WebWindows: Motivation and Application to Distributed Metacomputing

Geoffrey C. Fox and Wojtek Furmanski

Northeast Parallel Architectures Center
at Syracuse University
111 College Place
Syracuse NY 13244–4100

CRPC Presentation
March 22, 1995
Annual Review at Houston Texas

Contents

Introduction: Paradigm Shift towards Dynamic WWW

- World Wide Web is now the most promising candidate for the universal access core component of the NII.
- Current Web is ˜15,000 servers and expands at the rate of ˜1 new server / hour.
- Software industry starts adding value (Netscape, Netsite, Mosaic licences, HotMetal, Netforce, Web support in OS/2 Warp and Windows95)
- So far, Web was mainly used for static hypermedia such as local information pages, digital libraries, Internet directories etc. However, the WWW model offers also extension mechanisms (CGI, CCI) towards dynamic services and in fact arbitrary computation
- Early interactive Web services start popping. Examples include: WebCalc (NASA Goddard), Easy HTML (NCSA), WebChat (Internet Society), Virtual Doors (Ubique, Inc.), Visioneering's Imaging Machine (VRL, Inc.).

WebWindows: A Web based Internet Operating Shell

- Web browser becomes now a 'hot' desktop application. Software vendors are working on embedded Web support within conventional desktop applications.
- WebWindows is an opposite idea.
- WebWindows is an idea that the Web itself will evolve towards the leading computing paradigm integrating desktop computing and Internet services.
- Evolving Web browsers, cooperating with evolving Web servers will gradually take over the role and functionality of many other desktop applications.
- A "Virtual Microsoft", i.e. the Internet Hackers' community will collectively develop WebWindows as an open, public and modular distributed operating system for the Internet.
- These COLLECTIVE efforts will lead to phase transition taking passive Mosaic/HTTP to full blown OS of immense power

WebWindows – Implementation

- Technically, WebWindows has some aspects similar to X Windows (client-server) and NeWS/Display PostScript (interpreted). However, the Web model is higher level than X and simpler/more open than NeWS/DPS.

- Clients interprets MIME, servers interpret HTTP. Both protocols are currently very simple and extensible – MIME via new types, HTTP via new methods. These extensions is a slow process, under control of Internet RFC and STD documents.

- Additionally, both clients and servers functionality is extensible via CCI (Common Client Interface) and CGI (Common Gateway Interface) mechanisms. These extensions is a fast process, are MUCH easier and safer than changing existing browsers/servers and so useful for prototyping and experimenting with new ideas. **A revolution in Operating system Research – we can all do it!**

- CCI is still platform-dependent and not a finished design. CGI is rather stable (although only ˜1 year old!) and the base of most current dynamic extensions (including NPAC work on WebTools).

WebWindows – Milestones

- We refer to X Windows, NeWS and Display PostScript as to the first generation distributed interpretative client-server models.

- We view the current WWW as the second generation technology – it is client-multiserver and based on very simple telescripting (HTTP/MIME).

- With dynamic Web extensions, we are now starting the third generation systems which will be multiclient-multiserver (collaboratory), will include agents (such as Telescript, Safe-Tcl) and will evolve towards increased interactivity and televirtuality.

- In fact, the most attractive current MIME extension effort – VRML (Virtual Reality Markup Language) develops 3D extension of HTML in terms of some ˜40 new data structures (VRML Nodes) that will offer soon (mid '95) "3D clickable imagemaps" composed of distributed objects that are downloaded/telescripted and locally rendered in the real-time.

- WebWindows will gradually integrate these televirtual technologies emerging on the Internet with National and Grand Challenges.

WebWindows – Current Status at NPAC

- The resulting third generation phase transition expected soon on the Internet will create new markets for HPCC to provide real-time distributed simulations for the NII

- Anticipating these events, we pursue a Web software development effort at NPAC that develops dynamic WebTools for interactive content management, authoring and navigation.

- NPAC WebTools can be viewed as seeds towards WebWindows. Based on our early involvement, we expect to position NPAC as one of the focal and coordination sites in the WebWindows development process.

- Current WebTools (discussed below and demonstrated in the poster/demo session) are based on pervasive/vanilla Internet technologies: Forms based GUIs at the browser side, CGI/Perl based server side databases and computation.

- We now start integrating these base WebTools with NPAC HPCC technologies (HPF, parallel Oracle, ATM, ISDN) towards WebWindows based high quality NII services.

WebTools: Current Seeds towards WebWindows

- Today's browsers (Mosaic, Netscape) were developed from the perspective of static hypertext services and they need new design ideas to scale towards WebWindows and televirtuality. In particular, **on-board content authoring tools** are urgently needed.

- If WebWindows is to be useful for R&D and education, the current open Internet surfing needs to be augmented by a more organized and **controlled navigation tools** . Virtual Reality offers natural spatial organization metaphor and is coming quickly to the Internet via VRML. A natural strategy is therefore to add spatial dimensions to current in-time-only Internet surfing model.

- To support the concept of active spaces in such televirtual collaboratory environments, **virtual space management tools** are required so that participants, not computers are in control.

- In NPAC WebTools, we restrict ourselves to hypertext spaces and current Web technologies, and we prototype these three key concepts of the WebWindows technologies to come.

WebTools – Base Functionality: Manager, Editor, Navigator

- We call by World a navigable hierarchy of spaces (directories), forming the document tree of a WebTools server, and by HyperWorld a collection of such Worlds.

- HyperWorld Manager (WebWindows analog of File Manager) tool supports creation, destruction and manipulation (copy, move etc.) of spaces and places in the HyperWorld.

- On-line HTML editor (analog of simple default DOS or Windows editors 'write', 'e', 'edlin' etc.) supports interactive content creation on pages created by the Manager.

- HyperWorld Navigator, implemented as a Navigation Bar stamped on each page created by the Manager, supports disciplined navigation through this space.

- All WebTools are accessible from the navigation bar. The net effect is a redefined browser – a WebWindows prototype – cooperating with the HyperWorld database and activating dynamic services of the CGI extended Web server.

WebTools – Navigation Bar

- The functionality of the individual buttons is as follows:
  - [Root] – move to homepage
  - [Home] – move to top level category
  - [Out] – move out of current space
  - [Dir] – show the content of current space
  - [In] – move into the current space
  - [<=] – move one page left in the current space
  - [Self] – reproduce current page
  - [=>] – move one page right in the current space
  - [?] – search
  - [Portal] – wide jump to some remote (but related) space
  - [Bag] – grab something and carry with you
  - [Note] – make a note
  - [Edit] – edit current page
  - [Mgr] – invoke HyperWorld Manager
  - [Tool] – invoke a toolset
  - [Help] – get help

- [?], [Portal], [Bag], [Note], [Tool] and [Help] – not fully implemented yet. Trial implementation for WebMail in progress.

WebTools – Base Connectivity: WebMail

- Manager, Editor and Navigator form the base interactive toolset for the single user mode. To implement efficiently the multi-user interactions, we need a point-to-point message passing protocol. The most natural choice is RFC-222 or simply – e-mail and its MIME extensions RFC-1521 and RFC-1524.

- Indeed, these protocols are part of the current Web technologies, they are open and extensible, and also we already use them on the day-by-day basis for person-to-person electronic communication.

- WebMail makes the first step towards adopting e-mail and MIME as the base WebWindows communication models. It provides HTML+Forms interface to MH, similar to XHM or EXMH but fully Web based and of higher functionality, especially for large volume executive mail handling tasks.

- WebMail toolset is organized as a domain-specific implementation of the HyperWorld navigation model and includes: hypermedia message editor and viewer, mail database navigator, dynamic folder cabinets, and multifolder refiling capabilities.

WebTools – Planned Next Steps

- Current WebTools are organized as a collection of independent Perl4 packages (of the total size ˜20,000 lines of Perl), dynamically downloaded and interpreted upon the user requests/mouse clicks. This organization becomes inconvenient for larger scope services and we are now converting the source to the object-oriented Perl5 framework. Perl5 integrates interpreted C++, C, awk, sh, csh and works well with compiled C++ modules.

- In the planning stage are tools to support project management (calendar, notebook, scheduler) and collaboratory activities (WebMail based interactive forums). We are also exploring WebTools interfaces to agents, Web robots and relational databases.

- Finally, we are also planning the development of WebTools based CASE tools for collaboratory HPCC software engineering.

- Some of there activities, such as WebSpace, WebWork or WebHPL, currently at the level of pending proposals, are discussed below. WebHPL integrates interpreted distributed Perl5, HPF, parallel C++ and works well with compiled HPF and parallel C++.

WebSpace: WebTools for WebWindows Collaboratory

- WebWindows is an interactive multiclient-multiserver technology. When several clients connect to a single CGI extended dynamic Web server, they can share this server resources, thereby creating the notion of a virtual collaboratory space – we call it WebSpace.

- Dynamic CGI extensions by a collection of Perl packages as in current WebTools is not convenient for collaboratory environments. We plan instead to combine these packages to form another server – here we call it simply CGI server.

- CGI server maintains session internals, manages users and their databases, and interacts via the CGI protocol with the Web server.

- In a multiserver environment, WebSpace itself is distributed which raises the issue of scalable collaboratory technologies.

- We plan to address it in terms of scalable family of CGI servers, conceptually linked via the Hyperworld navigation metaphor and technically interacting via suitable agent protocols.

WebSpace – Planned Collaboration with ANL LabSpace

- We recently proposed a WebSpace to cooperate with and provide universal access Web interfaces to Argonne's LabSpace – a high-end hypermedia collaboratory environment for advanced science research.

- In this project, we plan to explore sequential and parallel CGI servers and E-Mail+MIME agents as a base computation and communication platform for scalable virtual communities.

- Eventually, each NII user/desktop will run a dynamic Web+CGI server. Next, there will be group servers, project servers, institution servers, community network servers and so on. Designing appropriate scalable agents protocols for such hierarchies is a challenging and non-trivial task. We plan to address a restricted problem of scalable WebMail support for interactive forums, ranging from individuals to domains to the LabSpace size.

- NPAC will contribute two new user domains to the scientific community of the LabSpace: education via the Living textbook project and small business via the InfoMall project.

WebWork: A WebSpace for HPCC Software Engineering

- We view WebSpace as a rather broad generic concept and collaborative technology. Specific interactions within such spaces are application/domain dependent. In the WebSpace/LabSpace model, examples include: a virtual field trip to a science lab, or a virtual corporation formed to electronically design and prototype a new product.

- The WebWork project, planned by NPAC, Boston University and Collaborative Systems Corp., will develop a WebSpace for (HPCC) software engineering.

- Large HPCC software systems, such as parallel compilers, Grand Challenge or National Challenge applications often involve geographically distributed developer teams (an extreme example is WebWindows itself which by definition involves the whole Internet).

- Current Web already facilities software exchange e.g. via NHSE. WebWork makes the next natural step towards Web based CASE tools for collaborative HPCC software engineering.

WebWork – Example from Dunn & Bradstreet

- They have communication difficulties hampering best use of their team of over 200 software engineers in Madras India
- As with other large companies, problem reports are rotated around world on 24 hour basis. If East Coast hasn't solved by COB, customer problem sent to open office the the West so that by 8 am EST, solution is waiting for customer with perhaps many offices working on it.
- We have proposed to them that local database solution is Oracle but world Wide Web linkage be in terms of WebWindows (WebSpace/WebWork).

WebWork – Collaboration with BU and CSC

- We recently proposed WebWork as Integrated Programming Environment Tools for National and Grand Challenges.

- The project addresses software development process for Integrated Challenges – a merge of Grand and National Challenges in domains such as healthcare or manufacturing that pose metaproblems hosted on world-wide metacomputers and linking all three aspects: simulation, information processing and collaboration.

- WebWork considers the world-wide metacomputer as a set of future generation WWW servers, locally/internally implemented in terms of suitable HPCC technologies (HPF, MPI, PVM, Fortran-M, pC++, CC++ etc.).

- These servers play a dual role as a publishing medium of partial solutions (or unsolved problems) and as nodes of the World-Wide Virtual Machine (WWVM) – an agents interconnected based implementation of a world-wide metacomputer.

- The WebWork project will develop CASE tools for the WWVM in a bootstrap fashion, starting from our current WebTools.

WebWork – Integrating Publication and Computation

- The technological foundation of this integration is a CGI-extended Web server that combines quality electronic hypermedia publishing with HPCC computation.
- WebWork constructs that implement it are: WWVM discussed above, VSL (Virtual Software Laboratory) and HPFCL (HPF Coordination Language).
- WWVM is like a complete software environment, including hardware (Web+CGI servers), software (locally published modules) and documentation (conventional Web pages).
- VSL is a collection of Web CASE Tools to prototype WWVM components such as: standardized software publication interfaces; Perl5 class editors and browsers; and HPFCL language bindings.
- HPFCL allows to specify and run complete applications on the WWVM. We discuss HPFCL and its WebHPL extensions later on. But first we describe our first simple WWVM pilot project – how to break security of Bank of England...

Factoring RSA Numbers and Security

- RSA security systems based on numbers:
  - $RSAm = Prime_1 * Prime_2$
  - A product of two large primes
  - RSAm has m decimal digits
  - RSA Corporation recommends m >= 200.
- Bank of England and English Savings and Loan based on m = 155 (512 binary digits)
- RSA129 cracked by factoring with email team using sophisticated version of Quadratic Sieve. RSA155 will used better Number Field Sieve.
- Need $x^2 = y^2 \text{ mod } RSAm$ and then $gcd(x+y,RSAm)$ likely to be interesting factor
- Find x and y by finding lots of interesting a's where $a = \text{product of small primes} = b^2 \text{ mod } RSAm$
- Given these a's factored into primes, multiply together so powers of primes are even. This gives desired x.
- This last step requires graph theory and solution (for Bank of England) of 5 million linear equations.

A WebWork Approach to Breaking Bank of England

- RSA155 requires about 300 teraops hours to solve with NSF.
- RSA129 needed about an order of magnitude less time. Can be done today faster if use Number Field Sieve.
- we have roughly 1–5 million independent calculations which form the rows of matrix (after clever graph theory manipulates and combines)
- Set of master servers publish problem to be solved with suitable demos, description of algorithm and full marketing attention.
  - Clients download software, run tests and certify with test case.
  - Clients pick the set of a's they hope to do in a week.
  - PC's pick the smaller numbers (need less memory and CPU)
  - Supercomputers pick the largest hardest numbers
- Clients return results – not so easy (email? cut and paste?)
- Best done as a set of cooperating servers where server performing factorization publishes its solution as a file on the WWW.
- Cooperating servers also better for computer administrations as can control set of clients at a given site.
- Initially use humans but replace by agents when software ready.

WebHPL: HPCC Infrastructure for WebWindows

- In WebWork, we also propose an interpreter of HPFCL – High Performance Fortran Coordination Language, which will support coarse grain distributed HPF computation. Compiled HPF modules, published on individual nodes of the WWVM, will be easily invoked by HPFCL scripts, integrated with GUI front-ends (such as Khoros etc.) and employed in collective computation on the WWVM.

- In WebHPL we further explore the concept of interpreted HPCC language environments and we propose an object-oriented Web based parallel programming environment supporting HPF and C++ for distributed metacomputing.

- WebHPL, or Web based High Performance Languages, is our most ambitious project in the area of Web and HPCC integration. It addresses both base software engineering and applications, and it refers both to backend and frontend layers of language compilers and interpreters, seeking a uniform programming model for interactive HPCC.

WebHPL – Integrating Compiler and Interpreter Technologies

- WebHPL is a hybrid Compiled/Interpreted environment with a more or less seamless use interface.

- We use compilers or optimized message passing in the local environments where low latency hardware benefits from careful optimization.

- We use interpreters where flexibility and power (e.g. fault tolerance) of agent approach beneficial and performance overhead in software matches intrinsic lower performance of hardware.

- We expect that our two-prong approach in WebHPL, attacking the problem both from the full compiler and the full interpreter perspective, will result in an interesting software engineering framework, 'scalable' along the compiler<—>interpreter axis.

- The interpolation mechanism along this axis is provided in terms of domain specific "little languages" that are interpreted but operate on compiled modules. UNIX shell, Perl or Tcl are examples of sequential little languages. WebHPL will develop parallel analogs.

WebHPL – HPF, HPC++ and Little Languages

- The WebHPL project specifies a software process that will result in a uniform HPF and HPC++ interpretative environment, offering a base HPCC infrastructure for WebWindows.

- We already prototyped and demonstrated at SC'93 a minimal HPF interpreter, based on NPAC F90D compiler front-end and the MOVIE interpreted agent back-end. In WebHPL, we will also address parallel and distributed C++ specification and interpreter development.

- HPC++ language model is not yet fully defined as C++ is larger, more complex language model than Fortran and there is no obvious subset to start from as it was the case with HPF.

- Therefore, rather than approaching the problem from the language perspective, we will adopt the applications perspective. Selected HPCC application domains will be analyzed and the existing parallel C++ class libraries will be inspected by the formal language analysis tools developed within WebHPL to extract domain specific "little languages" that effectively parametrize a given domain.

WebHPL – Domain/Language Specific Interpreters

- The Extractor module derives the closed type system from the Web-published class interfaces, optionally repackages the existing libraries and constructs a suitable little language interpreter for a given domain. This process is similar to the HPFCL approach described previously.

- WebHPL then automatically generates the C++ framework for a distributed interpreter of that "little language". Copies of the code are distributed over WWVM sites, compiled and linked using the native C++ environment.

- Successful little languages are published and reusable classes constructed in this process are accumulated. The idea is that this process might effectively converge to a practical HPC++ subset specification – defined simply as a union of little languages for several critical HPCC computational domains.

WebHPL – Integrating Agents with HPCC

- Our focus in the little language analysis will be on data parallel applications and coarse grain distributed computing on the WWVM. Agent languages seem to provide therefore a natural coordination and inter-site communication in this model.

- We plan to explore several currently popular agent languages such as: Telescript, Safe-Tcl, Smalltalk, KQML. We will also evaluate MovieScript protocol at NPAC (previously used as HPFI virtual assembler) as an agent language candidate.

- Specific use of agents in the WebHPL environment depends on the depth of interpretation. In the HPFCL-like little language approach, all major data structures are frozen inside the compiled modules and the relevance of agents is limited.

- On the other hand, in the extreme limit of a full interpretative HPL kernel, agents can naturally take over complexities associated with irregular decompositions, dynamic load balancing, etc. and retain clean and simple data parallel HPF and HPC++ kernels.

WebHPL – Towards Common HPL Interpreter

- In parallel with domain-specific little language interpreters, we will address the software engineering challenge of a full interpreter support, common for HPF and HPC++ language models.

- The underlying Web architecture is given by a distributed collection of CGI servers that are supported by all dynamic data structures required in the interpretation process. WebWindows offers dynamic insight into individual nodes of such a WWVM.

- We will analyze three candidates for the CGI server model: Perl5, agent interpreter such as Safe-Tcl Telescript, and MOVIE server.

- Perl5 interpreter model is interesting since it shares common aspects with a C++ subset and is likely to offer soon a collection of useful class libraries for various WebWindows functionalities.

- MovieScript was already successfully used as an interpretative kernel for HPF and it also offers dynamic object-orientation and preemptive multithreading, useful for experimentation with HPC++ specification and agents interfaces.

WebHPL – Application Domains: PDELab, Telemedicine

- We will select the following two application domains for WebHPL experiments: PDELab at Purdue and Telemedicine at CNY.

- PDELab maintains an impressive collection of solvers and grid generators, contributed by several developers and supporting various areas of scientific computing. On the other hand, the existing software can hardly cope with industry challenges, that often require automated decomposition of complex irregular domains and synchronized computation of many resulting PDE modules. We see this problem domain as very natural for coarse grained WWVM, running PDE specific subset of HPC++ and using agents for handling the system level coordination. Commercial matlab could use similar support.

- Telemedicine applications represent another example of Integrated Challenge where we plan to test our technologies. Regular HPCC components can be represented here by medical imaging, parallel rendering of a virtual human, or a PDE for virtual muscle simulation, whereas the agents component is responsible for integration, synchronization or near-real time delivery.