Introduction to Web Technologies and Their Applications

Geoffrey C. Fox *

May 20, 1997

Abstract

This paper discusses basic Web technologies, PERL, Java, JavaScript, VRML, databases (web-linked), and digital multimedia in contact of a layered WebWindows software architecture in which applications are built on top of multi-use services. We assume a community technology approach to both hardware and software. Perhaps the hardware model will be ATM connected PC's while the community software approach will not involve applications written for Windows NT/95 or UNIX, but rather for WebWindows, with interfaces defined by the standards of Web servers and clients. This universal environment will support WebTop productivity tools, such as WebWord, built in modular dynamic fashion, and will undermine the business model for large software companies. We illustrate these ideas with some examples including business enterprise systems (IntraNets), health care, financial services, command and control, manufacturing, and education.

1 Introduction

This article is meant to introduce and illustrate the use of six central Web technologies, which are described in Section 3. We assume the continuing trend to pervasive computers and networks implementing the National Information Infrastructure (NII) as the melding of today's Web, consumer PC, computer and communication infrastructure. Perhaps this will embody Gordon Bell's SNAP concept (Scalable Networks And Platforms), which suggests universal commodity PC and ATM network hardware with probably a growing use of Windows NT as the dominant operating system [Bell:95a]. In Section 2.1, we describe the classic Web client server architecture and Section 2.2 describes the standard turn of the century ($\sim$ 2000+) NII scenario. Section 2.3 is devoted to WebWindows, which describes the new universal API (application program interface) offered by the world viewed as a collection of Web servers and clients. We believe WebWindows is so much productive and powerful than conventional software interfaces that it will become the dominant and hardware independent "operating system" [Fox:96c]. In the final part of Section 2, we discuss NII applications and services briefly to show how they are built in terms of Web technologies and the WebWindows interfaces.

Section 3 discusses and illustrates our six chosen technologies—PERL, Java, JavaScript, VRML, web-linked databases, and digital multimedia.

Finally, in Section 4, we describe the role of the six particular applications—IntraNets and Business Enterprise Systems, Healthcare, the Financial Industry, Education, Manufacturing,

*Syracuse University, Northeast Parallel Architectures Center, 111 College Place, Syracuse, New York 13244, gcf@npac.syr.edu, http://www.npac.syr.edu

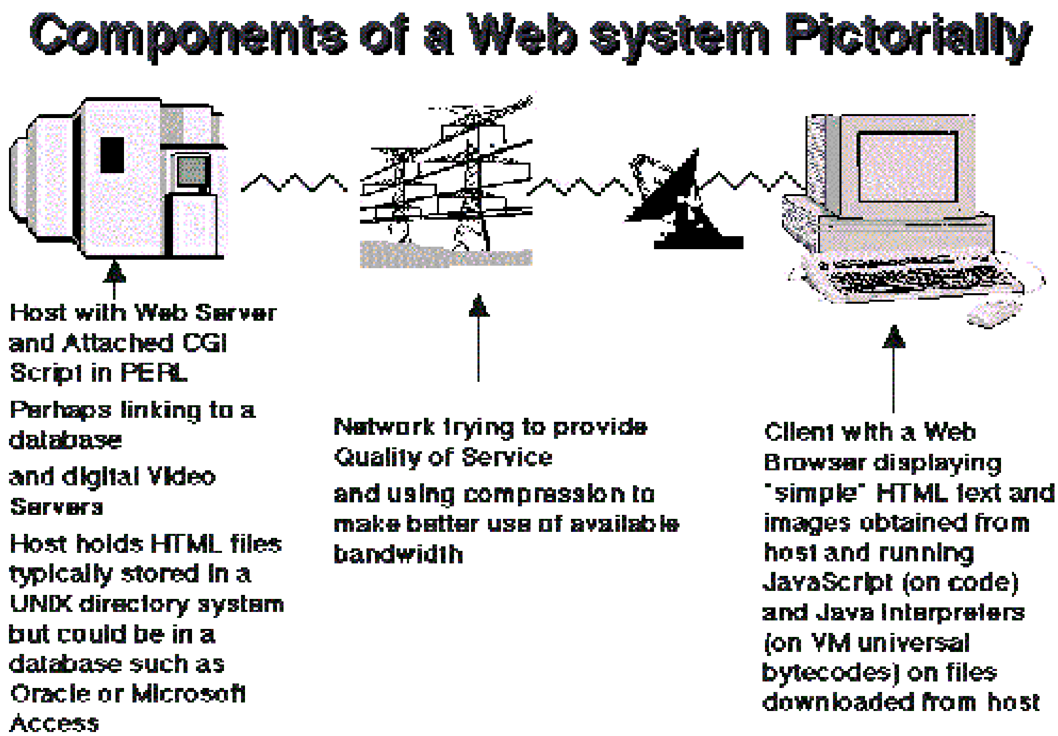


Figure 1: Synergy of InterNet and IntraNets

and finally, Crisis Management, or command and control. The last three applications are described in greater detail than the first three.

Already, the Web links some 400,000 Web servers together [IDS:97a] but technologies, such as Java and a growing use of interactive and collaborative applications will either directly (run a server on client machine) or indirectly (enhanced browsers) turn all clients into servers. Furthermore, JavaVM (Virtual Machine) supports client side multithreading, and so differences between (multitasking, e.g., UNIX) servers, and (multithreading, e.g., JavaVM) client technologies are, indeed, fading away.

2 Framework for the Web and National Information Infrastructure

2.1 Client-Server Computing

Figure 1 shows a classic Web set up indicating it is "just" a client server application, and its implementation requires appropriate client, server, and network technologies. We will learn mainly about the client and server software issues in this paper, and only describe enough about network hardware and protocols to put the other issues in contact. The original Web architecture, shown in Figure 2 was comparatively simple with clear functionality in each component. However, as the Web evolves, we move to the richer confusing scenario of Figure 3 with many overlapping concepts. Actually, the natural Web model is probably a server-server architec-

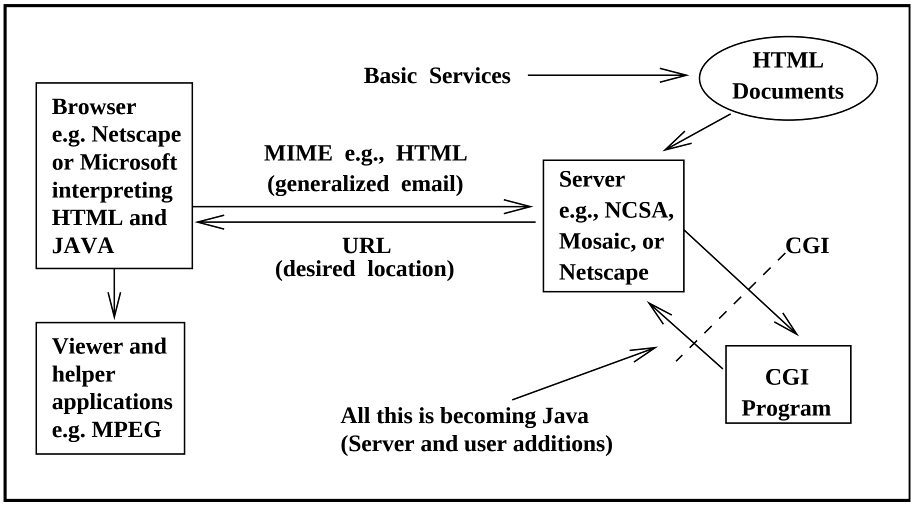


Figure 2: Simple representation of "old" (pre-1996) WWW software architecture showing emerging advanced services, as well as basic technology

ture, as all machines need capabilities embodied in servers. This fits with Vice President Gore's view of the NII where everybody is both a consumer and producer of information. We can, of course, implement this (crudely) already by running a Web server and Web client on every machine (see Figure 4(b)), as well as the distributed environment, Figure 4(a). The growing functionality in clients is natural because, although individual servers are typically much more powerful than individual clients, the total compute power (and dollar investment) is probably an order of magnitude for clients than servers (Figure 5).

2.2 The National Information Infrastructure (NII)

Gordon Bell's SNAP vision [Bell:95a] recognizes that the World Wide Web (NII), parallel supercomputers, and your organization's enterprise network will be built from the same pervasive communication infrastructure—twisted pair, coaxial cable, optical fiber, satellites and cellular phones connecting everybody and every institution together [Fox:97a]. Of course, different parts of this hardware will perform with different speeds and will be upgraded at different times in different parts of the country. However, a similar software base, Web technology, will dominate applications on this diverse infrastructure because it is built for the largest market, and therefore, will best leverage the software development effort into lower costs and higher functionality. Obviously, we now have an imperfect realization of this SNAP vision today, but there is enough infrastructure in pace to create the opportunity now being explored and created by America Online, Netscape, Microsoft, WebTV, and others.

In Figure 5, we estimate the compute and communication capability of the NII and find it 100 to 1,000 times that of conventional supercomputers. However, we expect large-scale parallel processors to be used not only in classic supercomputers, but also in the network of video servers, and more generally, WebServers that supply information to these clients. We term this scenario *InfoVISiON* for *Info*rmation, *V*ideo, *I*magery, and *Si*mulation *ON* Demand.

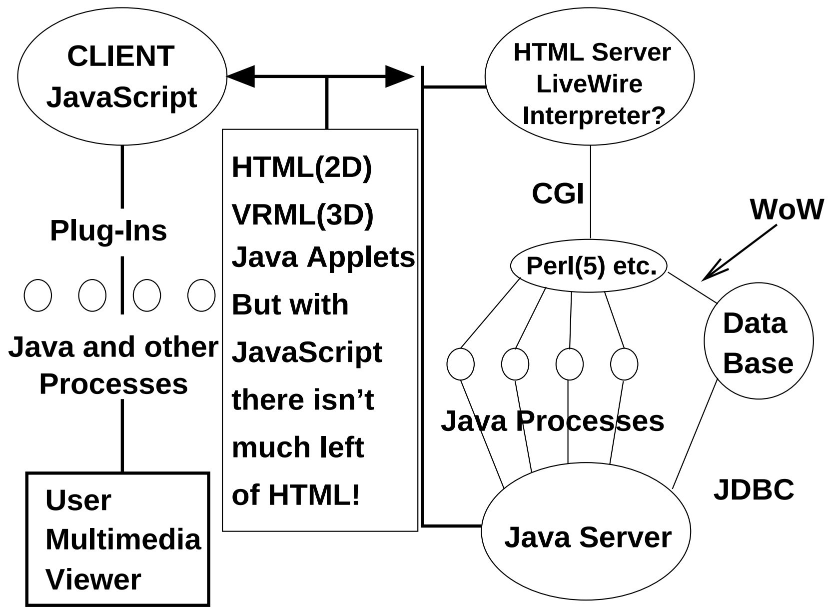


Figure 3: 1996 Java/Netscape client-server model with evolving/confusing/overlapping capabilities

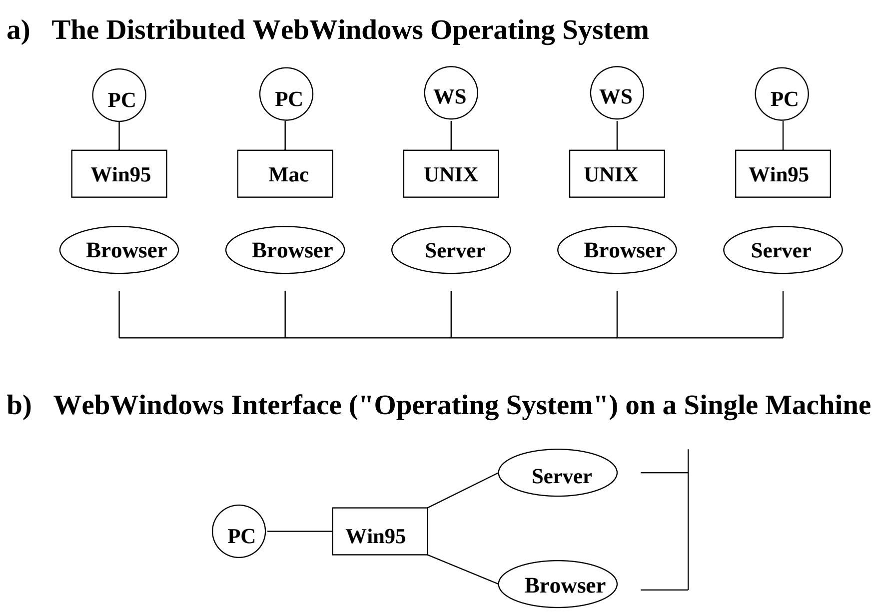


Figure 4: WebWindows for (a) the World or (b) one PC

NII Compute & Communications Capability in Year 2005 - 2020

100 Supercomputers at a Teraflop each	10^{14} (Fl)ops/sec at 100% Duty Cycle
100 Million NII Offramps or Connections at realtime video speeds	10^{14} bits to words/sec at about 10% to 30% Duty Cycle
100 Million home PCs, Videogames or Settop Boxes at 100-1000 Mega(Fl)ops each	10^{16} to 10^{17} (Fl)ops/sec at about 10% to 30% Duty Cycle
1,000 to 10,000 High Performance Multimedia (parallel) servers each with some 1,000 to 10,000 nodes	10^{15} to 10^{16} ops/sec at 100% Duty Cycle

Each of three components (network connections, clients, servers) has capital value of order $10 to $100 billion

Figure 5: An estimate of the communication bandwidth and compute capability contained in the NII and supercomputer industries

InfoVISiON includes the storage, query, and dissemination of this wide range of multimedia data. There is surely at least 100,000 hours of interesting video material in the archives of Hollywood studios, CNN, Routers, and network TV. If compressed in MPEG format, this corresponds to some 100 Terabytes of needed storage capacity. MPEG2 or other formats, such as motion JPEG, would require much more storage and are probably necessary to support editing and other video production applications. Note that the current WWW has only a small percentage of its storage devoted to video, whereas the future NII will presumably be dominated by video data.

Each NII offramp will, as shown in Figure 6, connect homes (offices, school desks) at the rate of 1–20 Megabits/sec to the set of NII InfoVISiON servers. This rate covers the range from MPEG compressed VHS to HDTV picture quality. Note that this performance is 100–1,000 times greater than today's conventional 28.8 Kbaud modem on a twisted pair (plain old telephone service POTS) connection.

Returning to Figure 5, we estimate that with an investment in large servers that is 10% of that in clients gives an InfoVISiON or WebServer market that is at least an order of magnitude larger than that for traditional supercomputers. This illustrates that it is the NII, and not large-scale number crunching, that is the best opportunity for (high-end) parallel processing. Notice also that the compute power contained in the year 2000 NII is some 10–100 PetaFLOPS—far larger than the compute capability of an individual TeraFLOPS supercomputer. This is explored in more detail in [Bhatia:97a], [Fox:97a], [Fox:97c] which discuss extending the Web to compute services, and harnessing the power of the World Wide Metacomputer.

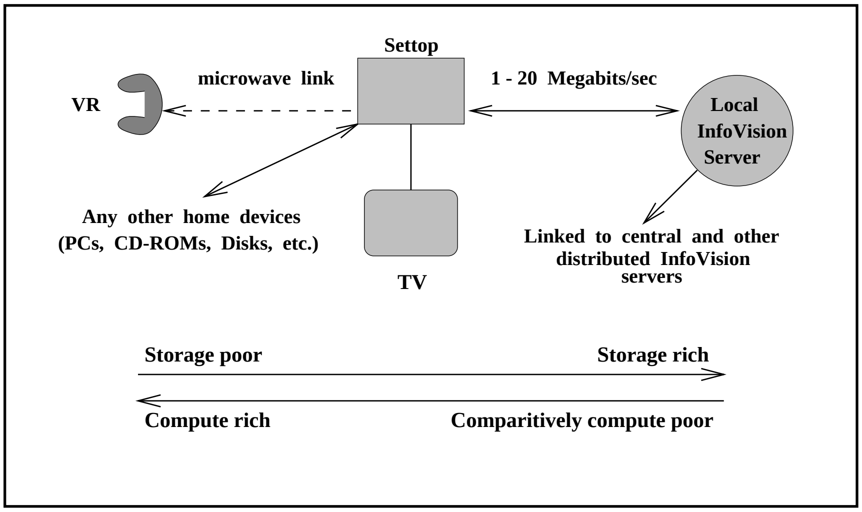


Figure 6: Basic InfoVISiON home scenario in the year 2000: Intelligent settop box interfacing the digital home to a hierarchical network of InfoVISiON servers

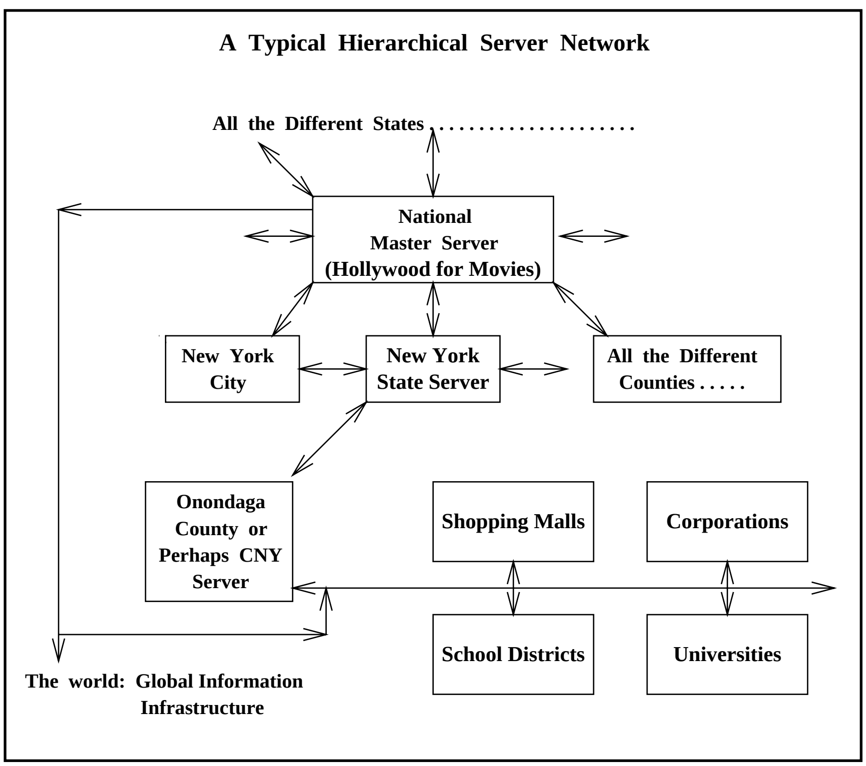


Figure 7: Typical hierarchical server network for Hollywood master system cascading down through a fragment of node systems shown for central New York

One could view this InfoVISiON scenario as the most amazing client-server application with 10^8 client/small servers and 10^4 large servers, but as mentioned in Section 2.1, we could instead implement this as a heterogeneous server-server architecture with 10^8 distributed nodes. This can be viewed as a fascinating parallel computer with many more nodes than traditional tightly coupled systems (by a factor of 10^5 compared to typical large 1,000 node parallel processor). It appears that although powerful, the communication backbone of Figure 5 will not allow every "client" long distance simultaneous access to every other client or server. Rather, we must enforce the guiding principle of all computer architectures—namely, data locality. This is illustrated in Figure 7. When Jurassic Park VI is released on the Hollywood Server, one will not have everybody accessing it there at an average distance of some 1,500 miles. Rather, this "hot" movie will be cascaded down (cached) through the hierarchy of servers so that any individual will find it on a server a few miles away. This strategy reduces the needed fiber for the NII trunks by a factor of about 100.

2.3 WebWindows

Figures 2 and 3 illustrate the overall architecture of Web software linking multimedia servers and clients with standard interfaces and technologies, such as HTML, VRML, and Java—some key concepts are defined in Table 1.

Table 1: Some WorldWide Web Technologies and Their Role

Clients (such as Mosaic and Netscape)	support browsing of hyperlinked documents, but have no internal interactive/compute capability [Andreessen:93a]
Servers	read HTTP and deliver requested service to client
HTML	document format supporting hyperlinks
HTTP	transport protocol defining interaction between Web servers and clients
URL	universal resource locator or addressing scheme for items on the Web
MIME	data format allowing agent-like (extended e-mail) communication
CGI	standard Web Server interface allowing sophisticated server extensions
PERL	rapid prototyping language (Script) aimed at text and file manipulation (systems programming)
Java	semi-interpreted C++ like language supporting an applet model of distributed computing
JavaScript	interpreted language for manipulating Web components in documents
VRML 1.0	universal description for static three-dimensional objects
VRML 2.0	dynamic 3D descriptor for virtual environments
JDBC	set of methods and drivers linking Java to general relational databases (Java Database Connectivity)

We believe that the current trend to build applications for this architecture will accelerate. Today, we already see WebMail, WebDatabase (see Figure 8), WebEditor, WebFoil, WebChat, but soon we see a set of more advanced products, such as "WebWord," which will offer the functionality (and more) of Microsoft Word (and similar products) but be built with Web technologies. WebWord is quite different (in architecture and hence implementation) from linking Word to the Web (Weberizing Word as in today's useful intermediate tool, Microsoft's Internet Assistant) and so Microsoft has no special advantage (other than a lot of money and resources) in this emerging WebTop Productivity field. WebWord will use Web standards for all internal and external representations, and so allow easy integration of new functionality, and customization of WebTop environments for particular markets. The Web will not only allow new technologies to develop better versions of current applications (such as word processors) but also produce new uses of computers.

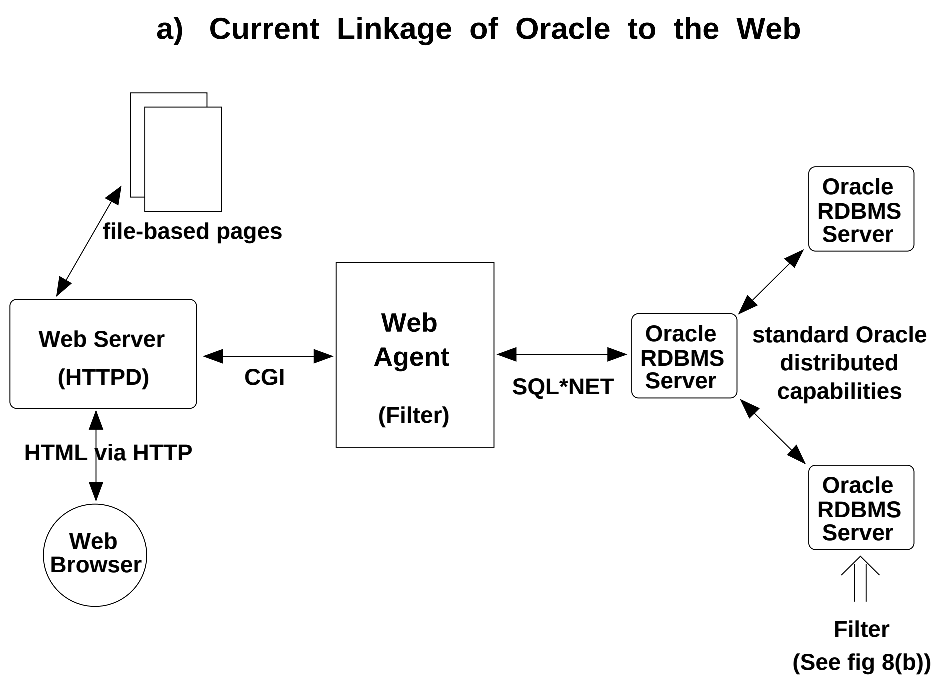
Remember that when we talk of products using Web technology, it does not have to be applied to the full world-wide Internet. Rather, WebWord could run a single PC which has both server and client capabilities (Figure 4(b)). Again, Web linked relational or object databases (Figure 8 and Section 3.5), and a future WebLotusNotes product could use Web Technology and Webserver systems to support a Business Enterprise Information system or IntraNets on a closed corporate network.

The Web naturally supports virtual organizations with several small entrepreneurial companies linked dynamically together. It will be hard for the large monolithic systems and software companies to be competitive. There will, however, be an interesting systems integration business, which will put together and support linked sets of Web technology modules for particular services in targeted markets and organizations.

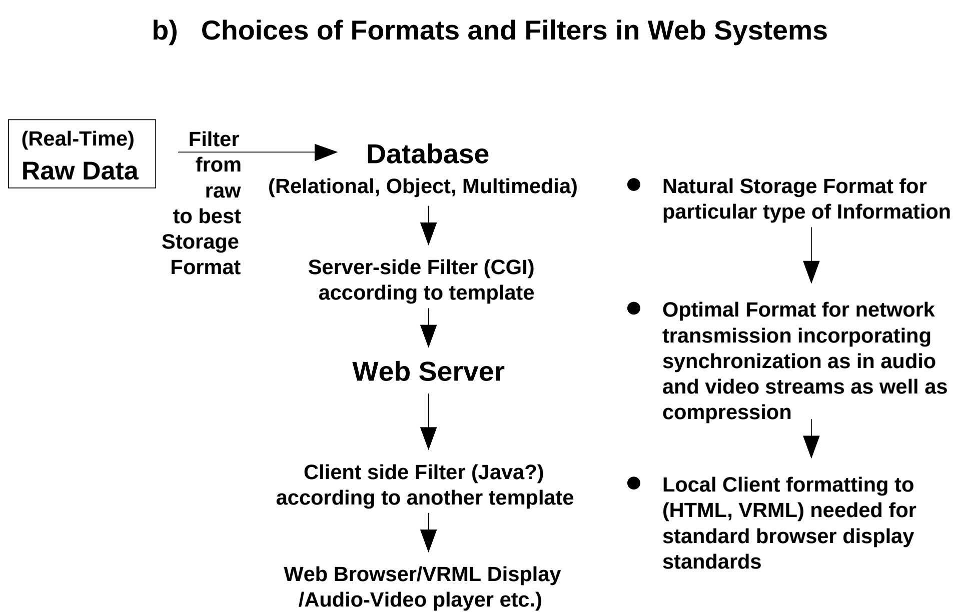
The above implicitly defines WebWindows as the target architecture for the future replacing UNIX, Apple Macintosh or the Win32 interface to Windows 95/NT. Bell argues that Windows NT will be the operating system of SNAP; we say that this may be correct but perhaps irrelevant as the real "action" will be at the higher level defined by the Web [Fox:96c]. This leverage of the highly functional high volume Web technology to particular applications on particular networks is shown in Figure 9. Note that some applications will require enhancements in the services provided by base Web technology. Security is a good example of a service that is a critical requirement for sectors, such as commerce and defense. In fact, the role and structure of operating systems could change for they need not support users directly, but just Web servers and clients. This changing operating system architecture is shown in the comparison of Figure 10 and Figure 11 where the latter also illustrates the blurring of the distinction between (Web) servers and clients.

2.4 Generic Services on the NII

We can define a simple layered architecture for Web (NII) applications which are built in terms of multi-use services as shown in Figure 13. "Multi-use" extends the well known dual-use civilian-military interplay to a set of capabilities shared by many different applications. Note the Web is an excellent implementation technology for the COTS (customer off the shelf) choice used in many new defense software systems. There is no precise definition of services and their difference from applications, for services are essentially generic applications, and most applications are complex metaproblems built recursively from services and "sub-applications"



(a)



(b)

Figure 8: (a) Oracle-Web Integration Architecture; (b) Choices of Formats and Filters in Web Systems

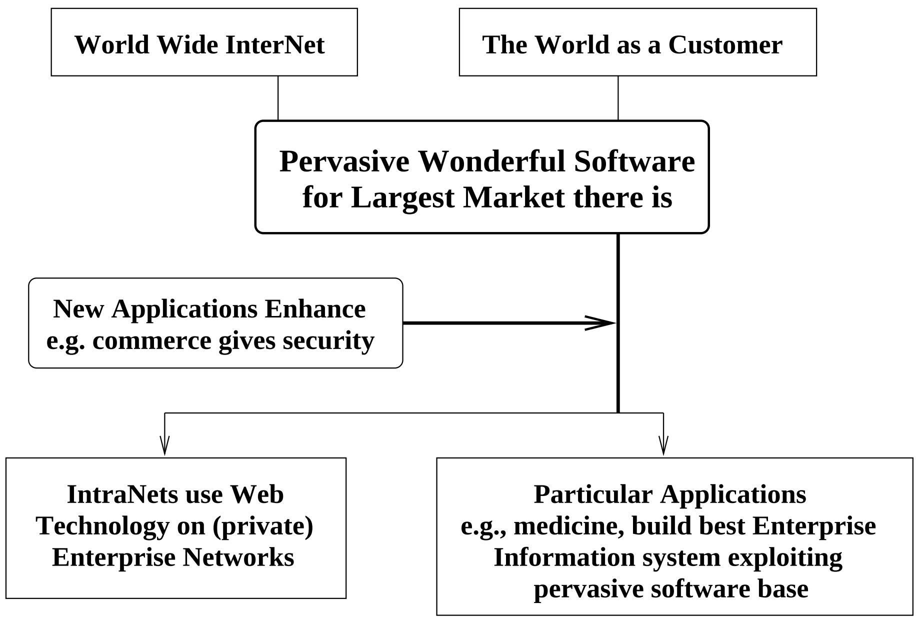


Figure 9: This illustrates how economies of scale combined with the richness of the underlying distributed computing model imply that Web technologies with well-chosen enhancements can be used for IntraNet applications.

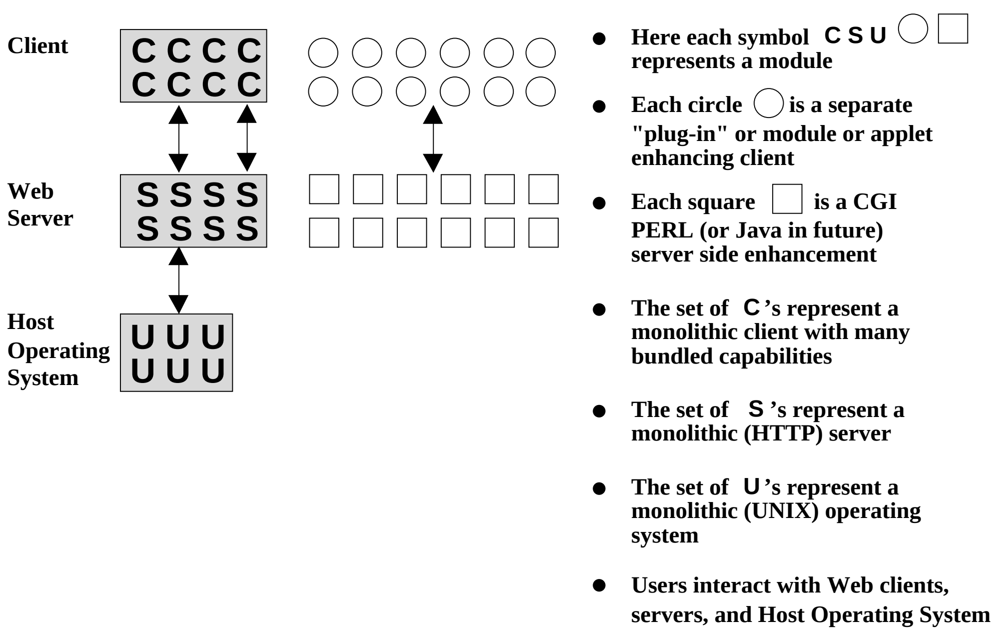


Figure 10: Architecture of Web Client-Server Software—Netscape and Internet Explorer Today

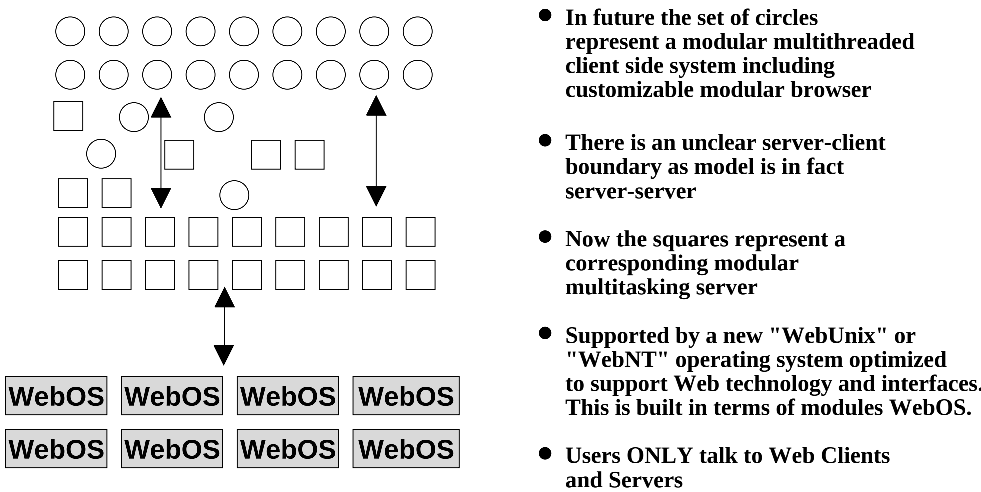


Figure 11: Architecture of Web Client-Server Software—The Future as suggested by Hotjava and Java Servers, such as Jeeves and Jigsaw? [Bhatia:97a]

[Fox:95c]. Table 2 maps services onto seven application areas.

Thus, there is a grey fuzzy line distinguishing services and applications. Five possible NII services include:

- **WebTop Services**—Publishing, Productivity, Software Engineering

  Here, we put base productivity tools including "WebWord," "WebLotus123," Web linkage to relationship databases, etc. Essentially, all applications use this service.

- **InfoVISiON**—Information, Video, Imagery, Simulation, ON demand

  As described in Section 2.2, this includes base database storage, management, query, and dissemination of the full range of multimedia archives of the World's distributed digital libraries. Note delivery of results of a simulation—such as access on demand to a weather model—is included in this service.

  Considering the two examples from Section 4, in manufacturing, InfoVISiON corresponds to delivery of data in a configuration controlled database. In command and control, this service allows commander to index and access videos of battle damage videos or real-time engagements.

- **Electronic Funds and Security**—Digital Cash, Security, Authentication, etc.

  This collection of services enables electronic commerce, including on-line banking and shopping. These services are also essential for the use of the WWW for processing and exchange of proprietary (manufacturing), and classified (military) data.

- **Collaboration**—Real-Time Interactive and "Batch"

  This includes desktop video conferencing, three-dimensional graphics MOOs, geographically distributed CAVEs leading to full televirtual interactions. Java-based client-server systems, such as Tango [Beca:97a] and Habanero [NCSA:97b] are very promising. The

Table 2: Services used by seven NII applications

Applications	**WebTop Productivity**	**Info VISiON**	**Commerce/ Security**	**Collab-oration**	**Meta Computing**
Society	X	X	X	X	
Education	X	X		X	
Enterprise Systems	X	X	X	X	
Health Care	X	X	X	X	
Command and Control/ Crisis Management	X	X	X	X	X
Manufacturing	X	X	X	X	X
Collaboratory	X	X		X	

emerging VRML 2.0 standards will be very important in building virtual environments. As discussed earlier, a wide variety of other types of interactive information exchange is necessary. This underlies the concepts of collaboratories (virtual research groups or scientific laboratories), and the virtual company of the next century's agile manufacturing environment. In the more static mode, we see workflow and configuration control, which allows tightly integrated projects, such as those needed to build a complex system including an aircraft (see Section 4.4) or a large software module with a distributed team.

- **Metacomputing**—A worldwide collection of computers organized together as a single computational engine for simulation or information processing [Fox:95a], [Fox:97a].

  This service can be used to control remote medical and scientific instruments; search the world for information; simulate the weather expected in a military engagement, or link computers in different companies for a multi-disciplinary optimization of a new vehicle.

In several articles ([Beca:97a], [Bhatia:97a], [Fox:97a]), we have explored the use of Web software as the basis of the operating and programming environments of parallel processors. In the pyramid view of computing (Figure 12), this uses the philosophy of Figure 9 to extrapolate software from the base to the top of the pyramid. This implies Web hardware (the collection of clients from Figure 5) could provide the world's fastest computer while Web software will run the largest tightly coupled parallel computers.

Some services listed above can be already prototyped in terms of today's Web technologies. For example, base WebTop or early Collaboration services are now becoming available. Some other services are still waiting for their pervasive enabling technologies, such as the physical infrastructure that will enable InfoVISiON or security that will enable Internet Commerce. Finally, the computationally extensive NII services, characterized above broadly as "Metacomputing" require a major extension of the whole Web paradigm, currently still focused on image and document services, but already gradually expanding towards computation and interactive simulation via technologies such as Java and VRML.

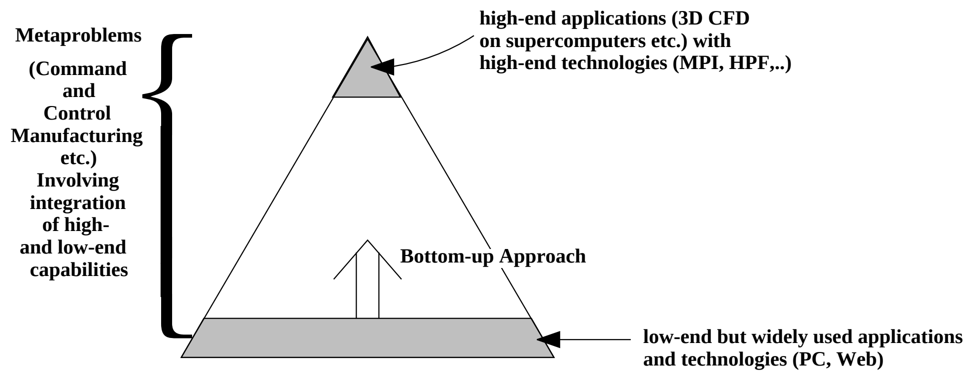


Figure 12: Integration of Large Scale Computing and Web Technologies

3 Some Specific Web Technologies

In this section, we discuss some of the specific technologies shown in Table 1 and Figures 2 and 3. We give simple examples of their use, and for languages, their syntax. We cover, in the following sub-sections, PERL, Java, JavaScript, VRML, Web-linked databases, and digital video (multimedia).

3.1 PERL4 and PERL5—CGI Scripts

PERL is a very well-know interpreted language that was developed (before the Web) as a powerful systems programming language. Although aimed at UNIX, PERL is also available on PC Windows platforms.

PERL is commonly used in so-called CGI Scripts that conveniently extend the capability of servers, as shown in Figures 2 and 3. In theory, one can use any language in a CGI Script, including higher-level shell Scripts or high-performance languages, such as C or C++. In fact, as shown in Figure 3, one can expect Java to be increasing used for server side extensions for both CGI and more sophisticated mechanisms. However, PERL currently has the most powerful text processing and (UNIX) file system handling capabilities of any common language, and is clearly currently superior to Java and JavaScript in this regard. The most common use of CGI Scripts is processing of HTML form input (see Figure 13) and this is dominated by text and file manipulation. Further, such work is often not compute intensive, and a flexible interpreted implementation has adequate performance. Thus, PERL remains the language of choice for CGI programs. Further, many Web applications require substantial use of (off-line) utilities to reformat text, and prepare information. Here again, PERL currently is the best implementation language. Note that hybrid approaches for CGI Scripts are quite common with a PERL interpreted "wrapper" invoking special applications, such as database engine or an optimized C or Fortran numerical simulation.

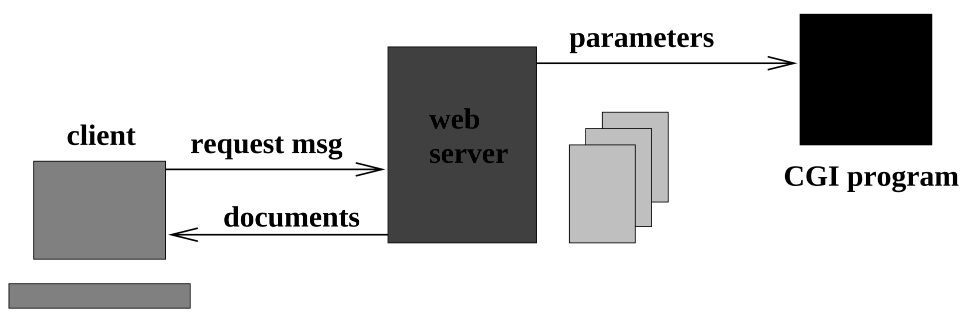
When we describe JavaScript, it provides an alternative way of processing HTML forms solely on the client side. However, this is only appropriate for some Web applications, and many require server side processing either due to the complexity of the computation, or the need to access information stored on the server file systems. Thus, PERL and JavaScript offer

a) Request
The client sends a request, conforming to the URL standard and formatted with a MIME header, to the server. The server parses the request and decides what to do:

–for FTP and other services, the server makes an appropriate request of its operating system and responds,

–for HTTP service, it retrieves the file named by the URL and decides what to do based on file type. An HTML, MPEG, AU, or any other file with recognizable file extensions is returned directly to the client with no further processing (except in the case of Server Side Includes—SSI),

–if the file is executable, the server executes it as a CGI program. The server processes the header to pass execution parameters as environment variables or as a STDIN stream to the CGI program.



b) Answer
The CGI program parses the input from the server and MUST generate a response—even if there is no data to send back, the CGI program must send an error or empty message since the http connection is still open and must be closed by the server. The CGI program will send a header to the server:

–If the header is type "Location," the server will send the indicated file to the client.

–If the header is "Content-type," the server will send all the data back to the client. This should be a properly formatted HTML page.

When the CGI program terminates, the server closes the connection.

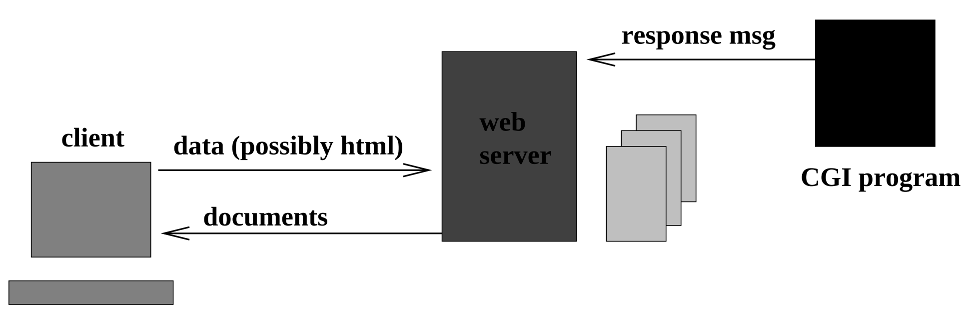


Figure 13:

```
#!/usr/local/bin/perl5 -w
#  Simple non CGI Hello World Example
#
print "what is your name?"; # Prompt User
$hero=<STDIN>; # Read Name Input
chomp ($hero); # Remove Trailing Newline
#
$today =`date`;
chomp($today); # Today's date captured from UNIX
#
print "Hello World to you my friend $hero on a great $today\n";
exit; # End Program
```

Figure 14: Simple PERL Program to read in name input and output a Cheery Greeting

complementary capabilities.

Java is already the preferred implementation of customized servers, such as those described in Section 4 for supporting collaboration. However, until Java has the same convenient text processing and flexible interpreted implementation, PERL will continue to be used extensively for CGI Scripts.

The most recent version, PERL5, has important extensions—especially in the area of multi-dimensional arrays and object-oriented features. However, although the new arrays features are very helpful, we find the object-oriented implementation much less elegant than Java, and it has not been broadly adopted.

In Figure 14, we show a simple "Hello World" program in PERL, showing the use of so called variable interpolation in character strings. In Figure 15, a HTML page allows the user to input a name that generates a "Hello World" HTML page using the PERL CGI Script in Figure 16.

3.2 Java and Its Use on the Web

Java is an object-oriented language that has similarities to C and C++. Its introduction, in 1995, had a major impact on the Web, as it allowed one to incorporate customized computer programs on the client (Web browser) side, as shown in the complicated Figure 3. Thus, Java, for the first time, allowed one to build balanced client-server applications. Note, Figure 5 points out that there is substantially more compute capability on clients than servers. Further, any server-based system is likely to have nontrivial latency (delay in response) and so Java can be used to build user interfaces where many relatively simple computations are involved, and the low-latency client-side approach is clearly preferable. However, Java is turning out to be of even greater importance than these Web architecture issues suggest. In particular, one expects it to challenge C++ and Fortran ([Fox:97e]) as the future language of choice for software engineering and scientific computation, respectively. This trend is due to a variety of reasons. For instance, the WebWindows approach suggest future large software engineering projects will build to Web standards, which implies Java will often be a natural implementation language. Again, Java offers an attractive object-oriented environment with the potential of optimizing Java compilers

(a)

```
<head>
<title>My Hello World CGI Script Tester</title>
</head>
<body>
<h1>Enter Your name in Text field of Form Please</h1>
<FORM ACTION="http://boss.npac.syr.edu:8080/cgi-bin/webwisdom/helloworld.pl" >
My name is: <input name="username" type="text" size="40">
</form>
</body>
```

(b)

Enter Your name in Text field of Form Please

My name is:

(c)

Hello World to you my friend Geoffrey Fox on a great Sun Mar 23 14:21:55 EST 1997

Figure 15: HTML Page defining Hello World with (a) HTML displaying as shown in (b). The result is displayed in (c).

being able to generate comparable performance to Fortran. Further, we can expect a growing trend to learn Java as a first language, as children are exposed to it from Web browsing and find using Java to enhance their home pages as an inducement to learn programming at school. These positive indicators are tempered by the immaturity of technology, which is only to be expected, but implies today (1997) that one should use traditional languages to build robust high-performance languages.

Java is used on the Web in a semi-interpreted mode shown in Figure 17. Here, the *javac* compiler converts Java language (java) into JavaVM (class files), which can be thought of as universal machine code. The JavaVM files and libraries are downloaded as needed from the server to client. At the client, the “bytecodes” of the JavaVM are interpreted, checked for security issues, and converted to native machine code.

Java comes with a powerful runtime that is aimed in particular at the graphics, networking, and user interface features that one needs for interactive Web pages. There is, in particular, a comprehensive AWT (abstract windowing toolkit) that supports such things as buttons, forms, windows, and their layout.

In Figures 18 and 19, we give, respectively, the Java source, and invoking HTML page that will display “Hello World!” Note this uses built-in (runtime) classes, **Applet** and **Graphics**.

We can expect Java technology to improve in many ways that will increase its performance and functionality. We will see better “Just in Time” compilers—essentially, clever interpreters of JavaVM that generate reusable compiled code on the fly. We will see a wide variety of new libraries (runtime) supporting areas as diverse as commerce and scientific computing. Again, as

```
#!/usr/local/bin/perl5 -w
#  Simple Hello World Example
#
require "cgi-lib.pl"; # Public Domain library to read form data
#
$statusofread =  &ReadParse(*formdata); # read data in associative array formdata
#
$hero="Anonymous"; # if no good form data!
if( $statusofread )     { # If $statusofread nonzero, data read OK
        $hero = $formdata{"username"}; # The input user name
}
#
$today =`date`;
chomp($today); # Today's date captured from UNIX
#
print <<EOF; # Generate the HTML responding page starting with HTTP line
Content-Type: text/html

<html>
<head>
<title>This is Result of my Hello world CGI Script </title>
</head>
<body color="white" >
<h1>Hello World to you my friend $hero on a great $today</h1>
</basefont>
</body>
</html>
EOF
exit; # End Program
```

Figure 16: CGI Script used in previous example

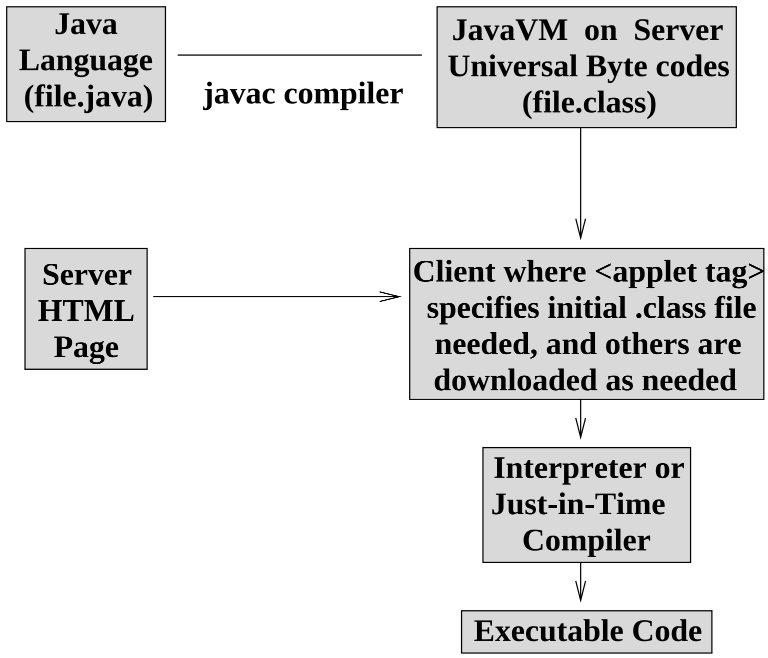


Figure 17: Structure of Interpreted Java Applets

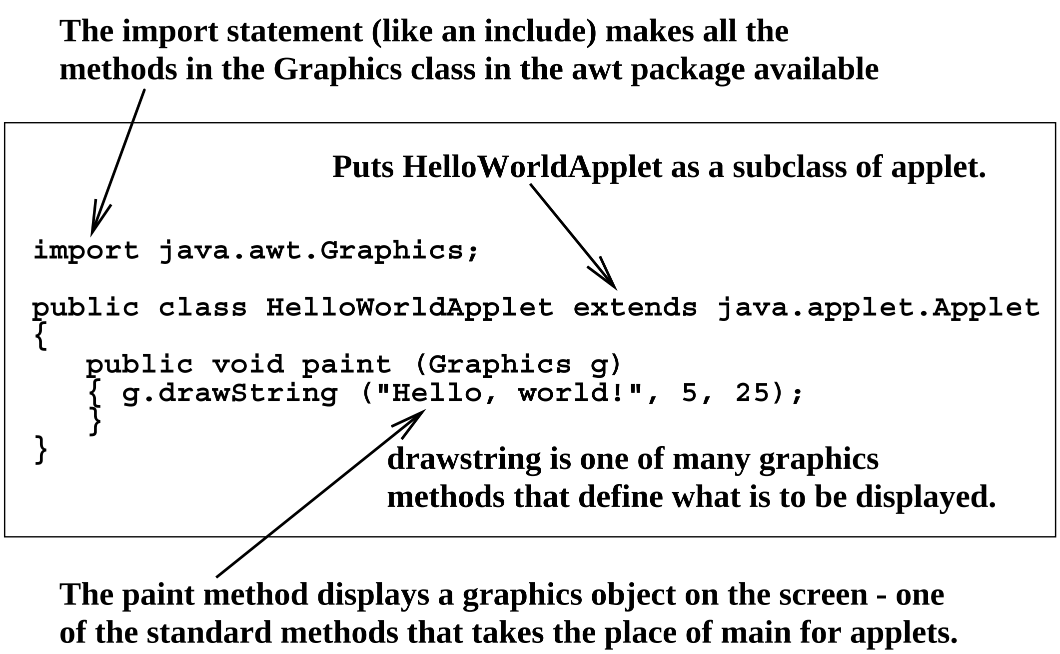


Figure 18: The Java Code for the Simplest Java Applet: Hello, World!.

You should name the file with your applet name, HelloWorldApplet.java, run the compiler (javac), getting a bytecode file HellowWorldApplet.class, which you put in a Web directory.

```
<html> <head>
<title> Simple Hello Page </title>
</head>
<body>
My Java applet says:
<applet code="HelloWorldApplet.class" width=150
height=25>
</body> </html>
```

Name of your applet class.

Browser uses a rectangule of width 150 pixels and height 25 pixels to display the applet.

Figure 19: Displaying Your Hello World Applet from a Web Page

well as the self-distributing software model of Figure 17, one will see native Java compilers that will get high performance by direct compilation of the Java language into the native machine code for particular machines.

3.3 JavaScript

JavaScript is a language that has some similarities, but many critical differences, from Java. One can view Java as an extension of a computer language (C++) and JavaScript as an extension of a text formatter HTML. Both are aimed at "animating Web pages" that is a superficial description of an interactive client-side application.

Figure 20 shows the simplest JavaScript "Hello World!" application. However, more interesting is the example of Figures 21 and 22, which shows a new HTML tag attribute ONCLICK being used to specify a JavaScript routine *compute* to process data specified in an HTML Form. We discussed, in Section 3.1, how this can be used to build very responsive client-side applications. Our final example, in Figure 23 shows how JavaScript can be used to generate parameterized HTML where JavaScript variables—possibly input in forms—are used to specify dynamic pieces of HTML pages. JavaScript is particularly useful in windows built from multiple frames as it allows actions in one frame to control the URL location and other properties of another frame.

Java was developed by Sun and is an industry standard. JavaScript was developed by Netscape and partially supported by other browser vendors including Microsoft. We believe that although it is no such a well established standard. JavaScript fulfills a clear role, and one can expect one or more technologies like this to be part of the Web "bag of tricks." In Section 3.2, we mentioned the AWT as the basic Java windowing toolkit. One can view JavaScript as the equivalent programmable AWT, but with the (Netscape) browser as the runtime. One can use the Java AWT to mimic JavaScript and have the advantage of a much more powerful and better

```
<HTML>
<HEAD>
<Title>A Test of JavaScript</Title>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!--A Conventional comment to hide JavaScript from old browsers
document.writeIn("<h1>Hello World!</h1>");
//scriptend-->
</SCRIPT>
<b>Continue with conventional HTML</b>
</BODY></HTML>
```

Figure 20: Hello World Example of Javascript

```
<HTML><Head><TITLE>Javascript with Forms</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--A Conventional comment to hide JavaScript from old browsers
function compute(form) {
if(confirm("Is this what you want?"))
  form.result.value=eval(form.expr.value);
else alert("Enter a new expression then!"); }
//scriptend-->
</SCRIPT></HEAD>
<BODY><FORM>
Enter an Expression:
<INPUT TYPE="text" NAME="expr" SIZE=15>
<INPUT TYPE="button" VALUE="DoIt!"
ONCLICK="compute(this.form)">
<BR>Result:
<INPUT TYPE="text" Name="result" SIZE=15>
<BR>
</FORM></BODY></HTML>
```

Figure 21: HTML Page for JavaScript

Enter An Express: 9+5 **DoIt!**

Result: 14

1) **confirm** is a native Javascript method popping up a window, requesting confirmation of requested acion

2) **alert** is a native Javascript method popping up a window with a message requiring user to place OK to get rid of

3) **onclick** = "Javascript Statement Block" naturally executes statement(s) when button clicked

Figure 22: HTML Display for JavaScript

```
<HTML><HEAD><TITLE>Javascript for Parameterizing HTML</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--A Conventional comment to hide JavaScript from old browsers
var imagewidth=600; // These could be changed by form input or some
var imagefile="npac.gif"; // computation based on size of window etc.
//scriptend-->
</SCRIPT></HEAD>

<BODY> ...... Bunch of Normal Stuff

<img width = "&{imagewidth};" src = "&{imagefile};"
```

Imagewidth and Imagefile are JavaScript Variables

his line involves JavaScript entities which allow dynamic names, sizes, etc. or HTML components defined on client side by user.

his line can also be written as five lines using the ful embedded language (!)

```
 SCRIPT LANGUAGE="Javascript">
 !--A Conventional comment to hide JavaScript from old browsers
 ocument.writeln('<img align=top width=' + imagewidth + ' src = "' + imagefile + '">

/scriptend-->
 /SCRIPT>
```

Figure 23: Parameterized HTML in JavaScript

defined language. However, JavaScript has a major advantage in that the current browsers essentially define a far more powerful runtime than that currently available to the Java AWT. In particular, the Hot Java browser, written entirely in Java, is currently not competitive with the standard non-Java browsers. JavaScript has a built-in advantage (independent of browser capabilities) that is fully interpreted, and it is much easier to rapidly prototype and modify new Web capabilities in JavaScript than in Java.

JavaScript is a more limited language than Java; JavaScript has objects, but no inheritance as in Java; Javascript relies implicitly on the undefined (to the user) browser multitasking threads whereas Java offers the user substantial control over the execution environment with well-developed thread support in the language. JavaScript is used for relatively simple client-side document processing; Java can be used much more extensively including applications involving image processing, simulation, and networking on client and server.

3.4 VRML for the Real World on the Web

VRML was inspired by the simple observation that the world is three dimensional, and one needs to manipulate and view on the Web three-dimensional objects, as well as the images and text supported so well by the earlier technologies. The initial VRML 1.0 was not a programming language, but rather a data descriptor allowing one to specify arbitrary static three-dimensional scenes with general dynamic viewpoint allowing navigation. The three-dimensional scenes could embed clickable regions (spawning arbitrary Web locations) in a natural generalization of the two-dimensional image maps available in HTML. VRML was first proposed in Spring 1994 with the initial 1.0 version available a year later. There is now an extended standard VRML 2.0, which we will discuss briefly later on. Note that VRML is an acronym for Virtual Reality (VR) Modeling Language, but VRML is applicable to any application involving three-dimensional objects, and is more general than Virtual Reality.

We illustrate VRML in Figure 24, which shows six lines of VRML 1.0 defining two unit size objects separated by three units in the x direction. VRML is based on SGI's Open Inventor graphics object database format with some restrictions and extensions allowing linking objects with particular Web sites. VRML three-dimensional worlds are built in a hierarchical fashion in terms of nodes grouped together by separators. *Cube* and *Sphere* are types of node as is *material* shown in Figure 25. Nodes have attributes, such as *diffusecolor* and *translation* in Figure 25. Grouping is illustrated in Figure 26 where the extra separator limits the effect of the *Material* node to the cube. The detailed syntax in Figures 24–26 are the old VRML 1.0 standard. There are some changes in VRML 2.0, but the essential ideas are unchanged.

A sophisticated use of VRML is shown in Figure 27 with a terrain overlayed by clouds from a weather simulation. However, the use of this technology has been handicapped by its need for high performance for both CPU and network. Some of this is inevitable—it takes more data to specify and more CPU processing to render three-dimensional objects than to process HTML text. However, much of the performance degradation was due to simplistic language design (data was transmitted as uncompressed ascii) and immature VRML browsers. We expect a combination of improved hardware and refinements in language and software to increase the use and usefulness of VRML.

The new version VRML 2.0 addresses some of the performance issues, but more importantly adds the concept of events "behaviors" that are essentially computer programs (scripts) that can

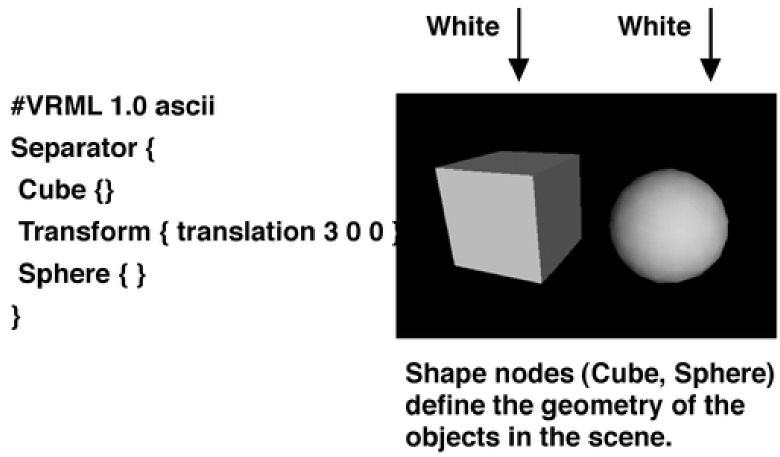


Figure 24: White Cube and White Sphere in VRML

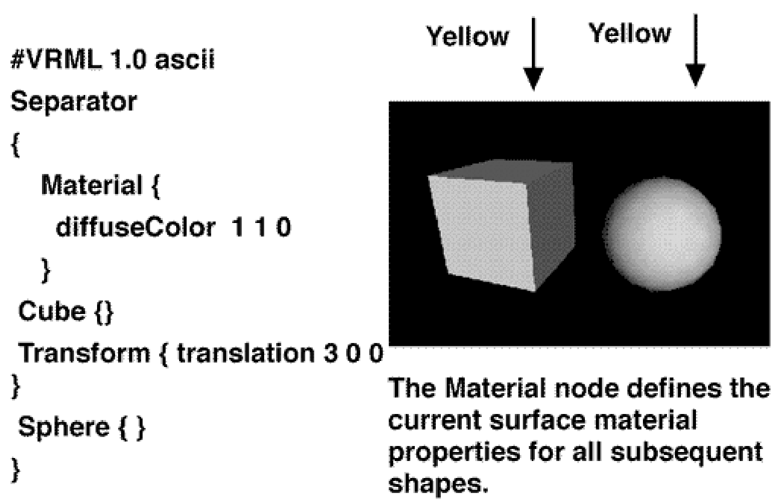


Figure 25: Yellow Cube and Yellow Sphere in VRML

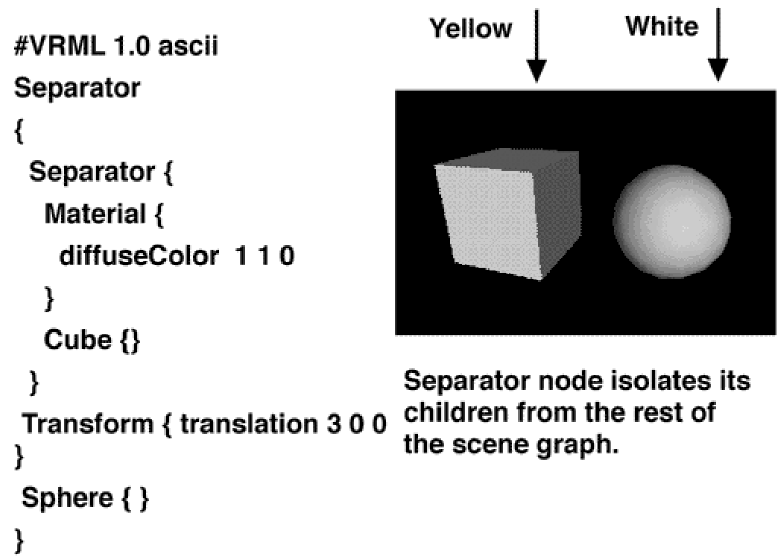


Figure 26: Yellow Cube and White Sphere in VRML

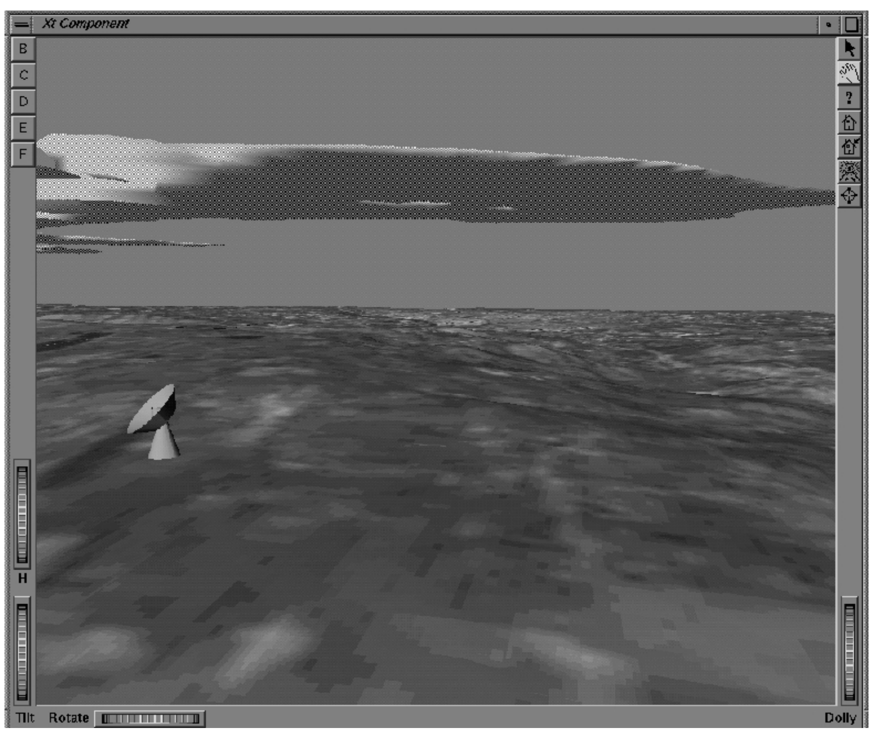


Figure 27: VRML for 3D Terrain with Weather Simulation Superimposed

be attached to the VRML nodes, and modify their attributes. Currently, Java and JavaScript scripts are supported. This allows scenes with objects moving with respect to each other, and interacting in ways defined by the scripts. This will allow one to generate true virtual worlds and has natural application to both VR and Web-based computer games. Although there is an agreed VRML 2.0 standard, which is becoming "official" as VRML97, there are other approaches. One can extend (*Java3D*) the Java runtime to support three-dimensional objects in a similar way to its current two-dimensional image support. Clearly, the ideas present in VRML 1.0 and VRML 2.0 will be part of critical base Web technologies; however, the particular implementation is much less clear than for Java, and we can only follow the Web community and see what paths are chosen.

3.5 Web-Linked Databases

This represents a different type of Web resource—namely, databases have been around a long time, but what's new is that they are now much cheaper and easier to use. Until two years ago, NPAC used essentially no commercial databases, except for a single research activity using parallel Oracle. Now, we have some dozen activities using them. These include

1. Newsgroup and "chat" multimedia messages stored in databases giving a searchable record of collaboration or discussion
2. Enterprise intranets with a Fortune 500 product catalog made available to the public through NPAC's Web database interface
3. Support of financial modelling "on demand" prototype with a database of stock information
4. Set of New York State images indexed in a database for educational use
5. Commercial CD-roms stored in databases for school reference
6. Storage of close-captioned text and metadata to index video
7. Electronic mail server with database replacing UNIX mail folders
8. Storage of full text of books to allow rapid search
9. Complete index of text in NPAC Web site
10. Index of customized chemistry information
11. Object database storage of three-dimensional maps of New York State
12. Publication and other information from NPAC's technical reports.

All these applications use Web-linked relational (Oracle) or object (Illustra) databases. The explosion at NPAC and other places of (relational) database use is easy to understand. Historically, we would have used flat files, or UNIX file system structure, to store such data. Note that Web site information organization or many UNIX mail systems use file systems to "label" data. The trend to substantially lower database prices (especially on PC platforms) and the convenience of Web interfaces has changed this, and given us systems that combine the indexing and secure storage capabilities of traditional databases with the friendly HTML or Java interfaces of the Web. Figure 8(a) shows a typical setup with a simple CGI Script linking Oracle to the Web. We have most experience with WOW, which has a very simple CGI Script, and the user query (input through an HTML form) is interpreted by Oracle's PL/SQL language. WoW PL/SQL libraries HTP and HPF that make it relatively easy to produce HTML pages that are passed from the database server to the client.

PL/SQL is a convenient high-level language for generating standard SQL queries, but it is Oracle specific. We expect the so called Java Database Connectivity (JDBC) with a host of Java methods for accessing any SQL database to be very important in the future.

In Figure 8(b), we emphasize that a complete Web delivery system could well involve many filters as data is stored in natural application oriented form, but transformed to suit network (e.g., compression) or browser (e.g., conversion to VRML). Java is a convenient language for building such filters.

Figure 28 illustrates the PL/SQL function to access a simple phone number database with simple HTML form (Figure 29 and resultant HTML Figure 30).

3.6 Remarks on Digital Audio and Video

Multimedia is of growing importance for the Web. Given the expectation that the Web will subsume current TV and cable services we can expect increasing amount of multimedia information. This implies important consequences for both the underlying network, and for the client and server. Initially, the Web offered downloaded (MPEG compressed) video, but this is not an appropriate approach except for short clips. Rather, one must use streaming servers where data is passed continuously from server to client. This implies important quality of service constraints on the network as one cannot afford delays. The current internet does not have the necessary bandwidths or reliable service except for audio (where the well-known Real Audio commercial product can deliver AM quality audio over modern phone lines). Of course, this audio and video delivery depends on compression technologies, such as fractal, MPEG, and Wavelet methods, whose details are closely guarded corporate secrets.

New networking hardware approaches, such as ISDN, ATM, and cable modems are very important for digital video as they offer high enough performance to support reasonable quality. However, the time scale and nature of their deployment is still unclear. One uncertainty is just how important to consumers, digital video is!

4 Six Typical Web Windows Application Areas

4.1 IntraNets and Business Enterprise Information Systems

Intranets have been popular recently. These correspond to building enterprise information systems using WebWindows technology Figure 9, as discussed in Section 2.1. WebWindows can encompass the world (clients talking to large servers around the globe) or just a single machine (a PC Web browser linked to a server on the same machine) (Figure 4. Further, we can apply these ideas flexibly to any enterprise of intermediate size and use Web technology to build general information services. These will use traditional concepts (e.g., databases) but link these to the

```
CREATE PACKAGE EXAMPLE IS
    procedure get_phoneno_by_name(name IN VARCHAR2); – only one procedure
END EXAMPLE;

CREATE PACKAGE BODY EXAMPLE IS
    CREATE PROCEDURE get_phoneno_by_name(name IN VARCHAR2) IS
    CURSOR person_cur(cname IN VARCHAR2) IS
    SELECT last_name,first_name,phone_no,phone_type
        from person_info_table,phone_list_table WHERE
        (person_info_table.person_id=phone_list_table.person_id) AND (last_name LIKE
          ('%' || LOWER(cname) || '%') OR first_name LIKE ('%' || LOWER(cname)
          || '%'));
          * lname person_info_table.last_name%TYPE; – a variable to hold last name
          * fname person_info_table.first_name%TYPE; – a variable to hold first name
          * phone phone_list_table.phone_no%TYPE; – a variable to hold phone no
          * ptype phone_list_table.phone_type%TYPE; – a variable to hold phone type
BEGIN
    htp.htitle('Query Results');
    htp.ulistOpen;
    OPEN person_cur(name); – open the cursor
    LOOP; – Fetch each row matching the query into variables repeatedly
    FETCH person_cur INTO lname,fname,phone,ptype;
    EXIT WHEN person_cur

    /* print out the query result */
    htp.p(htp.item ||' The '|| ptype ||' phone no. of '|| fname ||' '|| lname ||' : '|| phone);
    END LOOP;
    CLOSE person_cur; – close the cursor after it is done
    htp.ulistClose;
    END get_phoneno_by_name;
    END EXAMPLE;
```

Figure 28: the PL/SQL Package for a Phone Example

- A HTML form page to accept user search input and invoke the CGI script on the Web server to access the database

```
<html><header><title>A Web/Oracle
Phonebook Example</title></header>
<body>
<form action="http://myhostcps616/wow/get_phoneno_by_name">
Enter Last or First Name: <input type = "text" name="name" value="">
</form></body></html>
```

Figure 29: Web Page Search Interface to Wow Phone Example

rich WebTop environment. Note that this means one should not reject Web technologies because of today's low bandwidth or insecure Internet. If we build, today, a WebWindows Intranet, it can have whatever bandwidth and security the enterprise network supports. One of the applications explored by NPAC is a politics information system built around Newt Gingrich's recent visit of a politician to the University [Maxwell:96a]. This illustrates how WebWindows systems can be built for information storage and on-line discussion of any loosely knit enterprise from political parties through special interest organizations.

4.2 WebMed: Web Technologies in Health Care

The Intranet ideas above can, of course, be applied to medical enterprises where one can use Web-linked databases combined with several different user interfaces. Here, we use Java so that we can customize the 'look and feel' and access privileges so that doctors, hospital administrators, insurance companies, nurses, and patients can share the same information. NPAC has two interesting local projects—one with the University College of Nursing with a prototype system to help K–12 school nurses; the second uses simple virtual reality technology from David Warner, [Pulsar:97a] [Warner:95a], to allow disabled persons to both navigate the Web and to transmit medical sensor data from their homes to Web servers where it can then be analyzed conveniently by doctors.

4.3 Web Technology in the Financial Industry

An area of interest to NPAC is use of WebWindows in the financial industry. Here, Java can be used to provide excellent PC (home, office) interfaces to real time trading data. Java will allow downloading of simple modeling packages to aid on-line trading. However, today's complex financial instruments require computationally intensive Monte Carlo algorithms for accurate simulation. Here, we use Web technology to allow investors access to stock data, and simulation systems running on the classic MPPs developed in the HPCC program.

```
<html><header>
<title>Query Results</title></header>
<body><h1>Query Results</h1>
<ul>
<li> The office phone no. of Geoffrey Fox: 3154432163
<li> The home phone no. of Geoffrey Fox: 315xxxyyyy
</ul>
</body>
</html>
```

Figure 30: Result of Wow Phone Query is Another Page (assume user typed in 'geoffrey')

4.4 Web Technology in Distance Education

The Web can be expected to greatly improve education, both in traditional and distance formats. There is much debate as to the most effective learning methods—asynchronous or synchronous—and as to possible changes in the business model for education coming from the separation of the teaching and residential function of say, universities in a distance education model. We expect that these important questions can only be answered by a set of experiments using the approaches and technologies exemplified by the discussion below.

4.4.1 Infrastructure issues

Bandwidth The basic principle of multiple-use (Section 2.6) says that for distance education, we will use the hardware infrastructure, software, and other technology produced for the suite of NII applications including business, health care, and entertainment.

End-user access to the Web can be based on 28.8 kilobits/second or similar speed modems using POTS (Plain Old Telephone Service) connections, which in practical terms, limits us to displaying audio, text and graphics. Applications involving quality video delivery require ISDN access (128 kilobits/second) or ATM service (scales to Gigabit speeds, individual user typically requires a maximum of a few megabits/second).

Full screen NTSC video display requires about 1.5 megabits/second of bandwidth assuming MPEG1 compression. By comparison, a quad speed CDROM provides transfer rates of 0.6 megabits/second. We note that ISDN provides 10% of the performance needed for full motion video, but for educational applications we need perhaps only quarter screen displays with 15 frames per second (typical educational clip is one to two minutes). Improved compression schemes such as wavelets are becoming available but are not standardized and built into PCs as is MPEG.

The bandwidth required for digital video transmission (1.5 megabits/second) also supports the following applications relevant to distance education:

- Collaboration with real-time video of participants, supporting student-teacher and student-student and other individual interactions such as networked world-wide collaborations of students doing real science including remote control of instruments (telescopes, microscopes) and fetching data over web [NCSA:97b], [TANGO:97b];

- Simulations based on video animation produced either in batch mode or in real-time by a computer, including display of physical phenomena with tagged annotations linking beginning students with the insights of experts;
- Navigation in Geographical Information Systems using transmission of rendered scenes of 3-dimensional terrain (VRML based vector representation of video is more efficient than transmitting pixels), used for homework and student projects including electronic laboratories and field trips (see Figure 27);
- Text-indexed video clips selected from a large (100,000 hour or 70 terabyte) digital archive, supporting pedagogical (electronic books and similar material, also InfoVISION servers) and exploratory learning using the World Wide Web as a "virtual library" for projects.

Role of caching Caching content for use in distance education applications has important performance and quality of content implications. Education is typically carried out in a classroom setting, therefore a group of students browsing related content will naturally provide efficient use of cached resources. In a school experiment conducted by NASA Langley, 95% of the material accessed by students was found in cache, allowing a 28.8 kilobaud connection purchased by a school to look like a dedicated T1 line [Fox:95m].

Caching is also important to content quality. Pre-selecting material for later use in class can be used to ensure students browse relevant and appropriate material, and maintain efficiency in teacher preparation. Many schools limit Internet access due to the perception of easily accessible inappropriate material. Datamining the World Wide Web for resources relevant to K–12 classroom adds great value to Web content. KidsWeb [Kids:95a] was initially developed to support an educational program at NPAC for eighth graders in the Syracuse Area [YSP:95a], and is NPAC's most popular Internet access point.

As described in Section 2.2 and Figure 7, caching is also needed to support a hierarchical Web server/InfoVISION server scenario where content in high demand is replicated and stored close to end-user populations for bandwidth efficiency, and migrated to centralized archives for storage efficiency when demand lowers.

In future distance education experiments, we intend to distribute a CD-ROM to each student. Lessons will be delivered synchronously (all students are on the same page) by transmitting the same relative URL to each client. This approach uses modern (Java) collaboration technology such as Habanero [NCSA:97b] or Tango [TANGO:97b] to fetch data from the local CD-ROM, to display the same page on each client, for example a 100 kilobyte image. This effectively saves a factor of 1,000 in needed bandwidth and allows simple 28.8 kilobaud lines to support a graphically rich curricula.

4.4.2 Human interaction issues

Collaboration Collaboration is the aspect of the virtual K–12 school or university which is most sensitive to teacher-pupil separation. Distance education implies a distribution of students, an obvious obstacle to learning. However, Web technologies also allow school children to interact with their peers internationally. Distance education holds the promise of individualized learning on the users timeframe, but two-way communication between student and teacher must be supported. Collaboration between the learner and leading experts is technically possible through a hierarchical structure linking learners and a mix of distance university and leading expert partnerships. Learning remains a group activity even if conducted electronically, therefore multi-user collaboratory tools are needed to support group navigation, sharing of group experiences, and group authoring of content.

Collaboration is a particularly demanding distance education application because it cannot use caching and a hierarchical server structure, approaches which serve information access and delivery applications so well. Collaboration is sensitive to available bandwidth and requires the best compression and transport technology. As discussed above, a hybrid scheme involving a mix of locally cached (CD-ROM) material with high value interactive access at a distance is quite promising. Our Java collaboratory Tango supports this model, and we hope to perform a set of experiments with it soon.

Our project experience with videoconferencing suggests that 56 kilobaud is the lowest practical speed, ISDN (128 kbaud) is adequate, but full screen video at 30 frames per second requires the standard 1.5 megabits/sec bandwidth with MPEG1 compression. We are experimenting with the use of audio (which is practical at 28.8 kilobaud) as a form of annotation to enhance on-line course material. This can be supplemented by images or new video codecs such as Java H263 applets on low speed lines.

Authoring As discussed in Section 2.3, the ubiquity of the Web suggests that we will soon see Web operating systems replacing proprietary operating systems, and the growth of a Webtop (rather than proprietary desktop) personal productivity environment. Content authoring in a Web environment will become a central function in distance education.

Homework will be done in an electronic environment as part of the learning experience. Web-based content must therefore be interactive, and editable. Students must be ably to create, modify, and edit on-line hypermedia material as part of the learning process. As part of our computer science classes we have developed a virtual programming laboratory where students can develop, compile, and run (homework) problems [Dincer:97a], [Dincer:97b].

Active simulation Simulation support is needed to provide two-way interactivity that enables real-time interaction with a simulated world. Students must be able to interact with active simulations by guiding the process (e.g., setting parameters, triggering events), navigating regions of interest, and accessing explanatory annotations (written either by teachers, application experts, or other students).

Virtual reality could be a revolutionary technology to improve the learning environment by immersing student in physical or more abstract information spaces.

Navigation The Web is an excellent source of educational content (see Kids Web [Kids:95a], and ThinkQuest [Thinkquest:96a]) but the growing volume and complexity makes it too easy for students to get lost in cyberspace. Disciplined navigational tools are need, perhaps incorporating multi-sensory feedback [19] to create a coherent integration model of various multimedia modalities (hypertext, sound, video, simulations).

Structured navigation and free exploration must be balanced to allow the teacher to efficiently guide the investigation and the student to employ his or her own optimal model of learning. We have experimented in existing NPAC projects with unstructured exploration (e.g., KidsWeb, and a video archive) and structured courses and tutorials (e.g., on-line computational science courses [CPS:96a], [Ed:96a], [Fox:95n]).

Assessing the impact of new technologies and approaches on learning is itself an application where Web technologies are useful. Tools are needed to track students navigation pathways and their reactions to selected sequences, then we must use this information for evaluation and individual-student course refinement. It is straightforward to log the URLs of the pages accessed by teachers and students. We are using this today to link recorded audio of lectures to particular pages being discussed. Tango supports complete logging of a multimedia collaborative session. Related research on navigational patterns termed cognitive throughput by disabled learners is designed to anticipate the response of users in an information environment. Tracking navigational pathways and inter-target intervals allows us to anticipate future navigational direction (to pre-download content and improve performance), and support cognitive studies of user needs for improved information design [Sherman:97a].

The Unit of Information In traditional models, we teach and learn using lectures supported by books, notes, and presentations in the form of bulleted items and summary lists in a linear sequence. In the Web-based environment we envision, where a distributed hierarchy of information-on-demand servers are available, and students interactively explore active simulations, we must consider what is an appropriate unit of information.

The hyperlinked environment of the Web makes "sequential" books unnatural—people are better than computers at flipping sequentially through electronic pages. This may suggest an electronic encyclopedia model as the optimal model for the Web. This model would likely contain short, linked, modular entries produced by multiple authors having several different levels of description for a given subject.

The natural size of an information nugget is roughly a computer screenful. It is interesting to note that this is similar to the amount of data displayed on a viewgraph on an overhead projector. Thus we can expect presentation technology and techniques to be an important source of ideas for Web education.

4.4.3 Some Base NPAC Technology Experiments

Text-indexed video NPACs text-indexed video project combines basic research in video-on-demand technology, software design and implementation, and systems to provide an operational VoD service for educational applications. All development in this project is Web-based.

In contrast to entertainment oriented video dialtone systems where the content stream is nearly 100% video content, NPACs VoD project integrates video server technology within a larger information system. A video server management layer is linked to full text searchable databases and metadata used to access video content. This approach allows us to deliver digital video news-on-demand, educational documentary content, and multimedia coverage of events with educational relevance (e.g., a national political campaign event [Maxwell:96a]).

Examples of distance education applications of our VoD system include:

- Searching a database on topics such as siege of Stalingrad, arrest of Mahatma Ghandi, or threat to the Apollo 13 moon mission to retrieve the original, unedited film clips (in digital video format) from a Reuters historical video archive;
- Searching a documentary transcript database on technology in the classroom to retrieve segments of The Discovery Channel Assignment Discovery series on video graphics technology;

- Searching a transcript database on streamlining government bureaucracies to retrieve a segment of video from House Speakers Newt Gingrichs presentation and dialog with political science graduate students during a visit to Syracuse University [Maxwell:96a].

NPAC has focused on developing operational video server systems for education and related applications (e.g., decision support), while examining tradeoff s between image quality and bandwidth requirements. Transporting variable bit rate video streams (naturally occurring in video content) over a constant bit rate network channel (required in digital network delivery) requires a trade-off between image quality and bandwidth. Video compression techniques attempt to minimize the information content transmitted in each successive frames by carrying only the difference in scenes forward. The data requirements of displaying an action scene such as a sports event are much more demanding than for example a talk show [Podgorny:97a].

4.4.4 WebWisdom

Web Wisdom is a hierarchical tree of knowledge built from JavaScript and aimed at Netscape 3.0 Navigators [Fox:97e]. Web Wisdom is set up to display worlds of self defining Web Pages where the header of documents have appropriate JavaScript calls to define recursive, hierarchical structures.

This structure has natural applicability in education and in mail archives. Web Wisdom can handle several different worlds (limited only by the strength of JavaScript). We have currently implemented "Foilworld" which archives foils produced by NPAC, Family PhotoWorld and Administration.

FoilWorld is a part of WebWisdom devoted to accessing basic materials sorted by different criteria such as:

- Foilsets arranged in a simple alphabetic sort. This is for users who know the Foilset collection and want access to a particular set.

- Foilsets arranged by topics. This is for users who wish to browse topic areas. The user first accesses a list of areas, then an index to resources in each topic area.

FoilWorld has full support for Persuasion, Powerpoint, collections of screendumps, and general HTML pages. Family PhotoWorld uses the same technology to manage a collection of family photos. The digital images may come from scanning original prints, disks now available from commercial photofinishers, and images directly from digital cameras. The Administration area of WebWisdom allows users to select useful parameters, log URLs, and read and writing parameter sets.

The page syntax used in WebWisdom uses JavaScript to store metadata on each page which defines associated pages (children in the hierarchical structure). The page syntax also accommodates quick HTML versions of graphically rich pages, notes, associated illustrative (Java) programs, and multimedia material. We also have linked WebWisdom with an Oracle server which allows the user to search all pages at a given level or below a particular node.

A key facility of WebWisdom is a Java front-end which can create play-lists which produce new composite foilsets (the set of pages at any level of the WebWisdom world) with a mix of pages from different foilsets.

4.5 Manufacturing and the National Information Infrastructure

HPCC (High-Performance Computing and Communication) and the NII are used today, and can be expected to play a growing role in manufacturing, and more generally, engineering. For instance, the popular concept of agile manufacturing supposes the model where virtual corporations generate "products-on-demand." The NII is used to link collaborating organizations. Powerful computers are needed to support instant design (or more accurately redesign or customization) and sophisticated visualization and virtual reality "test drives" for the customer. At the corporate infrastructure level, concurrent engineering involves integration of the different component disciplines—such as design, manufacturing, and product life cycle support—involved in engineering. These general ideas are tested severely when they are applied to the design and manufacturing of complex systems such as automobiles, aircraft, and space vehicles such as shuttles. Both the complexity of these products, and in some sense the maturity of their design, places special constraints and challenges on HPCC and the NII.

High-performance computing is important in all aspects of the design of a new aircraft. However, it is worth noting that less than 5% of the initial costs of the Boeing 777 aircraft were incurred in computational fluid dynamics (CFD) airflow simulations—the "classic" Grand Challenge in this field. On the other hand, over 50% of these sunk costs could be attributed to overall systems issues. Thus, it is useful but not sufficient to study parallel computing for large scale CFD. This is "Amdahl's law for practical HPCC." If only 5% of a problem is parallelized, one can at best speed up and impact one's goals—affordability, time to market—by this small amount. HPCC, thus, must be fully integrated into the entire engineering enterprise to be effective. Very roughly, we can view the ratios of 5% to 50% as a measure of ratio of 1:10 of the relevance of parallel (classic MPPs) and distributed computing (i.e., the NII or Web!) in this case.

The maturity of the field is illustrated by the design criterion used today. In the past, much effort has been spent on improving performance—more speed, range, altitude, size. These are still critical under extreme conditions, but basically these just form a given design framework that suffices to buy you a place at the table (on the short-list). Rather, the key design criteria is competitiveness, including time to market, and total affordability. Although the design phase is not itself a major cost item, decisions made at this stage lock in most of the full life cycle cost of an aircraft with perhaps 80% of total cost split roughly equally between maintenance and manufacturing. Thus, it certainly would be important to apply HPCC/NII at the design phase to both shorten the design cycle (time to market) and lower the later ongoing costs of manufacturing and maintenance.

We take as an example the design of a future military aircraft—perhaps 10 years from now. This analysis is taken from a set of NASA sponsored activities centered on a study of ASOP—Affordable Systems Optimization Process [ASOP:95a]. This involved an industrial team, including Rockwell International, Northrop Grumman, McDonnell Douglas, General Electric, and General Motors. ASOP is one of several possible approaches to multidisciplinary analysis and design (MAD) and the results of the study should be generally valid to these other MAD systems. The hypothetical aircraft design and construction project could involve six major companies and 20,000 smaller subcontractors. This impressive virtual corporation would be very geographically dispersed on both a national and probably international scale. This project could involve some 50 engineers at the first conceptual design phase. The later preliminary and

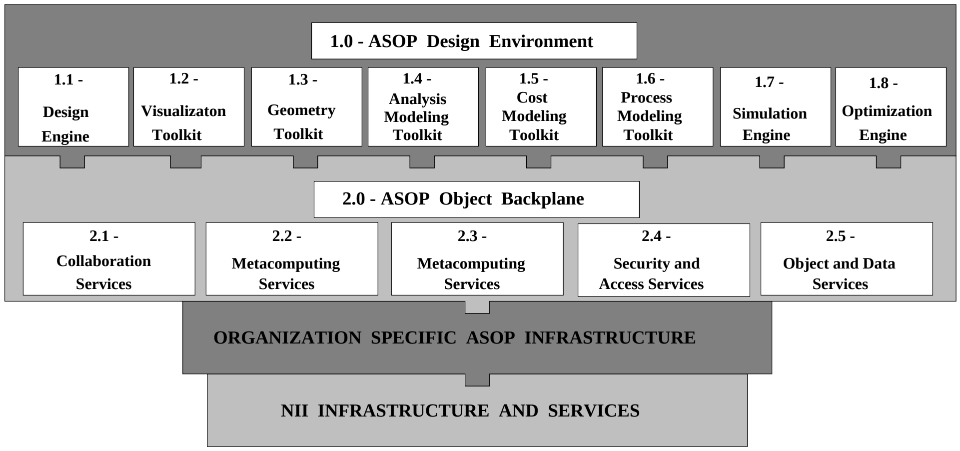


Figure 31: Affordable Systems Optimization Process (ASOP) Implemented on the NII for Aeronautics Systems

detailed design stages could involve 200 and 2,000 engineers, respectively. The design would be fully electronic and demand major computing, information systems, and networking resources. For instance, some 10,000 separate programs would be involved in the design. These would range from a parallel CFD airflow simulation around the plane to an expert system to plan location of an inspection port to optimize maintainability. There is a corresponding wide range of computing platforms from PCs to MPPs and a range of languages from spreadsheets to High-Performance Fortran. The integrated multidisciplinary optimization does not involve blindly linking all these programs together, but rather a large number of suboptimizations involving at one time a small cluster of these base programs. Here we see clearly, an essential role of the NII to implement these set of optimizations, which could well need linking of geographically separated compute and information systems. An aircraft is, of course, a very precise system, which must work essentially flawlessly. This requirement implies a very strict coordination and control of the many different components of the aircraft design. Typically, there will be a master systems database to which all activities are synchronized at regular intervals–perhaps every month. The clustered suboptimizations represent a set of limited excursions from this base design that are managed in a loosely synchronous fashion on a monthly basis. The configuration management and database system are both critical and represent a major difference between manufacturing and command and control, where in the latter case, real time "as good as you can do" response, is more important than a set of precisely controlled activities. These issues are characteristic of the high-performance distributed computing (HPDC) where, although loosely coupled, the computers on our global network are linked to "solve a single problem."

ASOP is designed as a software backplane (the NII or Web) linking eight major services or modules shown in Figure 31. These are design (process controller) engine, visualization, optimization engine, simulation engine, process (manufacturing, productibility, supportability) modeling toolkit, costing toolkit, analytic modeling toolkit, and geometry toolkit. These are linked to a set of databases defining both the product and also the component properties.

4.6 Command and Control or Crisis Management on the NII

Command Control (sometimes adding in Computing, Communications, Intelligence Surveillance, and Battle Management with abbreviations lumped together as BMC^{4}IS) is the task of managing and planning a military operation. It is very similar to the civilian area of Crisis management, where the operations involve combating effects of hurricanes, earthquakes, chemical spills, forest fires, etc. [NRC:96a]. Telemedicine was once considered a video conferencing application, but now this is becoming like command control in the interventional informatics concept [Warner:95a] where the doctor (cf. commander) needs a rich interactive information to win the "battle" i.e., treating the patient. Both the military and civilian cases have computational "nuggets" where parallel computing is relevant. These include processing sensor data (signal and image processing) and simulations of such things as expected weather patterns and chemical plumes. One also needs large-scale multimedia databases with capabilities similar to these described in Section 4.4.3. We have built a powerful Web based command and control prototype on top of TANGO [Beca:97b], [Fox:97c] that illustrates many of these concepts..

The NII is needed to link military planners and decision makers, crisis managers, experts at so-called anchor desks, workers (warriors) in the field, information sources such as cable news feeds, and large-scale database and simulation engines.

A key characteristic of the required NII support is adaptivity. Crises and battles can occur anywhere and destroy an arbitrary fraction of the existing infrastructure. Adaptivity means making the best use of the remaining links, but also deploying and integrating well mobile enhancements. The information infrastructure must exhibit security and reliability or at least excellent fault tolerance (adaptivity). Network management must deal with the unexpected capacity demands and real time constraints. Priority schemes must allow when needed critical information (such as chemical plume monitoring and military sensor data) precedence over less time critical information, such as background network video footage.

Needed computing resources will vary from portable handheld systems to large backend MPPs. As there will be unpredictable battery (power) and bandwidth constraints, it is important that uniform user interfaces and similar services be available on all platforms with, of course, the fidelity and quality of a service reflecting the intrinsic power of a given computer. As with the communications infrastructure, we must cope with unexpected capacity demands. As long as the NII is deployed nationally, computational capacity can be exploited in remote sites. The Department of Defense envisages using the basic NII (GII) infrastructure for command and control, augmented by "theater extensions" to bring needed communications into critical areas. The "take it as it is" characteristic of command and control requires that operating systems and programming models support a general adaptive mix (metacomputer) of coordinated geographically distributed but networked computers. This network will adaptively link available people (using perhaps personal digital assistants) to large-scale computation on MPPs and other platforms. There are large computational requirements when forecasting in real-time physical phenomena, such as the weather effects on a projected military action, forest fires, hurricanes, and the structure of damaged buildings. On a longer time scale, simulation can be used for contingency planning and capability assessment. Training with simulated virtual worlds supporting televirtuality, requires major computational resources. In the information arena, applications include datamining to detect anomalous entries (outliers) in large federated multimedia databases. Data fusion including sensor input and processing, geographical information systems (with perhaps three-dimensional terrain rendering), and stereo reconstruction from multiple video streams are examples of compute intensive image processing forming part of the needed Web-based distributed collaborative computing environment.

A critical need for information management involves the best possible high-level extraction of knowledge from databanks—the crisis manager must make judgments in unexpected urgent situations—we cannot carefully tailor and massage data ahead of time. Rather, we need to search a disparate set of multimedia databases. As well as knowledge extraction from particular information sources, the systematic use of metadata allowing fast coarse grain searching is very important. This is a specific example of the importance of standards in expediting access to "unexpected" databases. One requires access to databases specific to crisis region or battlefield, and widespread availability of such geographic and community information in electronic form is essential. There are very difficult policy and security issues, for many of these databases need to be made instantly available in a hassle-free fashion to the military commander or crisis manager—this could run counter to proprietary and security classification constraints. The information system should allow both network news and warriors in the field to deposit in near real-time, digital versions of their crisis and battlefield videos and images. Web linked databases should make this universal convenient access more feasible.

As mentioned, we expect that human and computer expertise to be available in "anchor desks" to support instant decisions in the heat of the battle. These have been used in a set of military exercises called JWID (Joint Warrior Interoperability Demonstrations). We note that this information scenario is a real-time version of the InfoVISiON service needed to support the society of the Information Age.

Command and Control has historically used distributed computing, as the relevant computer and communication resources, are naturally distributed, and not centralized into a single MPP. We see this growing into a Web-based information infrastructure for all the nation's enterprises, including business, education, and society.

Glossary of Concepts and Acronyms

Applets An application interface where referencing (perhaps by a mouse click) a remote application as a hyperlink to a *server* causes it to be downloaded and run on the *client*.

ASOP (Affordable Systems Optimization Process) refers to a process using multidisciplinary optimization to produce more affordable systems

CAVE A room with stereo images on the walls designed to create a televirtual environment.

CFD (Computational Fluid Dynamics) refers to computational solutions of differential equations, such as the Navier Stokes set, describing fluid motion.

CGI Standard Web server interface allowing sophisticated server extensions.

Clients Support browsing of hyperlinked documents, but have no internal interactive/compute capability.

COTS (Customer Off The Shelf)—An important concept in Defense Systems, labelling the use of commercial software snf system components rather than specialized limits which are expensive and hard to upgrade, even if initially of higher capability.

Data Fusion A common *command and control* approach where the disparate sources of information available to a military or civilian commander or planner, are integrated (or fused) together. Often, a *GIS* is used as the underlying environment.

Distributed Computing The use of networked heterogeneous computers to solve a single problem. The nodes (individual computers) are typically loosely coupled.

Geographical Information System (GIS) A user interface where information is displayed at locations on a digital map. Typically, this involves several possible overlays with different types of information. Functions, such as image processing and planning (such as shortest path) can be invoked.

Global Information Infrastructure (GII) The GII is the natural world-wide extension of the *NII* with comparable exciting vision and uncertain vague definition.

High-Performance Computing & Communications (HPCC) Refers generically to the federal initiatives, and associated projects and technologies that encompass *parallel computing*, *HPDC*, and the *NII*.

High-Performance Distributed Computing (HPDC) The use of distributed networked computers to achieve high performance on a single problem, i.e., the computers are coordinated and synchronized to achieve a common goal.

Hypertext Markup Language (HTML) A syntax for describing documents to be displayed on the *World Wide Web*.

Hypertext Transport Protocol (HTTP) The *protocol* used in the communication between *Web Servers* and *clients*.

InfoVISiON Information, Video, Imagery, and Simulation ON demand is scenario where *multimedia servers* deliver multimedia information to clients on demand—at the click of the user's mouse.

Integrated Service Data Network (ISDN) A digital multimedia service standard with a performance of typically 128 kilobits/sec, but with possibility of higher performance. ISDN can be implemented using existing telephone (*POTS*) wiring, but does not have the necessary performance of 1–20 megabits/second needed for full screen TV display at either VHS or high definition TV (HDTV) resolution. Digital video can be usefully sent with ISDN by using quarter screen resolution and/or lower (than 30 per second) frame rate.

Internet A complex set of interlinked national and global networks using the IP messaging protocol, and transferring data, electronic mail, and *World Wide Web*. In 1995, some 20 million people could access Internet—typically by *POTS*. The Internet has some high-speed links, but the majority of transmissions achieve (1995) bandwidths of at best 100 kilobytes/sec. the Internet could be used as the network to support a *metacomputer*, but the limited *bandwidth* indicates that *HPDC* could only be achieved for *embarrassingly parallel* problems.

Java A distributed computing language (*Web Technology*) developed by Sun, which has similarities with C++ but supports *Applets*.

JavaScript Interpreted language for manipulating Web components.

JPEG A compression technology for images based on using discrete fourier transforms on subimages. The compressed image quality can be varied and typically one chooses a compression factor of about 20:1.

JDBC (Java Database Connectivity) set of methods and drivers linking Java to general relational databases.

MAD (Multidisciplinary Analysis and Design, or Multidisplinary Optimization) refers to the coupling of several areas, such as structural dynamics and fluid flow in a combine tradeoff to produce higher capability vehicles.

Massively Parallel Processing (MPP) The strict definition of MPP is a machine with many interconnected processors, where 'many' is dependent on the state of the art. Currently, the majority of high-end machines have fewer than 256 processors. A more practical definition of an MPP is a machine whose architecture is capable of having arbitrarily many processors—that is, it is scalable. In particular, machines with a distributed memory design (in comparison with shared memory designs) are usually synonymous with MPPs since they are not limited to a certain number of processors. In this sense, "many" is a number larger than the current largest number of processors in a shared-memory machine.

Metacomputer This term describes a collection of heterogeneous computers networked by a high-speed wide area network. Such an environment would recognize the strengths of each machine in the Metacomputer, and use it accordingly to efficiently solve so-called *Metaproblems*. The *World Wide Web* has the potential to be a physical realization of a Metacomputer.

Metaproblem This term describes a class of problem which is outside the scope of a single computer architectures, but is instead best run on a Metacomputer with many disparate designs. These problems consist of many constituent subproblems. An example is the design and manufacture of a modern aircraft, which presents problems in geometry grid generation, fluid flow, acoustics, structural analysis, operational research, visualization, and database management. The Metacomputer for such a Metaproblem would be networked workstations, array processors, vector supercomputers, massively parallel processors, and visualization engines.

MIME Data format allowing agent-like (extended e-mail) communication.

MPEG, MPEG2, MPEG4 Compression technologies for video based on combining JPEG like compression for individual frames with differencing schemes for the motion between frames.

Multimedia Server or Client Multimedia refers to information (digital data) with different modalities, including text, images, video, and computer generated simulation. Servers dispense this data, and clients receive it. Some form of browsing, or searching, establishes which data is to be transferred. See also *InfoVISiON*.

Multipurpose Internet Mail Extension (MIME) The format used in sending multimedia messages between *Web Clients* and *Servers* that is borrowed from that defined for electronic mail.

National Information Infrastructure (NII) The collection of *ATM*, cable, *ISDN*, *POTS*, satellite, and wireless networks connecting the collection of 10^8–10^9 computers that will be deployed across the U.S.A. as set-top boxes, PCs, workstations, and MPPs in the future.

The NII can be viewed as just the network infrastructure or the full collection of networks, computers, and overlayed software services. The *Internet* and *World Wide Web* are a prototype of the NII.

PERL An interpreted language particularly targeted at text manipulations and systems programming.

Petaop or Petaflop The performance unit for a computer capable of 10^{15} (floating point) operations per second. Expected peak performance of high-end supercomputers in 2007 (aggressive technology) or later [Fox:97a], [http://www.aero.hq.nasa.gov/hpcc/petaflops/peta.html].

POTS The conventional twisted pair based Plain Old Telephone Service.

Servers Read HTTP and deliver requested service to client.

SNAP (Scalable Networks and Platforms) Gordon Bell's concept of a growing trend to commodity computers, and networks exemplified by ATM connected PCs running Windows NT.

Supercomputer A term describing the most powerful available computers with a price tag in the range \$10→\$100 M. Supercomputers can either use massive parallelism or a more conventional design.

Televirtual The ultimate computer illusion using *Virtual Reality* where the user is fully integrated into a simulated environment and so can interact naturally with fellow users distributed around the globe.

Teraflop A performance unit for computers capable of 10^{12} floating point operations per second. Currrent (1997) *supercomputers* can reach this level on selected applications [http://www.intel.com/pressroom/archive/releases/cn121796.htm].

URL Universal resource locator or addressing scheme for items on the Web.

Virtual Reality An illusion created by immersing a computer user in a three-dimensional interactive simulated environment.

Virtual Reality Modeling Language (VRML) A "three-dimensional" HTML that can be used to give a universal description of three-dimensional objects that supports *hyperlinks* to additional information.

WebTop refers to the implementation of a set of standard desk top and personal computer tools, which are essential in any computing environment.

Web Clients and Servers A distributed set of clients (requesters and receivers of services) and servers (receiving and satisfying requests from clients) using *Web Technologies.*

WebWindows The operating environment created on the World Wide Web to manage a distributed set of networked computers. WebWindows is built from *Web clients* and *Web servers.*

WebWork [Fox:95a] An environment proposed by Boston University, Cooperating Systems Corporation, and Syracuse University, which integrates computing and information services to support a rich distributed programming environment.

World Wide Web and Web Technologies A very important software model for accessing information on the Internet based on hyperlinks supported by *Web technologies*, such as *HTTP*, *HTML*, *MIME*, *Java*, *Applets*, and *VRML.*

References

[Andreessen:93a] Andreessen, M., "NCSA Mosaic: technical summary," May 1993. Technical report, NCSA.

[ASOP:95a] "Definition of requirements for an aeronautics affordable systems optimization process," in *MADIC Multidisciplinary Analysis and Design Industrial Consortium*, December 1995. http://www.npac.syr.edu/users/gcf/ASOPreport/ASOPRQD2.html.

[Beca:97a] Beca, L., Cheng, G., Fox, G. C., Jurga, T., Olszewski, K., Podgorny, M., Sokolowski, P., and Walczak, K. "Java enabling collaborative education healthcare and computing," *Concurrency: Practice and Experience*, March 1997. To be published in the Java Special Issue. Guest Editor: Geoffrey Fox.

[Beca:97b] Beca, L., Cheng, G., Fox, G. C., Jurga, T., Olszewski, K., Podgorny, M., Sokolowski, P., and Walczak, K. "Web technologies for collaborative visualization and simulation," in *Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing*. SIAM, January 1997. Syracuse University Technical Report SCCS-786.

[Bell:95a] Bell, G. "SNAP (scalable network and platforms): A view of computing in 2000+." Technical Report 24, RCI, Ltd., 1995.

[Bhatia:97a] Bhatia, D., Burzevski, V., Camuseva, M., Fox, G., Furmanski, W., and Premchandran, G. "WebFlow—a visual programming paradigm for Web/Java based coarse grain distributed computing," *Concurrency: Practice and Experience*, March 1997. To be published in the Java Special Issue. Guest Editor: Geoffrey Fox. Syracuse University Technical Report SCCS-787.

[CPS:96a] "CPS616 technology and applications for the information age." http://www.npac.syr.edu/projects/cps616spring96/index.html.

[Dincer:97a] Dincer, K., and Fox, G. C. "Using Java and JavaScript in the virtual programming laboratory: A web-based parallel programming environment," *Concurrency: Practice and Experience*, March 1997. To be published in the Java Special Issue. Guest Editor: Geoffrey Fox. Syracuse University Technical Report SCCS-788.

[Dincer:97b] Dincer, K., and Fox, G. C. "Design issues in building web-based programming environments." Technical report, Syracuse University, NPAC, 1997.

[Ed:96a] "Overview of education activities at NPAC," November 1996. http://www.npac.syr.edu/users/gcf/sc96/ed.html.

[Fox:95a] Fox, G. C., Furmanski, W., Chen, M., Rebbi, C., and Cowie, J. H. "WebWork: integrated programming environment tools for national and grand challenges." Technical Report SCCS-715, Syracuse University, NPAC, Syracuse, NY, June 1995. Joint Boston-CSC-NPAC Project Plan to Develop WebWork (unpublished).

[Fox:95c] Fox, G. C. "Software and hardware requirements for some applications of parallel computing to industrial problems." Technical Report SCCS-717, Syracuse University, NPAC, Syracuse, NY, June 1995. Submitted to ICASE for publication (6/22/95); revised SCCS-134c.

[Fox:95m] Fox, G. C. "Technologies and issues for virtual university," in *High Performance Computing '95, Tutorial*, August 1995. http://www.npac.syr.edu/users/gcf/virtuniv95/.

[Fox:95n] Fox, G. C. "Web technologies for education," in *Supercomputing '95, Tutorial*, December 1995. http://www.npac.syr.edu/users/gcf/sc95tutorial.html.

[Fox:96c] Fox, G. C., and Furmanski, W. "SNAP, Crackle, WebWindows!," *RCI, Ltd., Management white Paper*, (29), 1996. Syracuse University Technical Report SCCS-758.

[Fox:97a] Fox, G. C., and Furmanski, W. "Computing on the web—new approaches to parallel processing—petaop and exaop performance," *IEEE Internet Computing*, January 1997. To be published. Syracuse University Technical Report SCCS-784.

[Fox:97c] Fox, G. C., and Furmanski, W. "Java and Web technologies for simulation and modelling in computational science and engineering," in *Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing*. SIAM, January 1997. To be published. Syracuse University Technical Report SCCS-785.

[Fox:97e] Fox, G. C., "Web Wisdom hierarchical tree of knowledge." http://www.npac.syr.edu/users/gcf/wisdom/wisdom.html.

[IDS:97a] Network Wizards, "Internet Domain Survey." http://www.nw.com/zone/www/firstnames.html.

[Kids:95a] "Kids Web." A World Wide Web Digital Library for Schoolkids. http://www.npac.syr.edu/textbook/kidsweb/.

[Maxwell:96a] "Maxwell Agenda 96." http://kopernik.npac.syr.edu:8080/campaign96.

[NCSA:97b] "NCSA Habanero." http://havefun.ncsa.uiuc.edu/.

[NRC:96a] Steering Committee. "Computing and communications in the extreme: Research for crisis management and other applications," in *Workshop Series on High Performance Computing and Communications*. National Academy Press, Washington, D.C., 1996. Computer Science and Telecommunications Board, Commission on Physical Sciences, Mathematics, and Applications.

[Pulsar:97a] "A Web resource for a set of activities aimed at helping those with disabilities." http://www.pulsar.org.

[Podgorny:97a] Podgorny, M., "Northeast Parallel Architectures Center Video on Demand Project." http://trurl/rlvod/index.html.

[Sherman:97a] Sherman, E., Warner, D., Lipson, E., Guo, J., Lee, M., Berlin, A., Blanatovich, A., Panesar, R., and Shah, S., "Pulsar: The Web source for the disabled." http://www.npac.syr.edu/projects/pulsar/.

[TANGO:97b] "Tango." http://trurl.npac.syr.edu/t̃oju/CApp/newCA.html.

[Thinkquest:96a] "NPAC database interface to 1996 Thinkquest competition for K–12 student-teacher teams." http://www.npac.syr.edu/thinkquest/pub/index.html.

[Warner:95a] Warner, D., Sale, J., and Viirre, E. "A systems approach for developing an integrative healthcare information distribution infrastructure." Technical Report SCCS-749, Syracuse University, NPAC, Syracuse, NY, November 1995.

[YSP:95a] "The Young Scholars Program at Syracuse University." http://www.npac.syr.edu/EDUCATION/PUB/YSP/.