1996

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

NEW INTERFACES TO WEB DOCUMENTS AND SERVICES

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
Division: Avionics Simulation Division, Software Division
Branches: Computer Systems Engineering,
Requirements and Technology

MSFC Colleagues: Dan O'Neil, MSFC Colleague
Ron Newby, MSFC Colleague

This reports on investigations into how to extend capabilities of the Virtual Research Center (VRC)[1] for NASA's Advanced Concepts Office. The work was performed as part of NASA's 1996 Summer Faculty Fellowship program, and involved research into and prototype development of software components that provide documents and services for the World Wide Web (WWW)[2]. The WWW has become a de-facto standard for sharing resources over the Internet, primarily because web browsers are freely available for the most common hardware platforms and their operating systems. As a consequence of the popularity of the internet, tools and techniques associated with web browsers are changing rapidly. New capabilities are offered by companies that support web browsers in order to achieve or remain a dominant participant in internet services. Because a goal of the VRC is to build an environment for NASA centers, universities, and industrial partners to share information associated with Advanced Concepts Office activities, the VRC tracks new techniques and services associated with the web in order to determine the their usefulness for distributed and collaborative engineering research activities. Most recently, Java[3] has emerged as a new tool for providing internet services. Because the major web browser providers have decided to include Java in their software, investigations into Java were conducted this summer.

The World Wide Web

The WWW functions as a client and server system. Clients request information via an address called a Uniform Resource Locator (URL). For example, the URL http:://nova.msfc.nasa.gov:80 contains the location (msfc.nasa.gov), the machine (nova) the protocol for the communication (http - HyperText Transfer Protocol), and a socket (80) to connect to for this communication. Servers are programs that listen to sockets for client requests and provide services based on the port, or on information provided by the protocol associated with the port. Although web browsers support services such as ftp (default port 21, protocol ftp), gopher (default port 70, protocol gopher), etc., the term web generally means support for the http protocol (default port 80, protocol http). Within the http protocol, a negotiation of service is carried out by adding a MIME (Multimedia Internet Mail Extension) header to the communication. For example, the MIME type "text/html" indicates that the document being communicated contains text representing the HyperText Markup Language (HTML). HTML is the most common form of communication between web browser clients and servers.

CGI Scripts

In order to support dynamic creation of documents for clients or provide access to other services such as a database, servers may act as a gateway. A script or program is invoked by the server on behalf of a client request. The program or script that is executed creates or gathers information, and passes it back to the client through the web server. The Common Gateway Interface (CGI)[4] is the most common mechanism for communicating between a program and a Web server. CGI gateway programs can be scripts or programs written in any language. Generally the gateway programs generate HTML and provide this information to the Web server, but they can return the URL of another file indicating to the browser that it should get that file. HTML provides for forms as a means to collect information to interface with a CGI script.

If the web server and the service offered to this server are running on the same platform, then the server machine is performing most of the work associated with a client's query. Java presents an alternative capability to Web clients, and allows work to be shifted from the server to the client.

Java

Java is a new programming language. Because of its simple design, Java has emerged as a possible contender to connect users and information in a new way within the World Wide Web. Most web browsers now come with the ability to download and execute Java code. Within an HTML page a definition such as

```
<APPLET href="MailApplet.class" WIDTH=400 HEIGHT=300> </APPLET>
```

specifies that at this point a program will be downloaded to be executed by the client. That program will appear in a frame on the page. Now the client is able to display the results of a general purpose program, can interact with CGI programs, or can communicate directly with ports on the server. Currently a client applet program is restricted to dealing only with the server, but this may change. For security, the client applet is also unable to read or write files from the client's system, but this may change since many browsers provide this capability when client permission is given. The remainder of this paper illustrates experiments performed using a Java browser to connect to services provided by the Web server for the VRC.

A Client Server Example

The following example is a client program that displays two text widgets for input of a radius or a sphere's surface area. Additionally there is a submit button and a reset button. If the user inputs and then submits a radius, the surface area is printed in the area window. If a surface area is submitted, then the radius is printed in the window. The client does none of the calculation except the reset, and sends the numbers to a server over a socket. The server does the calculation and returns the information to the client. In this example, a server is written in C that listens to a port for a client. The client is a Java applet. Upon a connect, the server creates a child process to handle the client, and then continues to listen to the port. The child process reads from the socket

```
readin = read(new_desc,line,MAXLINE);
```

determines the two numbers, does the calculation, and returns the result to the client.

```
sscanf(line," %f %f", &f1,&f2);
      if( f1 != 0.0 )
      f2 = 4*3.14159*f1*f1;
      else if (f2 != 0.0)
      f1=sqrt(f2/(4 * 3.14159));

sprintf(line,"%f\n%f\n", f1,f2);

write(new_desc, line, strlen(line));
```

The client is displayed from an HTML page:

```
<HTML>
<HEAD><TITLE> Surface Area of a Sphere </TITLE></HEAD>
<BODY>
<APPLET CODE="Surface3.class" HEIGHT=100 WIDTH=200></APPLET>
</BODY></HTML>
```

The code concerned with the socket is shown below.

```
public static final int PORT = 4000;

public boolean action( Event evt, Object arg) {
    if (arg.equals("Submit")) {
          // now try to get a socket
     try {
       soc = new Socket("nova.msfc.nasa.gov",PORT);
       in = new DataInputStream(soc.getInputStream());
       out = new PrintStream(soc.getOutputStream()); }
     catch(IOException e)  {
          System.out.println(e.toString());
          System.exit(0); }
     out.println(r.getText());
     out.println(s.getText());
     try{
          r.setText(in.readLine());
          s.setText(in.readLine());  }
     catch(IOException e)  {
          System.out.println(e.toString());
          System.exit(0);  } }
```

A Java to CGI example

The VRC has many CGI scripts that perform as gateways to other services such as database interaction. It would be nice if there was a way for an applet to use the CGI mechanism that is already present in browsers. Unfortunately the Java library does not currently provide such a communication path. A CGI script can be invoked manually, the same as a Web browser,

```
Socket s = new Socket("nova.msfc.nasa.gov",80);
```

and deal with a cgi program through the server. But without a way to pass return information to the browser, cgi scripts that return html documents are of little use to a Java program. However Java clients will need to use CGI gateways as long as the clients are restricted to communicating with only the server machine.

An example was written to illustrate how Java applets may communicate with a VRC database via a server. The client runs as a Java applet on any system. The server runs on the VRC test web server (nova.msfc.nasa.gov), and the database is on a VRC database architecture (astrionics.msfc.nasa.gov). The Java client code is found in the appendix, but that part that

illustrates opening of the socket and manipulation of the two text windows (s is the name for the input of SQL commands, the window v is for output of the database query) is given below. Notice that the client opens a connection to the Web server, and provides a path to the CGI script that is running on the server (in this case the program /cgi-bin/sqlquery). The CGI script is a compiled C program that makes a database query and returns the text as output to the Java client.

```
public void submit(String sdata)
  {  String home = "nova.msfc.nasa.gov";
     String script = "/cgi-bin/sqlquery";
     int port = 80;
     Socket s = null;
     rdata = "";
     try
     {  s = new Socket(home, port);
        DataOutputStream os
           = new DataOutputStream(s.getOutputStream());
        DataInputStream is
           = new DataInputStream(s.getInputStream());
        os.writeBytes("POST " + script + " HTTP/1.0\r\n"
           + "Content-type: text/plain\r\n" + "Content-length: "
           + sdata.length() + "\r\n\r\n");
        os.writeBytes(sdata);
        String line;
        while ((line = is.readLine()) != null)
        rdata += line + "\n";
        v.setText(rdata);
        os.close();
        is.close(); }
```

Other New Browser Capabilities

For security reasons, browsers have not generally provided a way for servers to access the disk of the client machine. The most recent versions of Netscape have however included this capability as an experiment. It is still open as to whether this will become a standard part of HTML. In order to experiment with this capability, Perl5 was installed on the test VRC server, and Perl libraries that interface with Oracle databases and provide CGI capabilities were installed. A script was written that provides a user the capability to name a file and upload this file to the server. The script dynamically creates the HTML page for the user, and the critical portion of this code is given below:

```
print '<b> For uploading a file, enter a fully pathed file ',
      '<p> Examples:<ul><li>/home/my/filename - in UNIX
               <li> c:\path\to\myfile - in DOS
               <li> HD 40: Desktop Folder: This File - for Macs </ul>',
     ' <p>OR press Browse to browse your file system.<b>';
print $query->start_multipart_form("POST",
                  "/cgi-bin/carlisle/savefile",$CGI::MULTIPART);
print $query->filefield(-name=>'uploaded_file', -default => 'ignored',
                        -size=>50, -maxlength=>80);
```

```
print $query->hidden(-name=>'dirname',-default=> ["$root$file"]);
print '<br>';
print $query->submit(-name=>"Submit", -value=>'Submit this File');
print $query->endform;
```

Upon submission of this form, the cgi-script "savefile" is called. The Perl code for this script is:

```
#!/usr/bin/perl
use CGI;
$query = new CGI;
$filename = $query->param('uploaded_file');
print $query->header;
$dirname= $query->param('dirname');
# this is a hack that I am sure will fail for some systems and
#some file names
@tmp = split(/[\\\:\/]/,$filename);
$filename = pop @tmp;
$outfile = "$dirname/$filename";
#to actually write the file comment out the next line
#and uncomment the open and following while
print $dirname,'/',$filename;
#while(<$filename>) { print; }
#open(fp,">$outfile") || die("You do not have write permission at $dirname");
#while($bytesread=read($filename,$buffer,1024)) {
#    print fp $buffer; }
print '<p>';
print "File saved to $outfile"
```

Notice that the actual writing of the file has been commented out. This was done because currently this script has not been installed in a secure area on the server. Thus anyone would be able to upload files to the VRC, and a malicious or careless user could fill up and crash the disk of the server machine. This illustrates why the capability to upload information of unknown size to a server is a dangerous practice.

CONCLUSIONS AND ACKNOWLEDGEMENTS

Future work will continue to address the rapidly changing environment known as the Web, and will incorporate those features into the laboratory that create an environment for collaborative research at distributed locations.

The author would like to acknowledge those at NASA who have made this summer such an enjoyable experience.

---

1 http://nova.msfc.nasa.gov

2 http://www.w3.org

3 http://java.sun.com

4 http://hoohoo.ncsa.uiuc.edu/CGI/overview.html