

1995

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

**MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE**

WEB INTERFACES TO RELATIONAL DATABASES

Prepared By: W. H. Carlisle, Ph.D.
Academic Rank: Associate Professor
Institution and Department: Auburn University
Department of Computer Science and Engineering

NASA/MSFC

Office: Program Development
Division: Advanced Systems and Technology
Branches: Advanced Concepts Office,
Space Science and Applications

Office: Science and Engineering Astrionics Laboratory
Divisions: Avionics Simulation Division, Software Division
Branches: Computer Systems Engineering,
Requirements and Technology

MSFC Colleague(s): Vance Houston & Dan O'Neil, MSFC Colleague
Les Johnson, MSFC Colleague
Steve Jones, Ron Newby, MSFC Colleague
Brandon Dewberry, MSFC Colleague

This reports on a project to extend the capabilities of a Virtual Research Center (VRC)¹ for NASA's Advanced Concepts Office. The work was performed as part of NASA's 1995 Summer Faculty Fellowship program and involved the development of a prototype component of the VRC -- a database system that provides data creation and access services within a room of the VRC. In support of VRC development, NASA has assembled a laboratory containing the variety of equipment expected to be used by scientists within the VRC. This laboratory consists of the major hardware platforms, SUN, Intel, and Motorola processors and their most common operating systems UNIX, Windows NT, Windows for Workgroups, and Macintosh. The SPARC 20 runs SUN Solaris 2.4, an Intel Pentium runs Windows NT and is installed on a different network from the other machines in the laboratory, a Pentium PC runs Windows for Workgroups, two Intel 386 machines run Windows 3.1, and finally, a PowerMacintosh and a Macintosh IIsi run MacOS.

Summer Activities Overview

The project's completion will provide a World Wide Web interface into a database system, the Advanced Concepts Information System (ACIMS). Initial activities consisted of network installations and software configurations within the laboratory so that all systems would be Web clients. One client system within the laboratory, the Windows NT machine, was installed on a different network (138.115.xx.xxx) from the other systems (128.158.xx.xxx). Additionally a Mosaic web server was installed on both the NT and on the SUN/SPARC 20 machines. To support software development on the SUN/Solaris machine, the gnu C compiler, a Perl interpreter, and mSQL -- a public domain (for nonprofit institutions) database server, were retrieved from the network and installed on the SUN. During this time planning activities of the project were conducted. This involved investigations into current Web interfaces to database systems, and the installation of one such system for testing. None of the existing systems was as desired for the VRC system, and consequently a prototype system consisting of the mSQL database and Web pages that query and insert information into this database was implemented. Both C and Perl CGI scripts were written to interface the Web to the database server. The following sections present in more detail the information associated with this overview.

An Interface to the Virtual Research Laboratory: The World Wide Web

The World Wide Web (WWW)² has become the de facto standard for sharing resources over the Internet, largely because web browsers are freely available for the most common hardware platforms and their operating systems.³ The WWW provides the opportunity to share images, video, audio, etc., and extensions to the server standards now allow for clickable script execution. Soon to be available are animation⁴ and virtual navigation capabilities.^{5 6} Within the Web the VRC seeks to provide access to information associated with the Advanced Concepts Office. To understand how the Web can provide information to clients we first discuss how the WWW functions.

How the World Wide Web Works

The WWW acts with clients and servers. Clients request pages (screens) via an address called a Uniform Resource Locator (URL). This address contains not only information location but also specifies the way (protocol) in which communication is to be conducted. The most common communication protocol used by Web clients is Hypertext Markup Language (HTML). Requests are sent to a hypertext protocol server program (http). The client expects that a server will return text containing "markup," that provides formatting information as defined by the HTML language.⁷ The client, upon receiving the document, formats and presents it to the user. The power of the HTML languages is that other URLs can be part of the format (hypertext links). To support easy navigation the client software tracks the mouse. When clicked over the presentation of a URL link, another server request is generated.

Servers simply wait for client requests on the network, and return requested pages to the clients. Although this mechanism is the foundation for the World Wide Web, it is not sufficiently powerful to allow the server to itself become a client of other services such as a database server. For this capability it is necessary that a URL cause a server to execute a program and return the results of execution to the client. Such capabilities are provided by extensions such as the Common Gateway Interface (CGI).⁸ For this service, a client requests a URL that is a program in a special directory of the server. The server recognizes that requests to this directory are programs and runs the requested program. The client also sends information to the server that is required as input to the server programs, and so the first function of server programs is to retrieve any further information that was sent by the client, decode it, and then run the program. The client is expecting an acceptable type of document as a return, generally HTML, so the final function of the script is to generate an HTML document and return this document to the client.

As mentioned in the overview above, CGI server software for a Windows NT⁹ and a SUN Workstation¹⁰ were installed. Although the SUN system will be the primary server for the laboratory, it was desirable to experiment with the NT operating system and compare its capabilities with those of UNIX on the SUN. The laboratory's NT machine is installed on a

more secure CAD/CAM network. This network will not allow external clients, but machines on this network can be a client of other networks. This installation allows inter-network

builds a database query, then calls a database backend program to retrieve the information from the database. The script then builds a text (not HTML) return of the information. Another example of using CGI scripts to access a database server is an example system built by Oak Ridge National Labs to make a database of textile and apparel manufacturers and their capabilities available over the World Wide Web to the apparel industry (experimental site)¹³. This system is special purpose for the application, but provides a good example of how web interfaces to database systems can be built. The European Southern Observatory generated software (WDB)¹⁴ to make Hubble Space Telescope information (ST-ECF science data archive) available to the scientific community. The WDB system is a general purpose database interface, that is, not tied to a particular database, and so was investigated for its applicability to the ACIMS.

Although the interface between the WWW client and a HTTPD server is well defined, there is no standard for interfacing a database client with a database server (although efforts in this direction are being made). Currently each database system has its own separate database interface protocol, and the choice of the database determines the CGI program.

Since the http server must become a client of a database server, HTTP servers are implemented more easily within multitasking operating systems such as UNIX and Windows NT. Although the database systems for Windows NT are cheaper, they generally have their roots in DOS and cannot handle multiple clients who wish to access the database. Only the UNIX database systems having their roots in mainframe systems support the distributed database management that is desired for a networked system. Within UNIX the most common database systems are Sybase¹⁵, Oracle¹⁶, Informix¹⁷, Ingres and mSQL. The Oak Ridge software system and ESO system targeted Sybase as the database server. The gsql software targeted Oracle. Oracle was chosen for the final version of the ACIMS database, and while awaiting arrival of software to support a C or Perl interface to Oracle, an mSQL database system was installed on the SUN system. mSQL is a small, public domain, and open source database system.¹⁸

After installation of the mSQL software, a prototype database was built. The tables in this database are shown in the appendix. This database would allow for experimentation with the Web database server system known as WDB, an investigation into its possible use for the ACIMS, and experimentation with other CGI-script interfaces to database systems.

WDB

In order to experiment with the WDB system it was necessary to first install and configure the Perl¹⁹ programming language and an associated Perl interface to the mSQL database system. This interface is called MsqI.²⁰ Then the WDB software was installed. This software is similar to gsql in that it uses a single CGI script (WDB) for each query. In order to have a page associated with different tables in the database, the wdb script reads a page specification, and then builds an HTML page for the client to query the database according to the specification. The script then accepts a query, and builds a return to the client based on the results of the database query. WDB also provides software that can automatically generate a

specification from a database table. The prototype mSQL database and the script provided by WDB were used to generate a specification. The query page built by wdb from this specification is found in the Appendix. WDB is a powerful tool and could be modified to serve the purposes of the ACIMS. However, this would require interfacing WDB with Oracle. Efforts in this direction are being addressed by others. But it would still be awkward within WDB to make the user interface that which was desired for the ACIMS, so while awaiting ORACLE interface software (PRO*C) and license verifications from Oracle, a prototype system was built to illustrate the ACIMS interface. The HTML pages were designed by Dan O'Neil.

ACIMS Prototype Investigations and Experiments

C based CGI-scripts were written to interface mSQL with the HTML pages designed by Dan O'Neil. CGI scripts are generally written in C or in Perl, so the public domain gnu C compiler²¹ was downloaded and installed on the SUN system. The HTML initial page, presented to the user, is shown in the appendix. Based on the user's selection of radio and toggle buttons an SQL query of the mSQL database is presented to the database server. The server determines but does not return the database lookup information to the client. After the initial query, the client makes the C function calls to retrieve the resulting information, inserts this information within HTML markup, and sends the results back to the web client. For the initial test, the information was returned as an HTML table as shown in the Appendix.

To experiment with adding information to the database, Dan O'Neil designed pages to support insertion into tables. A CGI program was written in C to take the information provided by the "Concepts" page and add it to the Concept table in the database. The programs that were written were experimental, and prototypical. The Oracle based production system will require new CGI scripts, since each database system has its unique interface routines. The prototype

Acknowledgements

The author would like to acknowledge the help and support of those he has met this summer. Whenever something was needed, Steve was there to figure out how we could get it. Whenever I felt as if I was catching up, Dan O was generating new ideas from Washington. Behind the scenes, Dan O, Les, and Vance are making the concept of the VRC succeed, and in the trenches are Dan O, Brandon, and Ron, who will make the idea of a distributed and collaborative engineering environment known as the Virtual Research Center work with their enthusiasm for new and experimental systems, their abilities at integrating these ideas into a Web based environment, and their commitment to put in whatever time is necessary to get the job done. Additional thanks go to others "on the hall," i.e. Drew, Dan M, Joey, who helped me with questions and installations, and the rest, too many to mention, whose friendliness, hospitality and juggling instructions made the summer fun.

REFERENCES

- 1 <http://nova.msfc.nasa.gov>
- 2 <http://www.w3.org>
- 3 <http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.HTML>
- 4 <http://www.SUN.com>
- 5 <http://www.ncsa.uiuc.edu/General/VRMS/VRMLHome.HTML>
- 6 http://home.netscape.com/assist/net_sites/HTML_extensions.HTML
- 7 <http://www.ncsa.uiuc.edu/General/Internet/www/HTMLQuickRef.HTML>
- 8 <http://hoohoo.ncsa.uiuc.edu/CGI/overview.HTML>
- 9 <ftp://emwac.ed.ac.uk/pub/https/https>
- 10 <ftp://SUNsite.unc.edu/packages/infosystems/WWW.servers>
- 11 <http://home.netscape.com>
- 12 <http://ncsa.uiuc.edu/SDG/People/jason/pub/gsql/starthere.HTML>
- 13 <http://avalon.epm.ornl.gov/Dama2>
- 14 <http://arch-http.hq.eso.org/bfrasmus/wdb/wdb.HTML>
- 15 <http://www.sybase.com>
- 16 <http://www.oracle.com>
- 17 <http://www.informix.com>
- 18 <ftp://ftp.Bond.edu.au/pub/Minerva/msql>
- 19 <ftp://prep.ai.mit.edu/pub/gnu/sparc-SUN-solaris2/gnu>
- 20 <ftp://franz.ww.TU-Berlin.DE/pub/perl>
- 21 <ftp://prep.ai.mit.edu/pub/gnu/sparc-SUN-solaris2/gnu>

APPENDIX

Database = acims

| |
|---|
| Table |
| Concepts Missions People Concept_Workforce |

Table = Concepts

| Field | Type | Length | Not Null | Key |
|--------------|------|--------|----------|-----|
| Concept_Id | int | 4 | Y | Y |
| Concept_Name | char | 50 | Y | N |
| Description | char | 128 | N | N |

Table = Missions

| Field | Type | Length | Not Null | Key |
|--------------|------|--------|----------|-----|
| Mission_Id | int | 4 | Y | Y |
| Mission_name | char | 50 | Y | N |
| Description | char | 128 | N | N |

Table = People

| Field | Type | Length | Not Null | Key |
|------------|------|--------|----------|-----|
| Contact_Id | int | 4 | Y | Y |
| Firstname | char | 50 | N | N |
| LastName | char | 50 | Y | N |

Table = Concept_Workforce

| Field | Type | Length | Not Null | Key |
|------------|------|--------|----------|-----|
| Concept_Id | int | 4 | N | N |
| Contact_Id | int | 4 | N | N |

Prototype mSQL database

Concept Query Form

Please enter qualifiers in the fields below and press the 'Search' button.

Concept ID..... :

Concept Name..... :

System Level..... :

Description..... :

Operating Principle :

History..... :

Benefits..... :

Technical Issues... :

Assumed Infra..... :

Use full-screen output even if more than one row is returned.

Return max rows.

wdb 1.3a1 - 3-Mar-1994 Send comments to bfrasmus@eso.org

Query form built by WDB

| Available Tables (Select One) | Search Refinement (Optional) |
|---|------------------------------------|
| <input checked="" type="radio"/> Concepts | <input type="radio"/> People |
| <input type="radio"/> Missions | <input type="radio"/> References |
| <input type="radio"/> Technology | <input type="radio"/> Parameters |
| <input type="radio"/> Programs | <input type="radio"/> Requirements |
| | <input type="radio"/> None |
| Keyword: <input type="text"/> | Keyword: <input type="text"/> |

Last Modified on: June 28, 1995

HTML page to query ACIMS

RESULTS

| Concept Name | Description |
|-----------------------|---|
| Mag Lifter | A tube, filled with an inert gas contains a guideway for a sled carrying a launch vehicle |
| Tether Lunar Transfer | Platforms use tethers to swing payloads from one orbit to another |

Results of the query "SELECT * FROM CONCEPTS"
generated by previous HTML page

RESULTS

| Concept Name | LastName |
|-----------------------|-----------|
| Mag Lifter | Daughtery |
| Mag Lifter | Rather |
| Tether Lunar Transfer | Anderson |
| Tether Lunar Transfer | Daughtery |
| Tether Lunar Transfer | Rather |

Results of a "join" of People and Concepts

Concepts

[Reference][Requirements][Personnel][Parameters]

Concept Name:

Description:

Operating Principles:

History:

Assumed Infrastructure:

Benefits:

Technical Issues:

| | | | | |
|---------------|--|------------------------------|---------------------------------|---------------------------------|
| Concept Level | <input checked="" type="radio"/> System of Systems | <input type="radio"/> System | <input type="radio"/> Subsystem | <input type="radio"/> Component |
|---------------|--|------------------------------|---------------------------------|---------------------------------|

| | |
|--------|------|
| Cancel | Done |
|--------|------|

July 12, 1995

HTML page to input into Concepts table.
 Page generates SQL "INSERT INTO CONCEPT (Concept_ID, ..)"

