product Configuration Control and Management - numerous tasks and software versions exist, necessitating the need for Configuration Management (CM) of products specific to particular test activities.

- Information Access - Quick accessibility to a product's test documentation in the form of plans, procedures, data, software, test results, and reports; and

- Test Design and Execution - We must create and execute tests, which includes the use of numerous computers, commercial products, and custom tools as well as manual activities necessitated by our test environment.

We have already implemented numerous computer aided software engineering (CASE) technologies within our basic test processes[10]. We began looking for additional supporting technologies that would solve some of the above issues, allow for customization through some programmability, and support our distributed test environment of multiple workstations and microcomputers. A study led to our interest in the possible use of WWW concepts, mixed with our existing environment [2] to solve some of the above problems.

2.0 Implemented Uses of WWW Technologies

Most of us have seen the Web and "surfed" its pages. It can be fun and there is an abundance of information available. As a technology grows, a good measure of its worth is its ability to be used to solve different problems. The Web certainly offers advertisers and entertainment a new outlet. It also expands the Internet to be a more user friendly world through the WWW's GUI. But more interestingly to us, it offers practicing software test engineers a supporting media that can aid in solving some of the problems of testing. In the following sections, we outline what we have done.

Figure 2 shows our project's home page. We have incorporated buttons to allow access to our software test tools (documentation), software quality assurance (SQA) controlled test documentation, group information (such things as status reports and phone numbers), as well as a variety of other buttons. This serves as a standard entry point for project engineers in their daily efforts[3]. An information button takes non-project visitors to a page that offers information about our group's capabilities. From this page, the areas discussed in subsequent sections can be accessed.

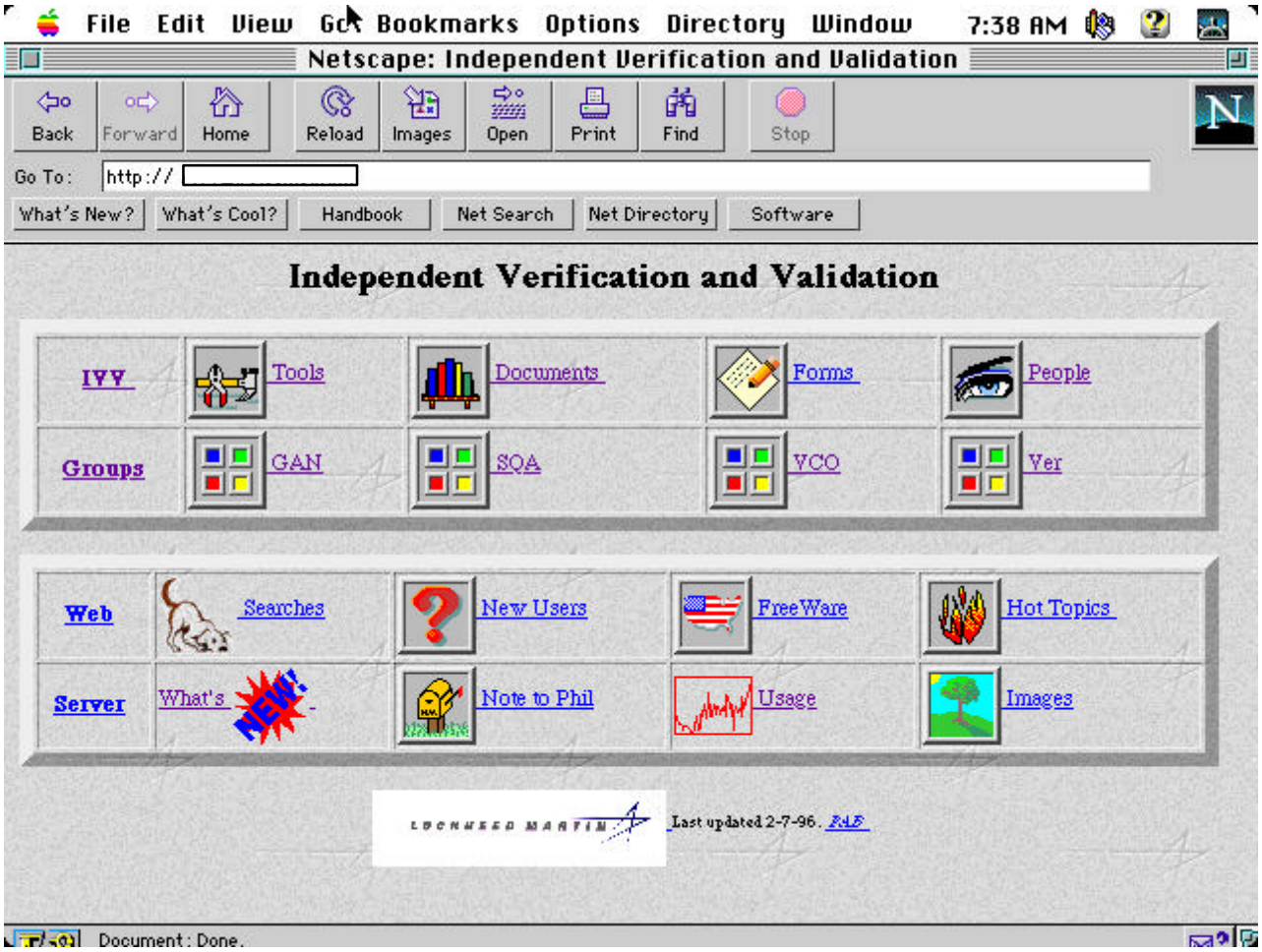

Figure 2 Testers Home Page

2.1 On-line Software Quality Assurance (SQA) and Configuration Management (CM) Pages

A major issue for testers is that information (files, documents, tools, programs, data, drawings, etc.) used in testing must be easily and quickly accessible, correct and current. Modern software systems have numerous levels of documents (requirements, design, code, data, executable, and test) all of which must be managed efficiently. Providing the CM information, controlling the files (e.g., write protected), and accessing the data are all necessary for success.

We have established informational pages for each product configuration under test. Within these pages are links to correct files and documentation. These links are defined and used by testers but maintained and controlled by our internal SQA group. This separation allows better control by SQA which in turn supports better

review and audit by the SQA group. The system is online and easy to use for quick support of project test documentation needs.

Some of the activities in this area that previously involved manual transfer, review and input have been automated with the use of the Web. These areas include:

- Notification and access to software files and documentation deliveries via email generated by Web support tools;
- Identification of and access to the most current software configuration using a Web browser; and
- Consolidation of project configuration into one location on the Web server.

This automation has resulted in :

- Timely electronic notification to engineers and correspondents;
- Improved status identification that in turn is accessible to other Web based tools;
- Improved configuration control because of the server; and
- Improved test design and execution, since tools have direct access to Web information on the same system that they are executing.

The SQA home page is created in HTML. The tester's home page links, along with the SQA page, allows the engineers to access the SQA area with a click of the mouse. The ability to access files is available from any project computer system assuming the recipient (client) has the correct application software.

The maintaining engineer can give a file a specific name (link name) to make access "a click of the button" or the "call of a variable" within code. The link name is then included in the HTML code which presents it to the user. For example (figure 3), if the latest tool version resides in a particular directory path and is named PCTS_rev_4, a UNIX and HTML link can be set up so that the SQA page only reflects the words "PCTS". Upon clicking on "PCTS" the UNIX and HTML code will automatically go to the directory/filename path. The information resident within the SQA page allow the engineers and/or the automated test procedures to access to the latest configuration without much more effort than a link to the SQA area. We have found that a holistic system is more useful. This is one where we encompass both the user's and tool's needs in how we establish links, pages, and associated notations [5].

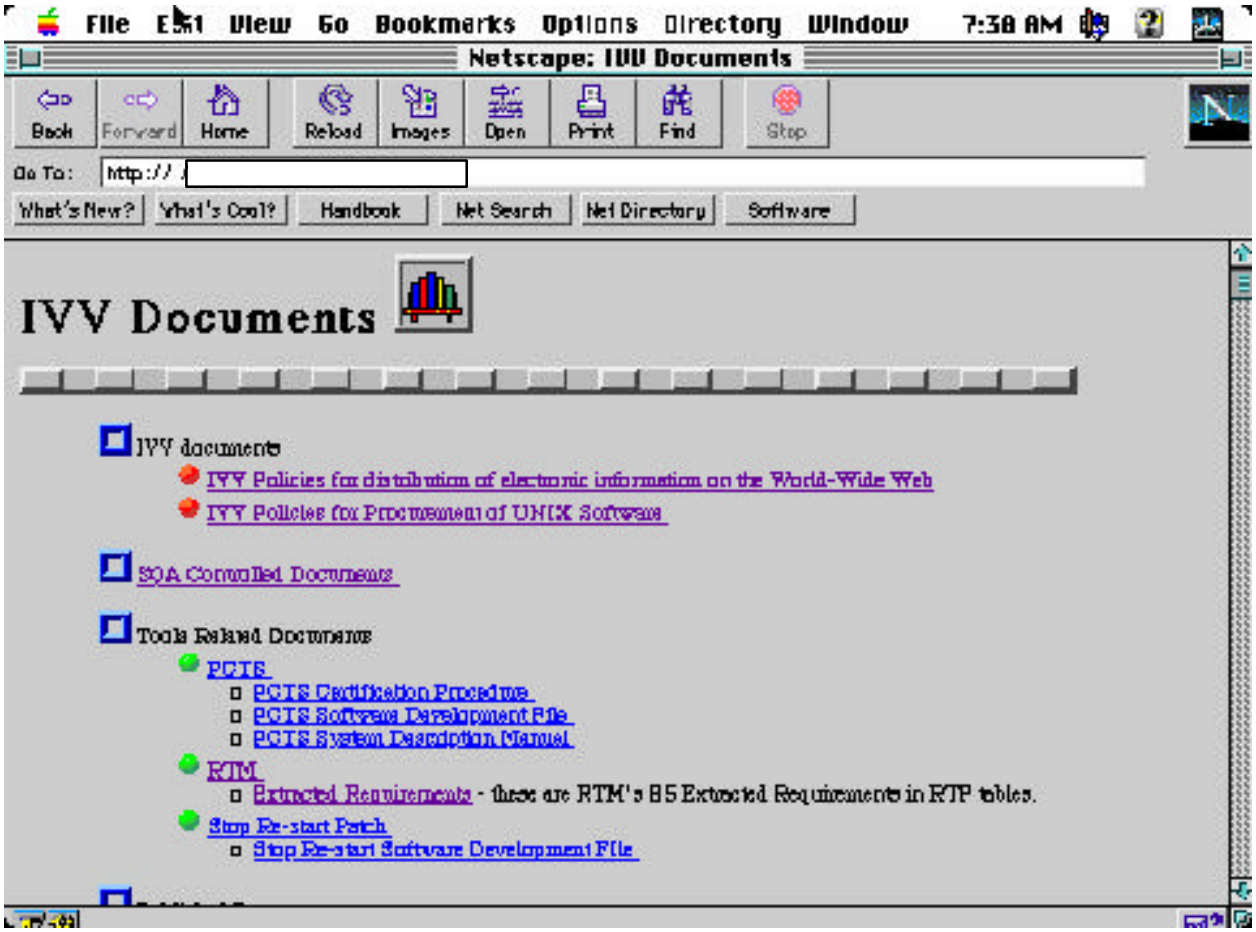

Figure 3 - Documentation Under SQA Control/Access Page

2.2 Electronic Test Procedures Written in HTML

Another significant area we have used Web technology in is test automation. Here we have created a tool that allows the generation of executable test procedures, both done using HTML.

We have created various templates and forms that allow first the generation of specific test case procedures, which we call software test and evaluation procedures (STEPs) based on project standards, and then the execution of tests using the completed HTML-based test procedures in an on-line, interactive mode. This approach highly automates and standardizes test procedure generation and execution. Tests are designed by filling in HTML template forms which in turn generate test procedures in the form of other HTML documents. The basic process is seen in figure 4 and is initiated with an HTML-based tool called STEPGEN.

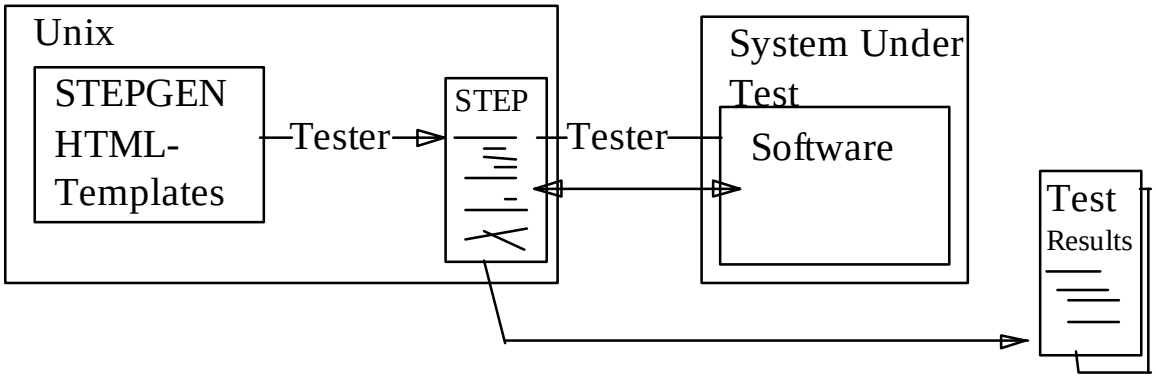

Figure 4 **STEP Generation Process**

STEP design is done by the test engineer (Tester), who specifies all input fields as well as a path name to store and execute the STEP. All STEP inputs and outputs, shell files used, and the STEP itself, are stored in the specified location for review and analysis by engineering after the run, as well as ultimate SQA CM. We have a variety of standard inputs that must be completed as well as customization options with each test procedure. These are filled in by either: 1) selecting a preprogrammed selection list that is linked to the SQA CM page and project libraries, or 2) entering test unique information into pre-defined places.

As fields are entered, the system "builds" a new HTML file ("STEP" in figure 4) that is itself an executable test procedure. In order to allow the user to process smaller pieces of the STEP, the STEPGEN offers the feature of partial submittals at the end of logical sections of the STEP. The user can then exit the browser and begin generation of the later STEP sections at a subsequent time. This supports test design over a number of sessions or designers.

The custom procedural section of the STEPGEN tool offers the user the ability to use previously designed procedure items or custom procedure fields, allowing great latitude in design. Additionally, custom and pre-designed procedures can be mixed freely with order independence.

When completed, STEPs are submitted for electronic on-line review and approval. Once approved for use, the STEP file can be executed via the Web. Test procedures prompt testers for actions interactively and spawn actual test tool execution, as well as providing both the input and output retention.

2.2.1 Using the Resultant Electronic STEP

Following the approval of the STEP, the tester executes the STEP by launching a Web browser and opening the URL to the directory/path name where the STEP was created. A STEP Navigator allows the tester to navigate to any of the test sections easily. By following the instructions and procedures and calls within the STEP the tester is assured of achieving the desired results.

The executable part of the HTML STEP has the following parts:

- Test Prerequisites, any special H/W, S/W setup required;
- Test Preparation, configuration of systems and preparation for the STEP execution;
- Test Execution, execution of pre-designed procedures; and
- Test Termination, gracefully completing the procedural part of the STEP.

The tester follows the instructions or executes shell procedures from within the STEP to test the software on the "System Under Test". As sections of a test procedure are completed, a tester selects a submission button that causes a report CGI to execute. The report CGI saves all inputs/entries up to that point into a Test Results file. This file is used to retain results and shows which procedures were satisfactorily completed. The tester then reviews output for compliance with success criteria. This may include automated checks and analysis to ensure they function as intended.

After the STEP has been completed, the tester notifies SQA and management, informing them through e-mail. By using the electronic "as run" STEP results section and the associated hyperlinks, the reviewers can easily view inputs/outputs and results. When the review is completed, an HTML review form is used to inform the tester of results of the review, whether any outstanding issues are still open, and if regression testing is required. If SQA and the functional group lead approve the STEP results the entire directory is either archived to tape, or access rights changed to limit write access and possible modification, ensuring repeatability and CM of the test itself.

2.3 Non-SQA Project Documentation Repository & WWW Library

We have placed our plans, status reports, resumes, software development files, software test tools, test report and other information in a project library which is accessible by all team members. Information placed in our library includes test cases and test software information as well as presentations and papers (e.g., like this paper) written by project personnel in a variety of formats and implementations.

These provide a repository of lessons learned, reference information, and current documentation accessible to all project staff. Additionally, for non-project personnel (within LMA), we have provided access to portions of this library to aid in company reuse efforts. Software reuse, including more than just code products, is supported by the Web home pages, buttons, and indexes. Additionally, since the reused products are maintained locally (within the project), we maintain versions and control of these libraries. This approach of libraries using the Web and HTML is becoming a company standard for software and information reuse libraries. Project and central Web pages direct users to local pages that contain the specific information.

Figure 5 shows our test tools page. Here we locate and control the documentation of tools, both commercial and custom built, that we need to do our testing. This information is available at any point to engineers. This proves useful when designing tests or troubleshooting test results if we suspect problems with our tools. This information also facilitates reuse on other test programs both by our engineers and others within LMA, as the page is accessed via external links and other home pages within LMA. These libraries are dynamic in nature and change as information is added and problems with tools or documentation is fixed.

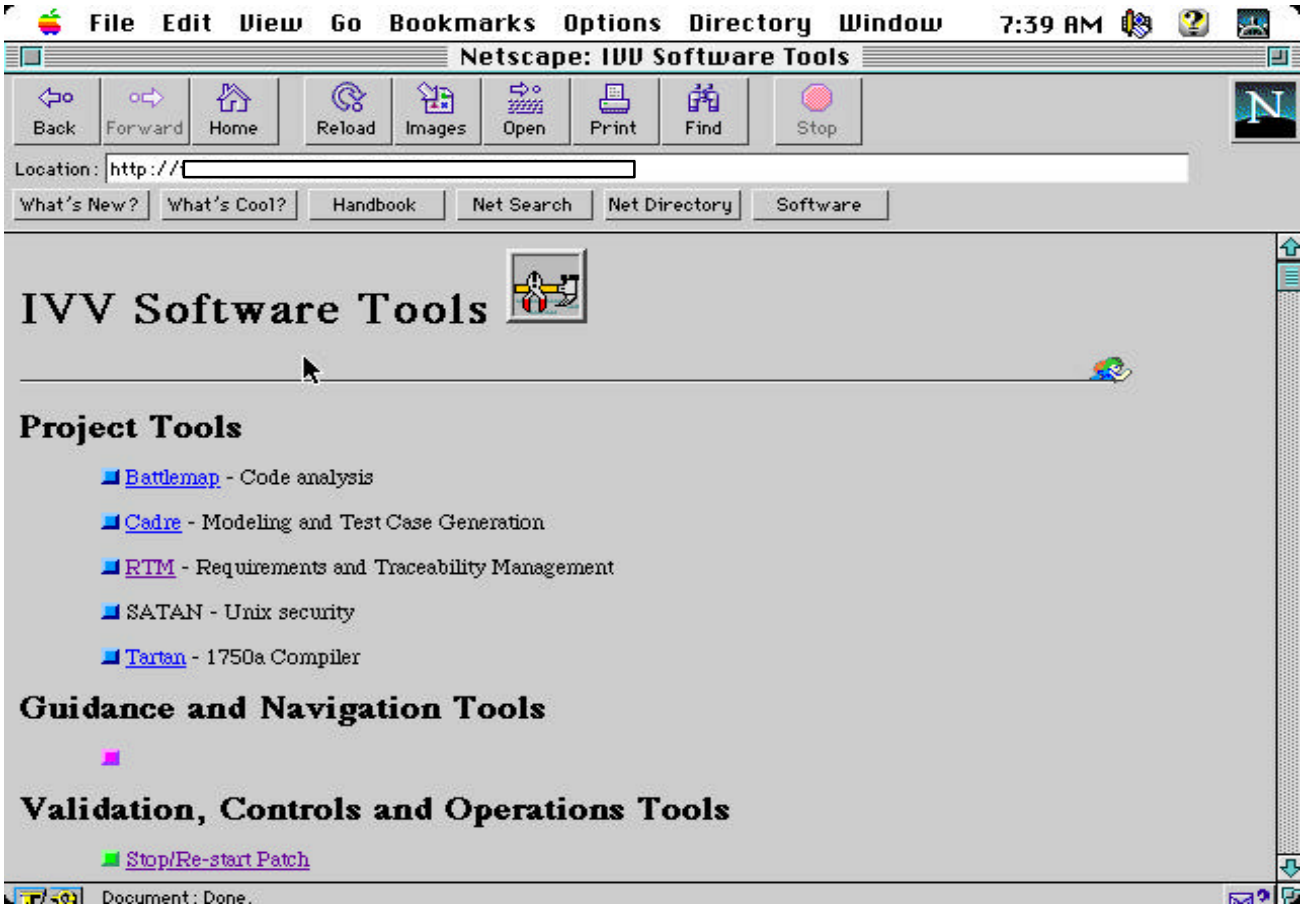

Figure 5 Tool Library

3.0 Development of Implementation

We have implemented our system with a World-Wide-Web server which has a number of advanced capabilities that meet the needs of our project. Any individual within the corporation may access our Web server using a networked computer and "Web browser" software. It is important to note, however, that the information that is distributed via the Web is both hardware and software independent. We may at any time substitute the server software with a number of other free or commercial server packages. The World-Wide Web pages and documents can be quickly re-hosted to another server platform (Macintosh, PC, or UNIX), and end users can receive the information without noticing the change.

Our server exists on a Sun/Solaris UNIX platform which can distribute information to Macintosh, PC, and other UNIX client machines. The server software consists of several free applications, namely W3C httpd (formerly CERN httpd) 3.0 and a few support tools that assist with maintenance and enhance the server capabilities. The

server resides behind a corporate firewall for information security and is equipped with two additional methods of information protection to further restrict access of selected documents and pages to only our project. The server has the capability to distribute documents from a variety of popular commercial software, including Microsoft Office, as well as the Web's default HTML pages, Java applets, and multimedia (sound, picture, video) files. Information distributed from the server can be received from most networked computer platforms including PCs, Macintosh and UNIX workstations. Finally, the server has the capability to execute Common-Gateway Interface (CGI) scripts, which have automated some of our testing processes.

Client [6] implementation includes Macintosh, PC and UNIX workstations using Netscape, Mosaic and Internet Explorer. This multi-platform server and client capability is one of the greatest strengths of the World-Wide Web. All standard Web documents (HTML) and many popular commercial applications (Microsoft Office, etc.) can be distributed seamlessly from any Web server to any Web client. Thus, implementation of these tools can be achieved by most projects without extensive effort.

The idea of using HTML/CGI as a language and a browser, like Netscape, as a user interface came about as a result of a project directive to pursue practical uses for the World Wide Web (WWW).

4.0 Advantages of Using the WWW in Testing

We have found that use of WWW technology has numerous advantages, some of which are listed below.

- Breaks STEP generation and execution into smaller, more manageable pieces.
- Ensures consistency of STEPs through the use of approved templates.
- Provides full custom capability using a hybrid of approved templates and custom design.
- Allows easy selection of data by use of drag-down menus, check boxes, and radio buttons.
- Exploits the incorporation of "Value" fields to "hard-code" entries after they have been entered. This ensures repeatability and soft documentation of STEP inputs.
- Utilizes hyperlinks to quickly and easily view input, output, and analysis that exists on-line.

- Documents the version, if any, of all templates used in STEP generation decreasing the time needed for review of each STEP prior to approval.
- Provides a standard STEP form used by all functional groups, adding consistency to the STEP.
- Intuitive and easy to use GUI.
- “Educational” links are available to help the user with the STEP generation and execution processes.
- Browser are device independent and allow multi-platform use with consistent results. The document is rendered successfully on whatever platform it is viewed.
- HTML is a runtime language requiring no compilation.
- Implementation tools are readily available (even free sometimes).
- Use of “electronic only” features like CGI and GUI interfaces helps eliminate paper and encourages a paperless office.
- Engineering a light speed by the facilitating quick and parallel production, execution, review, analysis, and audit of automated test procedures.

5.0 Lessons Learned from Setting Up a Web System

While this system works and is useful, it was a learning experience setting it up. Issues were raised early-on by management about the security of data and whether this was more a system for “play” than work. Both of these have been addressed by standard company policy and properties of our implementation. We continue to have concerns about how quickly this technology area is changing. There is always a danger when you are at the cutting edge of a technology that may be heading down a path that becomes a “dead end”. We believe that by staying current with Web technologies and trends we will minimize these risks.

Below are some specific issues with our current implementation.

- Restrictive I/O, there is no provision for reading files of data (solution would be to create Common Gateway Interface (CGI) tools in PERL or C).
- HTML and CGIs are susceptible to corruption as it parses input strings with special characters (like <, $ etc.).
- Initial development of some features took planning and development time that increased as the complexity of the system expanded.
- Net traffic tends to slow response time.

6.0 Future Efforts

We find this technology encouraging in that it supports many of our goals in testing and documentation. We will continue to explore the application of this technology and stay current with industry trends. We see the GUI interface of some of the pages needs better human factors engineering, and hope this will change as users and producers understand information access better. Additionally, the integration of our pages with reporting methods, planning tools, and other management tools seems logical. It is conceivable that the customer may benefit from access to some of the Web pages directly, though issues of company security would need to be addressed before this would be possible. Also, technical upgrades to new versions of HTML, as well as Java, and “applets”, will be explored.

7.0 Summary

Our project, which performs independent verification and validation of flight software, along with others at Lockheed Martin, are showing that the World Wide Web has much to offer towards improved state of the art technology. World-Wide Web technology is a viable means of corporate internal and external information sharing, as well as a software test support tool. The capability of providing a “user friendly” interface to quickly access and distribute current and controlled configuration files, documentation, and test data is invaluable to a project such as ours. Further, the Web and HTML’s ability to support generation of hyper documents that directly support and document testing shows direct support of engineering activities. Our usage of the WWW is only just beginning. For additional and starting information on the Web see [7,8,&9].

8.0 References

1. Kaner, Falk, & Nguyen, Ch 12 & p 430 source versions - Test Planning and Documentation, Testing Computer Software Van Nostrand Reinhold, 1993

2. Larry Press, “McLuhan Meets the Net”, Communications of the ACM, June 1995, Vol 38 # 6. (Quote on hybrid)

3. Douglas Engelbart, “Toward Augmenting the Human Intellect”, Communications of the ACM, August 1995, Vol 38, # 8.

4. Dorothy Grahm, “Software Test Tools: A New Classification Scheme”, Journal of Software Testing Verification and Reliability, Oct-Dec 1991, Vol 1, #3.

5. Rivlin, Botafogo, & Shneiderman, “Navigating in HyperSpace: Designing a Structure-based Toolbox”, Communications of the ACM, June 1994, Vol 37 #2.

6. Hal Berghel, The Client’s Side of the World Wide Web”, Communications of the ACM Jan. 1996, Vol 39 # 1.

7. User Documentation on Netscape Home Page, Documents under - http://home.netscape.com/toc.HTML, Copyright © 1996 Netscape Communications Corporation.

8. A Beginner's Guide to HTML, Document under - http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.HTMT National Center for Supercomputing Applications / pubs@ncsa.uiuc.edu, 1996

9. WWW Development Info http://WWW.Stars.com/Vlib/ and http://www.yahoo.com/

10. “Testing Critical Software: Practical Experiences”, J. Hagar, Safety and Reliability in Emerging Control Technologies, Proceedings of the Second IFAC Workshop, 1995.

BIO

Jon Hagar Phil Bell Earl Burba Debra Wittekind
Lockheed Martin Astronautics Company
Mail Stop H0512
P.O. Box 179
Denver, CO 80201
303-977-1625
303-977-1472 (fax)
hagar@den.mmc.com

Speaker: Debra Wittekind Biography
Debra is a lead engineer in software quality assurance engineering with Lockheed Martin Astronautics in Denver, Co. She is currently assigned to software/hardware quality on a realtime flight software test project. Debra holds a B.S in Computer Information Systems and Management Science from Metropolitan State College of Denver and an A.S in Computer Programming from Brevard Community College in Florida.. Debra has over 15 years experience in evaluation of Space Launch Systems design, development, test and configuration, including the Space Shuttle and Titan IV. Debra's interests include software engineering, quality assurance, and WWW technologies.

Authors:
Jon Hagar Biography
Jon Hagar is a lead software engineer supporting software verification and validation testing at Lockheed Martin Astronautics in Denver, Co. He has a B.S. Degree in math with specialization in civil-engineering and software from Metropolitan State College of Denver, CO, and a M.S. degree in computer science with specialization in software engineering and testing from Colorado State University. Jon has worked in software engineering, particularly testing/verification and validation, for over 15 years. The projects, he has supported, are primarily booster and space related. He has experience in the software domain of real-time reactive embedded control systems as well as test software development using numerous languages, including JOVIAL and Ada. Jon is a member of the ACM and IEEE. He teaches classes at Lockheed Martin Astronautics, and has published articles on software reliability, testing, formal methods, and critical-systems, as well as having presented at NASA and Software Productivity Consortium working groups. Jon's work interests include software testing, verification, validation, system

engineering, reliability, neural-networks/GAs in testing, test support tools, and quality assurance.

Phil Bell Biography

Phil is an associate engineer in software engineering with Lockheed Martin Astronautics in Denver, Co. He is currently assigned to software test bed setup and programming for a flight control system, working with electronic, digital computer and software aspects. Phil holds a B.S. in Engineering - Electrical specialty from the Colorado School of Mines and his professional interests range from Object Oriented Design to Real-time digital feedback control systems. Phil is also the Web administrator for the IVV project.

Earl Burba Biography

Earl is an senior engineer in software engineering with Lockheed Martin Astronautics in Denver, Co. He is currently assigned to verification testing and software test tool programming. Earl a B.S. degree in computer science from Colorado State University and an MIS degree from the University of Denver. Earl’s interests include WWW technologies, verification, validation, IVV, and software engineering.