SNAP, Crackle, WebWindows!

Geoffrey C. Fox
gcf@npac.syr.edu

Wojtek Furmanski
furm@npac.syr.edu

Northeast Parallel Architectures Center
111 College Place
Syracuse University
Syracuse, New York 13244-4100
http://www.npac.syr.edu

Abstract

We elaborate the SNAP—Scalable (ATM) Network and (PC) Platforms—view of computing in the year 2000. The World Wide Web will continue its rapid evolution, and in the future, applications will not be 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, WebLotus123, and WebNotes built in modular dynamic fashion, and undermining the business model for large software companies.

We define a layered WebWindows software architecture in which applications are built on top of multi-use services. We discuss examples including business enterprise systems (IntraNets), health care, financial services and education. HPCC is implicit throughout this discussion for there is no larger parallel system than the World Wide metacomputer. We suggest building the MPP programming environment in terms of pervasive sustainable WebWindows technologies. In particular, WebFlow will support naturally dataflow integrating data and compute intensive applications on distributed heterogeneous systems.

1 SNAP and the Bottom-up Approach

Gordon Bell [Bell:95a] described the natural snappy evolution of computing to a universal parallel or distributed system of commodity networks (such as ATM), and commodity PC compute nodes. Here, we build on his extrapolation, and describe our vision of corresponding software environment at both the application and operating system levels. The current HPCC (High Performance Computing and Communications) activities can interact with this evolving world in two distinct ways–firstly, the World Wide system of several hundred million PC's and set top boxes is a remarkable parallel computer to which we can apply our expertise in parallel algorithms and methodology. Secondly, it is natural to build parallel computing environments in terms of pervasive World Wide Web (WWW) technologies. This bottom-up approach illustrated in Figure 1, should be contrasted with the more typical top-down HPCC approach. As an example, it is certainly laudable and useful to port HPF (High Performance

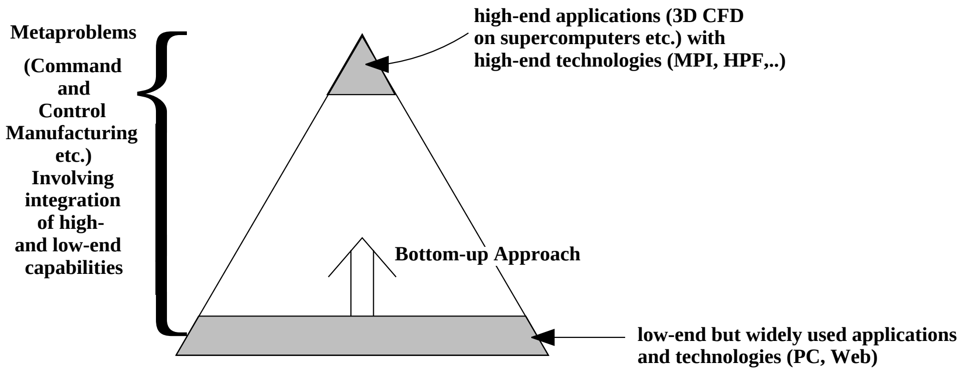


Figure 1:

Fortran) to run on a SNAP network of PC's; this will allow the (small) HPCC community access to such hardware, but it will not persuade the PC community to use HPF. Our bottom-up philosophy would be illustrated by the development of parallel versions of Java—the new language of the Web—optimized for MPPs (Massively Parallel Processors).

1.1 The NII—National Information Structure

The SNAP vision recognizes that the World Wide Web (NII), MPPs 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. 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 by America Online, Netscape, Microsoft, and others.

In Figure 2, 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 Massively Parallel Machines (MPPs) 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 *Information, Video, Imagery*, and *Simulation on Demand.* This 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 few percent of its storage devoted to video—the future NII will be dominated by video data. Each NII offramp will, as shown in Figure 3, 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 compressed

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

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 2, we estimate comparable investment in clients and servers and find an *InfoVISiON* or *WebServer* market that is at least an order of magnitude larger than that for supercomputers. This illustrates that it is the NII, and not large-scale number crunching, which is the best opportunity for parallel processing. Notice also that the compute power contained in this future NII is some 10–100 PetaFLOPS—far larger than the compute capability of an individual TeraFLOPS supercomputer. This motivates our interest in WebWork, described in Section 4, and aimed at extending Web to compute servers, and harnessing the power of the World Wide Metacomputer.

One could view this *InfoVISiON* scenario as the most amazing client-server application with 10^8 clients and 10^4 large servers. However, Vice President Gore has articulated NII democracy with everybody able to host information. Further new technology, just as Java and WebTools (Section 4), suggests instead 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 MPPs). It appears that although powerful, the communication backbone 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 4. When Jurassic Park VI is released on the Hollywood Server, one will

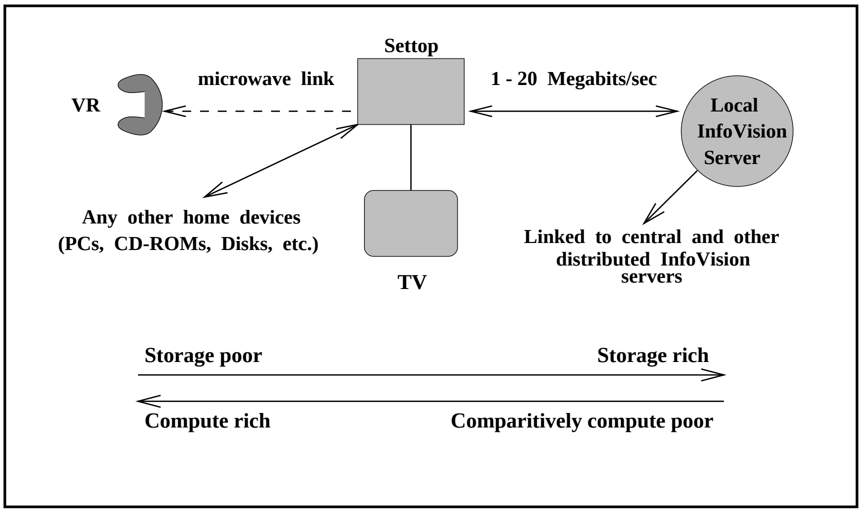


Figure 3: The basic *InfoVISiON* scenario as seen by a home in the year 2000 with an intelligent settop box interfacing the digital home to a hierarchical network of InfoVISiON servers

not have everybody accessing it there at an average distance of some 1,500 miles. Rather, this "hot" movie will be cascaded down 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 WebWindows

Figures 5 and 6 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.

We believe that the current trend to build applications for this architecture will accelerate. Now, we see WebMail, WebDatabase (see Figure 7), 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) and so Microsoft has no special advantage 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) but also produce new uses of computers. The current Web roughly implements a client-server computing model, but as client-side Web technologies improve one sees an evolution to a new paradigm—the server–server computing model of Section 1.1. This does not say that clients will disappear, but rather they will evolve to incorporate much of the functionality now in servers. In fact, current clients are much larger software packages than

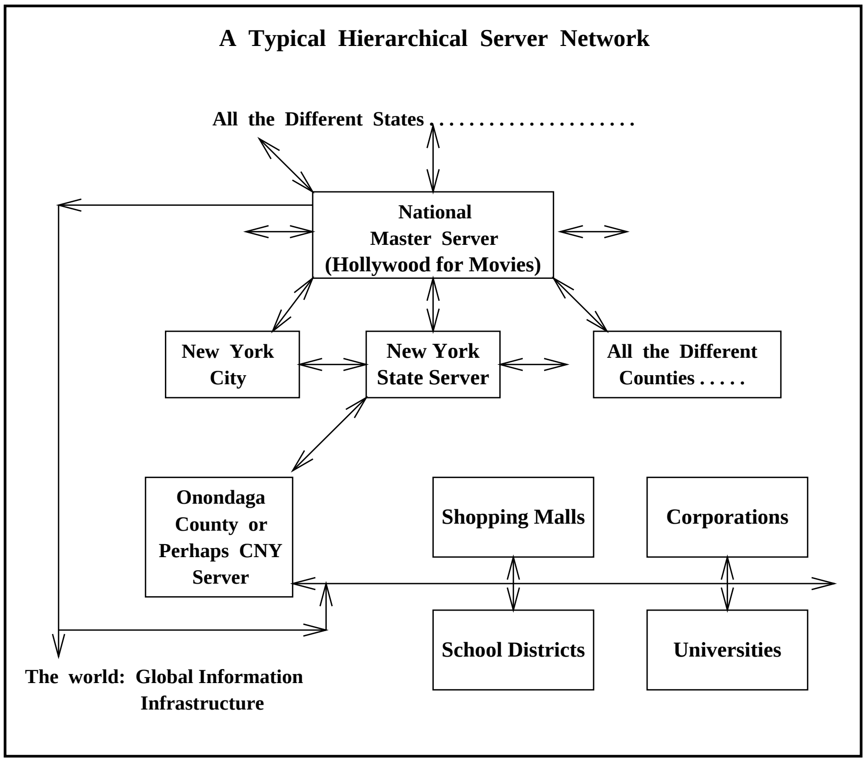


Figure 4: A typical hierarchical server network depicted for a master system in Hollywood cascading down with a fragment of node systems shown for central New York

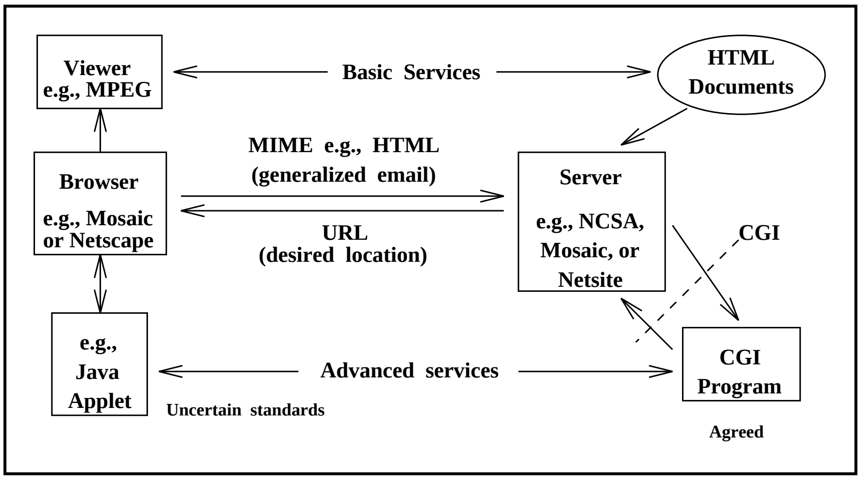


Figure 5: A simple representation of the "old" (pre 1996) software architecture of the WWW showing emerging advanced services, as well as the basic technology

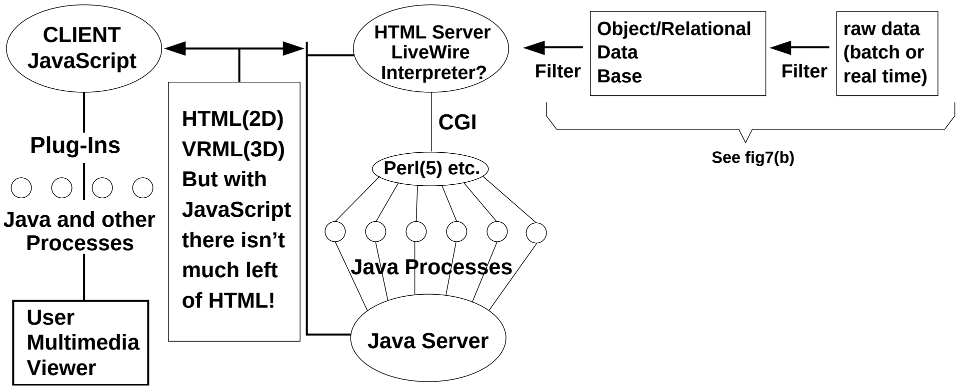
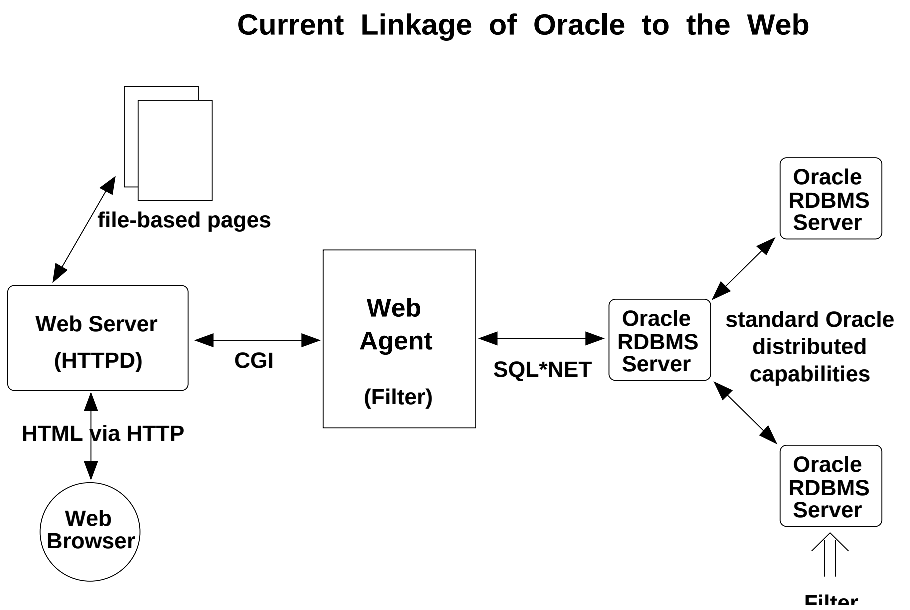


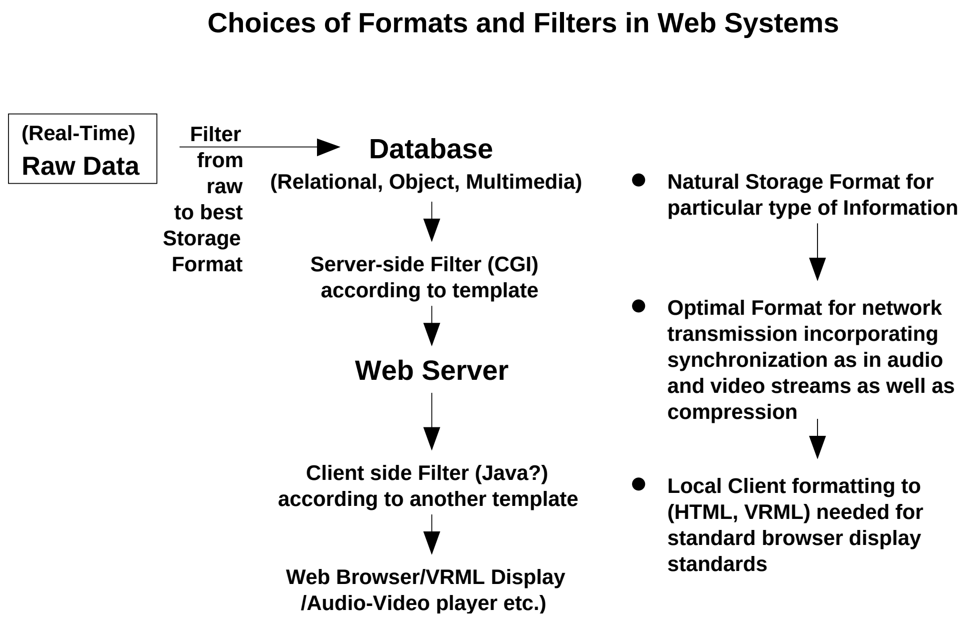
Figure 6: The 1996 Java/Netscape Client Server Model with evolving/confusing/overlapping capabilities.

Table 1: Some WorldWide Web (WWW) technologies and services

- **Clients** (such as Mosaic and Netscape) support browsing of hyper-linked documents, but have no internal interactive/compute capability [Andreessen:93a]
- **Servers** read HTTP and deliver requested service to client
- **HTML**—a document format supporting hyperlinks
- **HTTP**—a Transport Protocol defining Interaction between Web servers and Clients
- **MIME**—a data format allowing agent-like (extended email) communication
- **CGI**—a standard interface allowing sophisticated server extensions
- **PERL**—a rapid prototyping language (script) aimed at text and file manipulation
- **Java**—a semi-interpreted C++ like language supporting an applet model of distributed computing
- **JavaScript**—an interpreted language for integrating Web components
- **VRML 1.0**—a universal description for static three-dimensional objects
- **VRML 2.0**—a dynamic 3D descriptor for virtual environments
- **Livemedia**—a set of standards for multimedia applications on the Web



(a)



(b)

Figure 7: (a) Oracle-Web Integration Architecture; (b) Choices of Formats and Filters in Web Systems

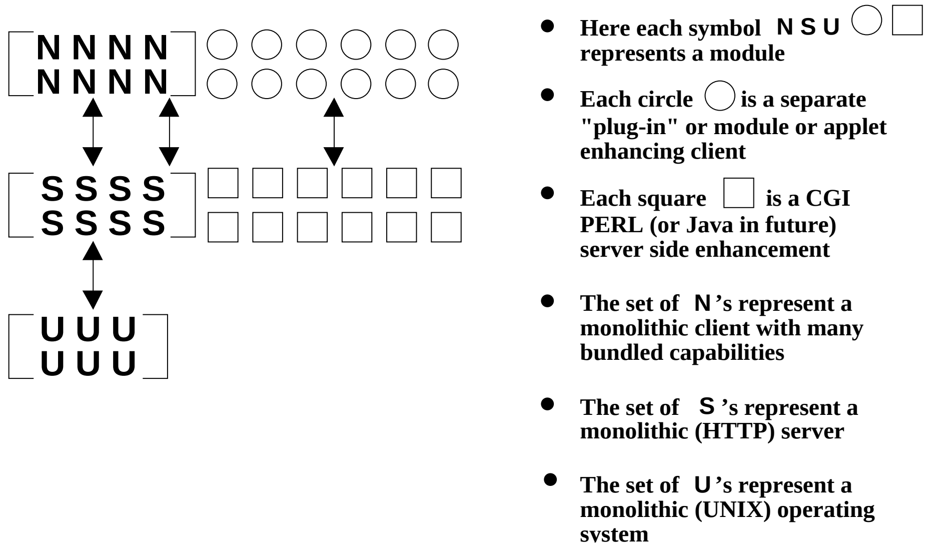


Figure 8: Architecture of Web Client-Server Software—Mosaic/Netscape Today

most servers! Already, the Web links some 50,000 servers together, 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 these days. This implements the democratic Web described in Section 1.1 where all of us can be consumers and producers of information for the Web. 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. Again, Web linked relational or object databases (Figure 7), 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. 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 architecture is shown in the comparison of Figure 8 and Figure 9 where the latter also illustrates the blurring of the distinction between (Web) servers and clients.

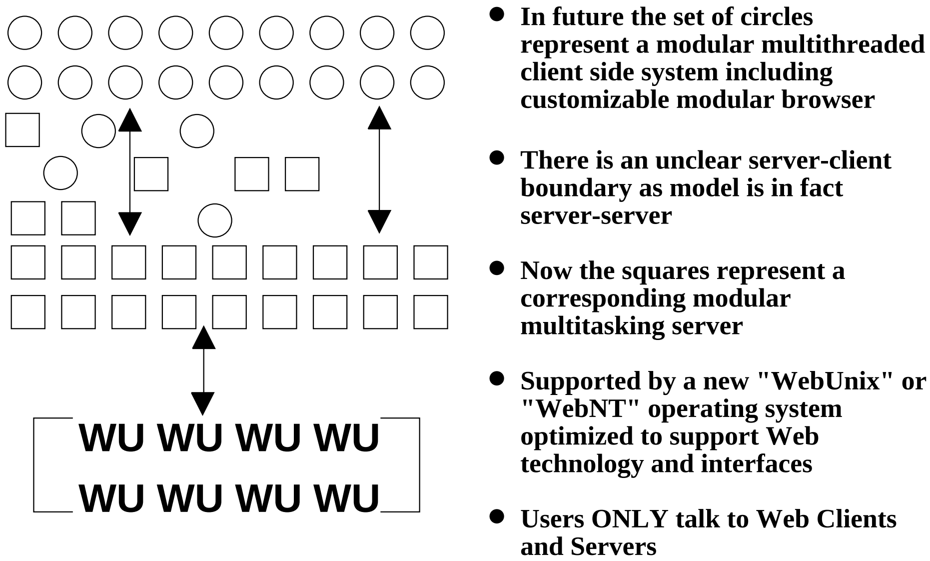


Figure 9: Architecture of Web Client-Server Software—The Future as suggested by Hotjava?

3 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 10. "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 [Fox:95c] built recursively from services and "sub-applications." Thus, there is a grey fuzzy line distinguishing services and applications. We now elaborate five possible NII services.

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

  We have already discussed these "WebWord"-like products.

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

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

- **Commerce**—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 and classified (military) data.

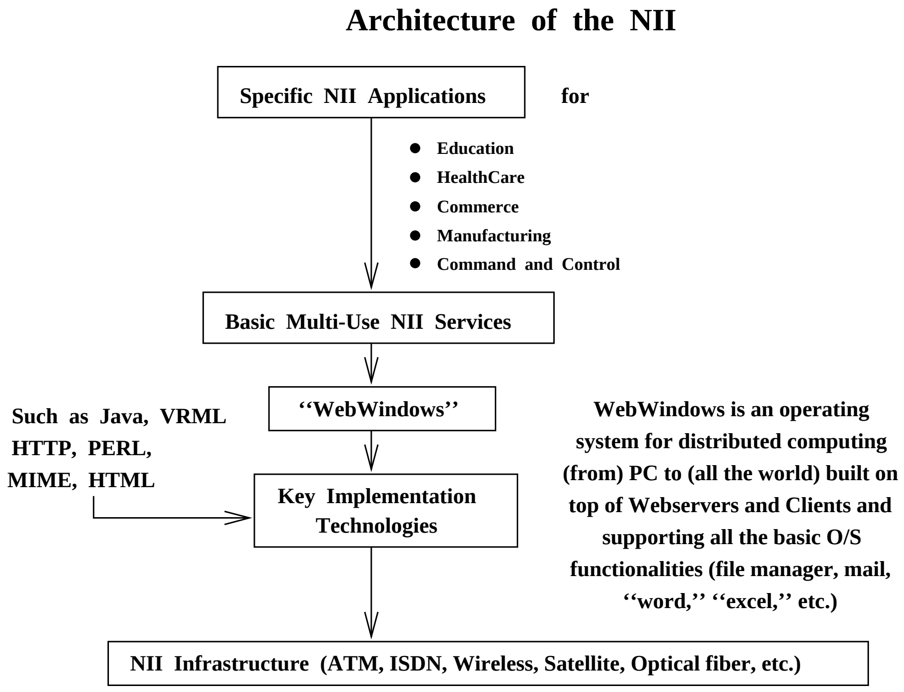


Figure 10: A layered view of Web (NII) software building applications on top of generic services that are in turn built on pervasive technologies.

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

  This includes desktop video conferencing, three-dimensional graphics MOOs, geographically distributed CAVEs leading to full televirtual interactions. The 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 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].

  This service, discussed in Section 4 can be used to control remote medical and scientific instruments; search the world for information; simulate the expected weather, or link computers in different companies for a multi-disciplinary optimization of a new vehicle.

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 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, as reviewed in Section 4, a major extension of the whole Web paradigm, currently still focused on static page services, but already gradually expanding towards computation and interactive simulation via technologies such as Java and VRML.

3.1 Four Typical Web Windows Application Areas

IntraNets and Business Enterprise Information Systems

Intranets have been popular recently. These correspond to building enterprise information systems using WebWindows technology. 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). 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 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 to the University. 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.

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 [http://www.ponfar.com/i3] 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.

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.

Education on the Web

As well as developing WebWindows products to support education, we also are teaching students about these new Web technologies. Not unexpectedly, there is currently tremendous demand both for our students and for our tutorials on the Web technologies from enterprises wanting to get into these areas. Amusingly, we can bootstrap this process and use Web technologies to put all our courses and laboratory material online. Thus, the Web community can use the Web itself to both collaborate, teach, and learn.

Our prototype (and evolving Netscape 1.1, Hotjava, and now Netscape 2.0) WebFoil system now stores over 160 separate presentations, and 6,500 "foils" with quite complete courses available in practical information, and simulation HPCC technologies. We see this as quite quickly evolving to a virtual university. Already, we are offering a small scale "Graduate Certificate in Computational Science" (focused on MPI, HPF, PERL, and Java with applications) remotely to Harbin University in China. Combining these online resources with the customized virtual reality interfaces mentioned above seems to allow remarkable new ways of delivering education to the disabled.

4 HPCC and MPP Programming Environments

We originally proposed the bottom-up approach to HPCC (Figure 1) in WebWork—a collaboration between NPAC, Boston University, and Cooperating Systems. We successfully demonstrated the concept in the use of compute enhanced Web servers, which were used to complete the factoring (http://www.npac.syr.edu/factoring.html) of the largest number (RSA130) ever tackled. Other prototypes [Furmanski:96a] include a Web interface to HPF, and a Java user interface for building compute networks and mapping general distributed software systems (http://www.npac.syr.edu/projects/webbasedhpcc/index.html).

We expect that all HPCC environments can and should be built with Web technologies. This will allow use of WebTop tools to provide a much better software engineering environment. A nice example of this is a class project of Kemal Ispirli, which showed how Java client applets can animate (in real-time) performance data fed from the Web servers controlling the compute Web. This illustrates the advantages of the Web approach, which links MPP computing to the powerful Web information capabilities. The other key feature of the WebWork approach is the use of a pervasive technology base. No longer does a relatively small HPCC community have to develop the full software environment. We can build on the distributed computing capabilities of the Web and "just" add the special synchronization, latency hiding, and coordination needs of parallel processing.

There are two particularly clear initial Web based computing projects. The first, MetaWeb generalizes the RSA130 application to full support of cluster computing with a Web technology base. MetaWeb is the natural basis for Web implementation of a distributed computing environment, as envisaged in current NSF resolicitation of their supercomputer centers.

We found in the factoring problem that a major problem with our CGI enhanced Web servers that supported RSA130 sieving was that they did not provide the standard support which one expects from clustered computing packages (http://www.npac.syr.edu/techreports/hypertext/sccs-748/index.html). This includes load balancing, fault tolerance, process management, automatic minimization of job impact on user workstations, security, accounting etc., which we will incorporate into MetaWeb. Note two immediate examples of advantages of Web based approach. We automatically support all platforms including Windows, as well as UNIX (fully heterogeneous systems), and we naturally can use "real" databases (such as DB2/Oracle) as these have already been excellently linked to Web, as shown in Figure 7. Here, we expect to directly store all system and job information in the relational (object if you use databases such as Illustra) database. Initally, we expect to use CGI scripts linking to existing Perl or C modules. Eventually, one would expect to migrate to a full Java based system. A further important feature of our proposed system is the natural linkage of performance visualization. We intend to build on idea of Ispirli to link Pablo (http://www-pablo.cs.uiuc.edu, [Reed:93b]) to the Web compute servers so that one can both store the performance trace in the associated databases, and display them either offline or in real time using Java applets.

The RAS130, and other MetaWeb applications, are natural for current Web technology because they do not require high bandwidth or low-latency communication. Eventually, we should build a full parallel environment using Web technology, but note first another programming paradigm that is also insensitive to deficiencies in the current internet, and high Web software overheads. Thus, our second initial Web computing system is WebFlow, which supports coarse grain software integration in the dataflow model. The concept of coarse grain data-flow based computations is well known with Khoros and AVS (Advanced Visualization System by AVS, Inc.) being popular. Using a convenient graphical interface, the user may interactively select precompiled modules and establish data dependences between them, and in this way define order of execution of the modules. Each module encapsulates a code to perform a specific task and communicates with the world via ports which accept or return data of one of a set of predefined types.

Many researchers have shown that AVS can be used as a general tool for supporting the dataflow computing models, which describes the top layers of many applications, such as Multidisciplinary Optimization or domain decomposition, where large grain software systems are linked in the complete application. Support for heterogenity and task parallelism in AVS allowed for a series of very successful experiments at NPAC [Cheng:93a;93c;95c] to integrate independent pieces of code running on different platforms, including MPPs into a single applications, including financial modelling and computational chemistry and electromagnetism. In fact, we believe that the dataflow paradigm can be used widely, and implement "task parallelism" in most applications where a given dataflow module can, of course, be a parallel (say HPF) program. Systems such as Hence (http://www.netlib.org/hence/index.html) and Code (http://net.cs.utexas.edu/users/code/, [Browne:95b]) also demonstrated this concept. We are currently prototyping an adaptive mesh implementation in the dataflow paradigm.

The Web naturally supports dataflow for this is essentially the model by which distributed information is accessed in the Web client-server model. Furthermore, the already established Web based framework for electronic publication can be naturally extended to support Web publication of software modules within some standardized plug-and-play interface. Thus, it is natural to design WebFlow which integrates compute and information (database, VRML visualization) services in this paradigm. WebFlow builds on the capabilities of MetaWeb to manage and allocate individual processes linked to compute enhanced Web Servers. WebFlow then adds PVM capability for communication that we have already demonstrated on our prototype Web interface to HPF (http://nova.npac.syr.edu:3000/wwvm2/). Based on lessons learned from message pasing systems such as PVM, we are now designing and prototyping the WebVM protocol for object-oriented (MIME based) Web server-to-server communication. Such WebVM architecture (with several evolving implementation options, such as base HTTP+CGI, JavaVM or JavaOS) will enable linking MetaWeb nodes towards "compute-webs," i.e., Web computational surfaces scalable from Intranets to the World-Wide Computer. The management of linked components will use a Java interface for initial network editing and dynamic display. Further, we will sup-

Table 2: Possible Components of a Future WebFlow Environment

Toolkit	Entity	Description	Technologies
WebAgent	rule	Intelligent Agents, Distributed AI	Telescript, Safe-Tcl
WebAuthor	widget	Interactive GUI Authoring	JavaScript, LiveWireWebCASE
WebCASE	applet, module	Software engineering	JDE, TDE
WebCMS	process(or)	Cluster Management	Codine, EASY, ...
WebConf	person	Conferencing/Chat	IRC, MOO, LiveMedia
WebDBMS	table/object	Relations & OO DBMS	JDBC, Oracle, Informix, DB2, ...
WebImg	image	Imaging	AVS, Khoros
WebPDA	note	Personal Digital Assistant	Netscape 2, Lotus Notes, WebTools
WebSim	avatar	Distributed Interactive Simulations	VRML+, DIS/DSI
WebVoD	stream	Streamlined Video & Audio	LiveMedia
WebWorld	3D object	3D Scene/World Design	VRML 1.0
WebHPL	array	multi-server scalable HPCC	Java, E, PCRC, HPJava
WebTVR	world	Web based televirtuality	VRML 2.0, VRML+, MOO

port a simple interpretative script that allows adaptive linkage of dataflow modules, which is needed in many applications such as the adaptive mesh refinement. The Web implementation of dataflow naturally allows full integration of VRML visualization where we would expect to use an object database to store the graphics objects produced by the applications. In the later stages, as collaborative (VRML 2.0) environments become developed, WebFlow becomes a way of implementing distributed virtual worlds as a Web implementation of the current military systems, such as DSI (Distributed Simulation Internet).

As shown in Figure 11 and Table 2, WebFlow not only provides a pervasive Web technology base for the dataflow programming paradigm, it can integrate the full range of information, compute, and other NII services with a common visual and scripted frontend. We expect such visual (and later on sensory) authoring tools to become soon increasingly popular for assembling and monitoring components of complex multi-language WebTop systems. Custom visual authoring tools will be offered for various domains and granularity levels. Fig 11 illustrates a future top level WebFlow tool for engineering large scale software systems in terms of coarse grain modules - WebToolkits. Emergent application domains include distributed medical information systems for telemedicine, military command and control, or enterprise information systems for large dynamic corporations.

5 Implications

We are seeing a paradigm shift in computing that will have profound implications for the user, software, and systems/hardware communities. Curiously, the hardware will become more uniform due to the pressure of low-cost volume production. This will allow a richer more modular and diverse software world where it is hard to see how companies like Microsoft can survive in their current form. Users have more opportunities, but there are corresponding risks and difficulties. WebWindows systems will soon dominate the Intranet enterprise systems. However, Web technology is still immature, and one cannot immediately use it in most cases.

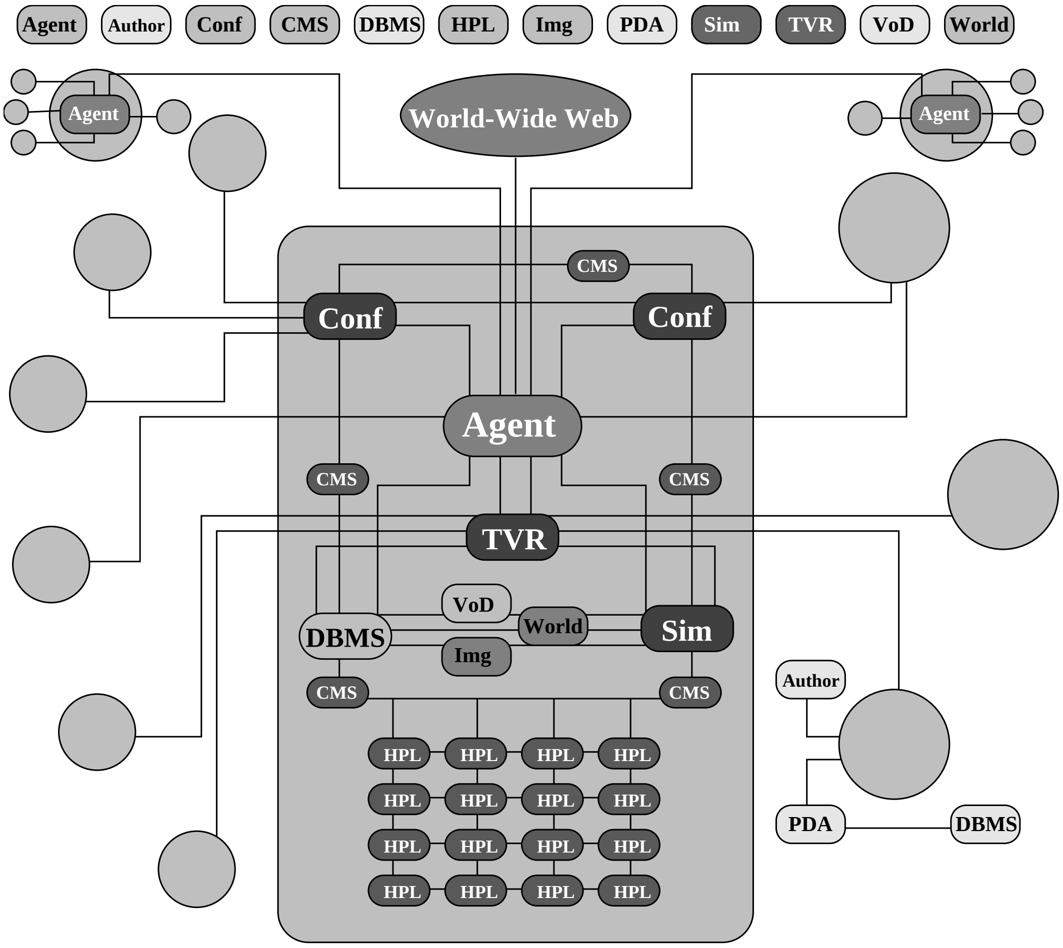


Figure 11: Example of using WebFlow for large scale software integration. Modules correspond to WebToolkits collected in Table 2.

Thus, I recommend waiting—in particular, now is a dreadful time to purchase or install a conventional Enterprise System—it will be a dated albatross quite soon. Education offers a mixed message. For the student, the situation is wonderful as Web-based curricula will allow better material delivered effectively to remote sites. Here, we see a lot of opportunities for new approaches to continuing education with "on-the-job" courses being offered on the Web by innovative universities to corporations adjusting to a charged world. On the other hand, there are longer-term implications that are not so attractive for universities—the Web-based virtual university threatens the traditional residential concept that is an essential financial foundation of most universities.

In nearly all fields, this computing resolution offers challenges and opportunities. Success will, of course, go to those who seize the correct opportunities! Perhaps this article will give you hints to the way of making wise decisions.

References

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

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

[Browne:95b] Browne, J. C., Hyder, S. I., Dongarra, J., Moore, K., and Newton, P. "Visual programming and debugging for parallel computing," *IEEE Parallel and Distributed Technology*, 3(1), Spring 1995.

[Cheng:93a] Cheng, G., Lu, Y., Fox, G. C., Mills, K., and Haupt, T. "An interactive remote visualization environment for an electromagnetic scattering simulation on a high performance computing system." Technical Report SCCS-467, Syracuse University, NPAC, Syracuse, NY, March 1993. Proceedings of Supercomputing '93, Portland, Oregon, November 15–19.

[Cheng:93c] Cheng, G., Mills, K., and Fox, G. "An interactive visualization environment for financial modeling on heterogeneous computing systems," in R. F. Sincovec, editor, *Proceedings of the 6th SIAM Conference on Parallel Processing for Scientific Computing*. SIAM, Norfolk, VA, March 1993.

[Cheng:95c] Cheng, G., Fox, G., Lin, T., and Haupt, T. "A computing framework for integrating interactive visualization in HPCC applications," *Concurrency: Practice and Experience*, November 1996. Submitted for Publication in Special Issue: Commercial and Industrial HPC Applications.

[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: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.

[Furmanski:96a] Furmanski, W., and Fox, G. "Prototyping a WebVM based WebFlow environment for distributed computing and visual programming." Technical Report SCCS-, Syracuse University, NPAC, Syracuse, NY, March.

[Reed:93b] Reed, D. A., Aydt, R. A., Noe, R. J., Roth, P. C., Shields, K. A., Schwartz, B. W., and Tavera, L. F. "Scalable performance analysis: The pablo performance analysis environment," in A. Skjellum, editor, *Proceedings of the Scalable Parallel Libraries Conference*. IEEE Computer Society Press, 1993.