1. Introduction

The Worldwide Web as a front-end to the Internet is equally used by scientific users, commercial use, educational and entertainment purposes. It provides Internet users with a uniform and convenient means of accessing the wide variety of resources (pictures, text, data, sound, video) available on the Internet. Web browsers make the Internet a more user-friendly environment by integrating all those resources into a single tool that eliminates the necessity for the novice users to struggle with a steep learning curve. As commercial computing companies incorporate browsers into the window system of the computer, the worldwide web becomes an integral part of the individuals computing environment. This concept has been termed WebWindows [**WebWindows**] where applications are written to the interfaces defined by Web Servers and Clients. This concept has important implications for HPCC as the rich Web Infrastructure allows the development of much more powerful software than was previously possible. The most ambitious uses of this concepts involve new approaches to metacomputing or speculations on high performance optimizing Java compilers. Some interesting projects in these areas were described in a recent conference at Syracuse whose proceedings is available on the Web [**JavaforCSE**].

In this proposal, we are tackle a more focused problem – Web Interfaces to individual HPCC platforms and a particular application – education and training. However we stress that our technology and lessons we draw should be important for the general area of Web-based computing. In the educational area, we have seen that the teaching of Java and JavaScript is aided by the integration of these programming languages with the Web which allows integrated interactivity for both teacher delivery and self-paced student review. An important goal of the VPL (Virtual Programming Environment) is provide such capability for traditional languages – in particular for HPF, MPI and the corresponding sequential Fortran and C.

For the users of high-performance systems the need for a VPL is notable as they are in general utilizing remote facilities. Many of the tools developed for using these facilities are increasingly web-based for user-documentation, training, system monitoring and the VPL adds program development and analysis to this list of Web supported capabilities. The only requirement for the VPL is the use of a Java-enabled browser and this means it immediately gives pervasive access to HPCC facilities from a broad range of clients. Further it naturally supports an interoperable interface where the details of backend systems can be hidden from the user and as HPCC gets used by more users while there still many changes in architecture, this must be very important.

Our project has two components – development of the VPL and supporting tools themselves and secondly its ongoing testing and deployment in a set of Virtual Workshop training modules.

In summary, we propose a basic VPL tool which is a Web-based integrated, parallel programming environment comprising a visual file manager, "easy" compilation and execution of the MPI based programs written in FORTRAN and C, and for data-parallel programs written in F90 and High Performance Fortran. Further it will offer visualization capabilities for analysing data output from the program and performance information recorded in Pablo's SDDF format [**SDDF**]. We will combine and extend existing education and training material from Cornell and Syracuse including Virtual Workshop modules on HPF and data visualization, offered by Land in Cornell's CS422/522, as well as the introductory computational science class CPS-615 offered annually by Fox from which we will take material on Iterative solvers and general programming models.

We have already a good VPL prototype built by Kivanc Dincer as part of his Ph.D. thesis at Syracuse and the proposing team is already collaborating in enhancing it for use in the Cornell Virtual Workshop on HPF. In the following sections, we describe the basic VPL technology and education and training status in more detail. In Section 3, we define the proposed next steps in this two year proposal. In Section 4, we summarize the successful existing collaboration between Cornell and Syracuse in the areas of content and technology for Web-based education.

2. Background Information

2.1 Technical Background

Here we describe a prototype of the Virtual Programming Laboratory which has been developed at NPAC by Kivanc Dincer ([**Dincer 97A, Dincer 97B**]) and then further modified to run at the Cornell Theory Center in the HPF module of the CTC Virtual Workshop.

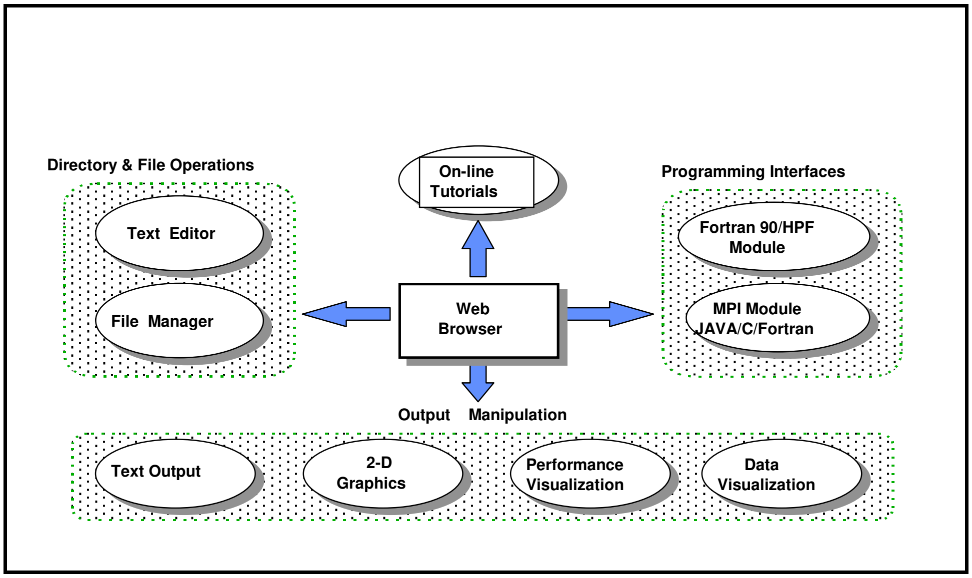

Figure 1. VPL Client-site configuration items.

The current Virtual Programming Laboratory (VPL) (Figure 1) is a Web-based, integrated, parallel programming environment consisting of a visual file manager for manipulating files and directories in a user's account, laboratory modules for compiling and executing message-passing MPI [**MPI**] programs (written in Fortran, C, and Java) and data-parallel programs (written in Fortran 90 and High Performance Fortran [**HPF**]), performance analysis and visualization subsystem to depict executed programs' performance behavior as animated or static displays, and a graphic plotting component to materialize output data as two-dimensional plots. We will briefly elaborate these capabilities in the following but note that although the basic capabilities have been tested in the fall 96 semester Syracuse CPS615 class, the more advanced visualization and security features are still immature and the intent of this proposal is to enhance and integrate them into delivered teaching and education modules. Security and some other features have been recently added at Cornell and will get their first user test in an HPF virtual workshop module.

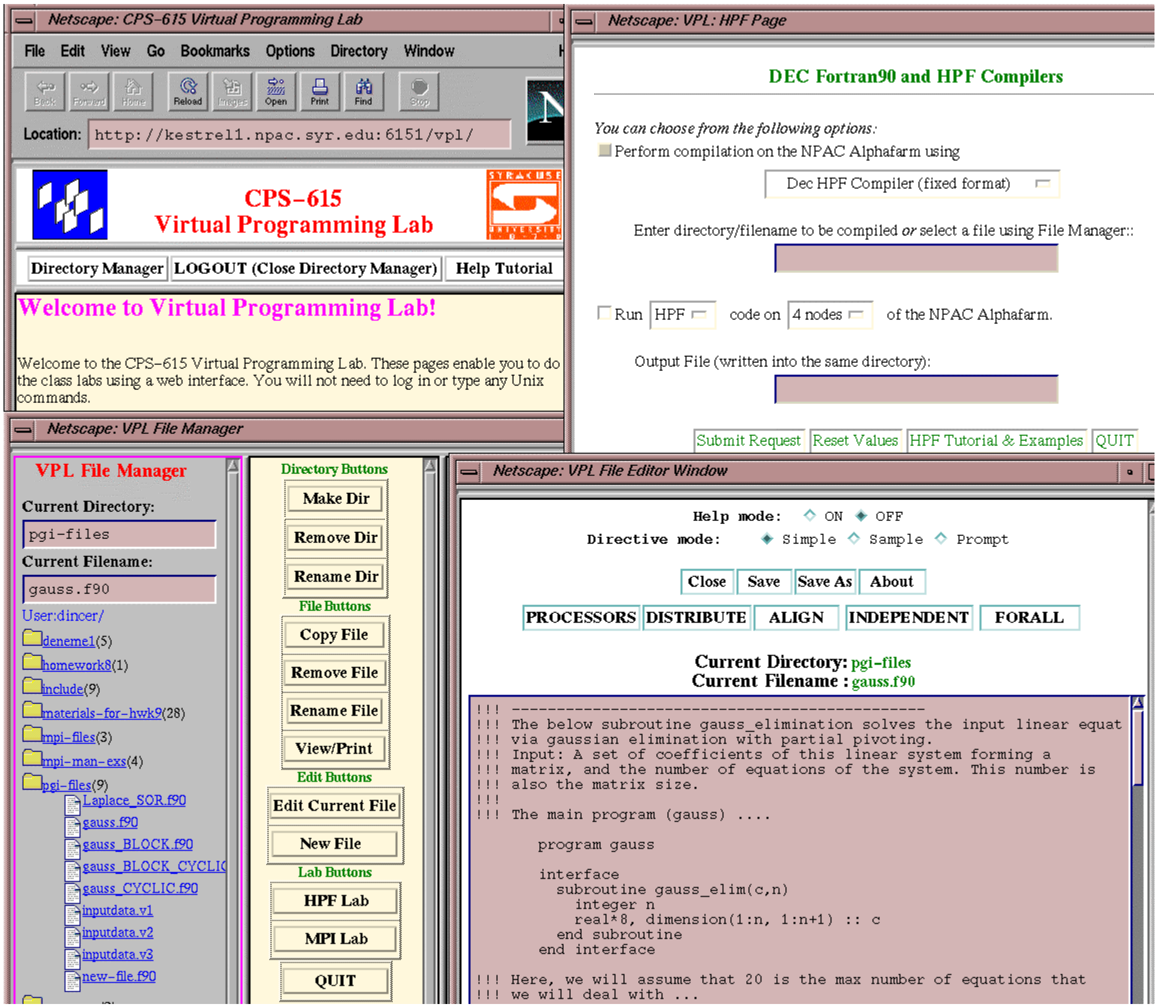

Figure 2. A snapshot of the VPL system.

The VPL uses Java and JavaScript on the client side and the standard CGI mechanisms on the server. We can expect Web technologies to continue to evolve with in particular increasing use of Java servers. We will naturally take advantage of these changes as appropriate. However the current Web allows one to set up flexible client server architectures and implement a powerful system. We note the EPIC (EPCC Interactive Courseware) [**EPIC**] system that was developed and successfully used by Edinburgh. This addresses similar goals to the VPL but is essentially a pure client side solution and so different from our project which keeps editing and visualization on the client but uses the server to run the large scale simulations and host the HPF and other compilers. WebSubmit [**WebSubmit**] system of NIST Information Technology Laboratory is another Web-based system similar to ours for submitting jobs to an IBM SP-2 load leveler. Northwest Alliance for Computational Science and Engineering uses a Web-based terminal emulator [**WebTerm**] written in Java and other Web tools for interactive teaching of UNIX and computational science and engineering from an integrated Web-based environment.

Prototype Capabilities: We briefly summarize the current capabilities of the prototype VPL. The User Interface illustrated in Figure 2 and Figure 4, is implemented using standard JavaScript for integration of client capabilities, and Java for visualization and some user interface issues not easily implemented with a JavaScript enhanced window toolkit. Two critical basic components are the *file manager* and *editing* capabilities. We choose to keep files on the server and have a set of simple capabilities to manipulate files and directories – one can view this as a web interface to the UNIX shell file manipulation utilities. These capabilities are customized to those needed to support programming with an appropriate directory structure. The prototype VPL supports two types of file editing. Originally we only used a CGI-based text editor supporting JavaScript (for rapid turnaround); however for the Cornell VW version, we are adding support for client side editing using pico, emacs and vi. Note that we can integrate with our editor, prompts and help systems to aid the user inserting particular HPF directives. The supported MPI and HPF laboratories also hide tiresome details of *makefiles* and the idiosyncrasies of particular backend parallel machines. In the architecture shown, the File Manager does not support access to the full UNIX file system which is critical for security but also aids producing a clean interoperable interface.

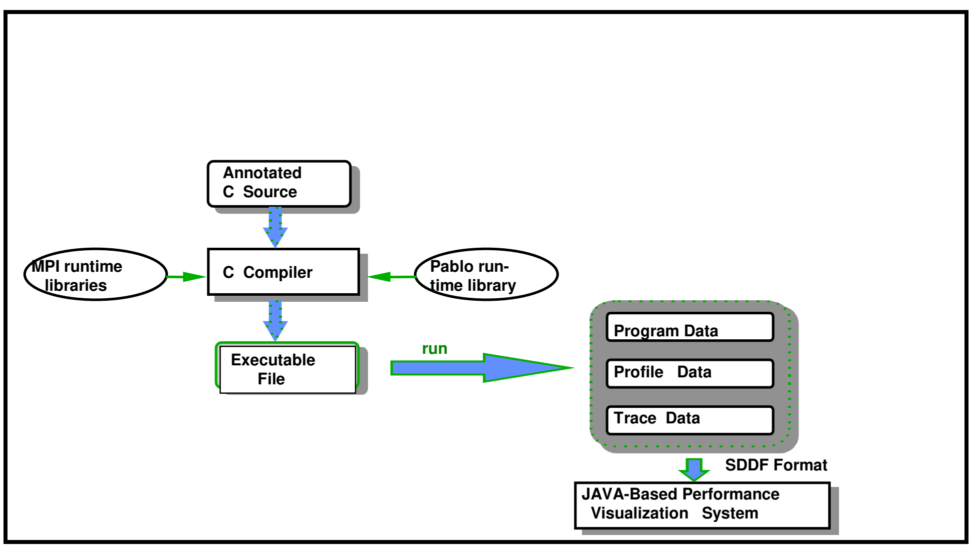

Figure 3. Preparing visualization traces for an MPI   program written in C.

We expect Java to grow in importance as a pervasive client side technology for building interactive analysis and visualization capabilities. The VPL currently has prototype capabilities for both performance and data visualization as illustrated in Figure 3 and Figure 4. The present Performance visualization capability is built around the Pablo technology developed at Illinois [**Pablo**]. This utility comprises a set of Java Applets to support the Pablo SDDF format.

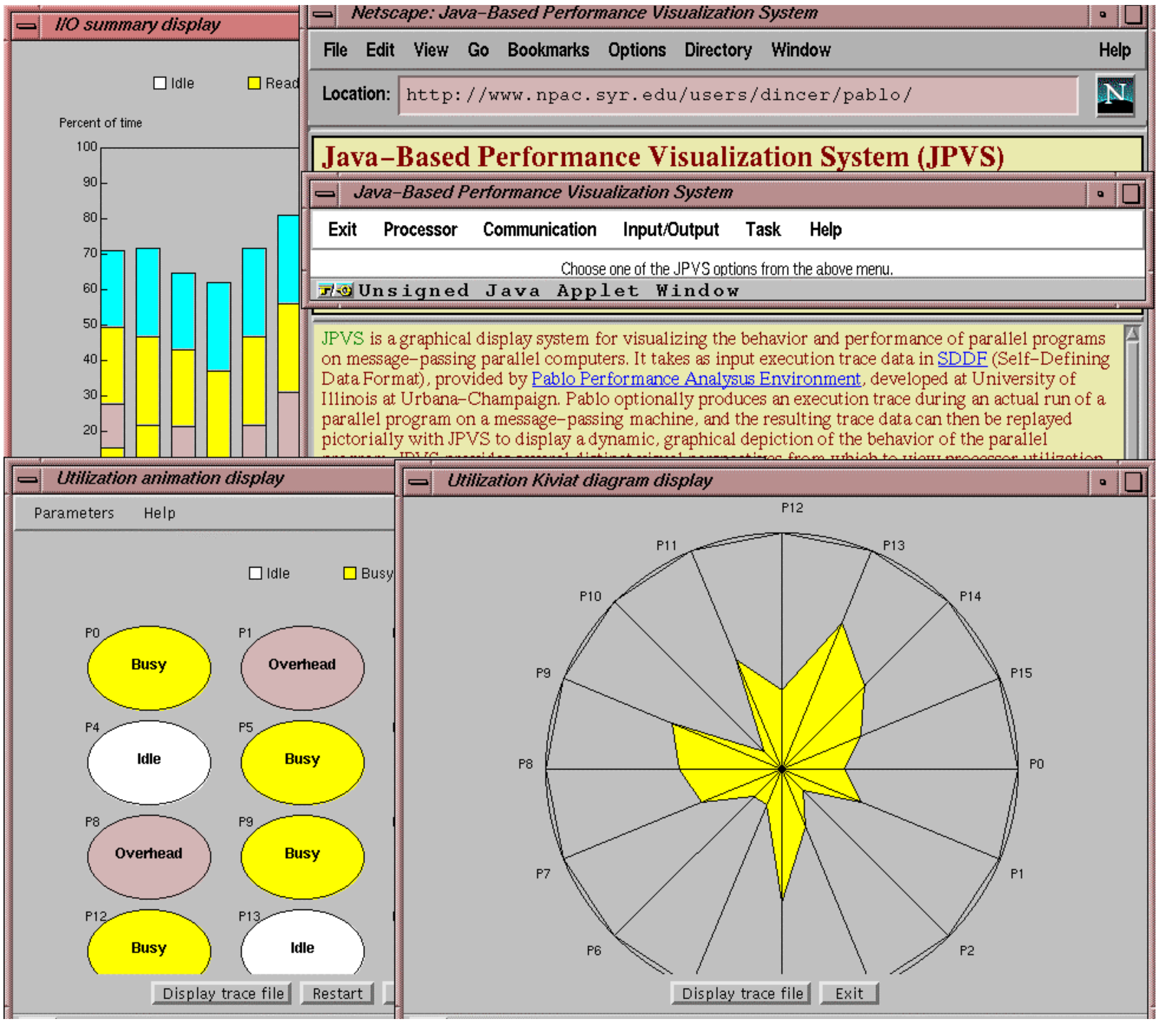

Figure 4. A snapshot of a sample JPVS session.

We anticipate a growing availability of Java Applets to support data visualization. These will be both in the form of light-weight data visualization Java applets running on the users' platform and Java wrappers to more powerful visualization tools running on the high-performance compute server. Limited versions of both these models have been developed by CTC and NPAC. The former is limited in functionality at this time, the latter is limited in the that the only visualization tool to have been integrated is IBM's DataExplore, we intend to create wrappers for VTK [**VTK**] and possibly for others tools such as MATLAB**.** A key base technology illustrated in Figure 7, is the set of wrappers which allow the client side Java applets to interact with running parallel Fortran, C , MPI and HPF codes (i.e., computational steering) and to access and analyze distributed data at runtime. Completion and testing of this capability is an important feature of this proposal. Note that the resultant architecture integrates interpreted and compiled environments; a feature we expect to be of growing importance and whose attractiveness is illustrated by the success of systems like MATLAB.

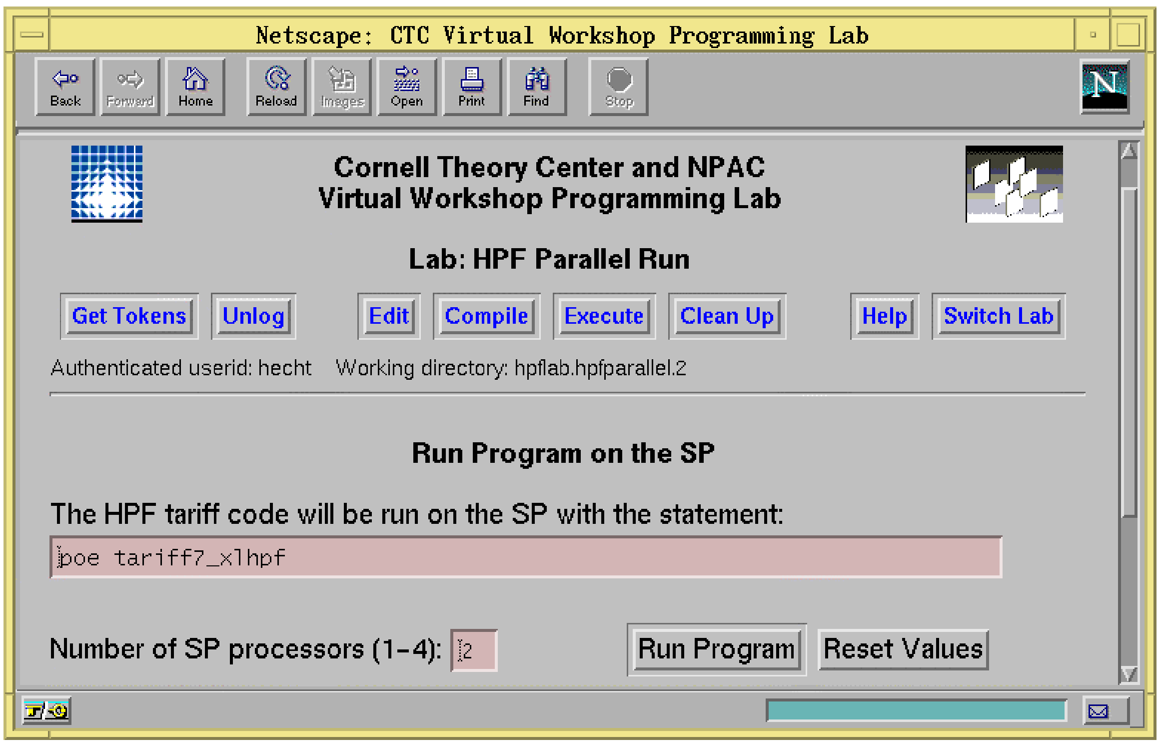

Figure 5 . A snapshot of the Virtual Workshop Programming Lab.

Prototype enhancements through CTC/NPAC collaboration: Finally we note the set of capabilities in the area of security and user interfaces which have added as result of our collaboration to produce a prototype VPL interfacing to the SP2 at the Cornell Theory Center (CTC) and used in the HPF Virtual Workshop. This reuse of the prototype VPL in a new environment encountered a new set of restrictions and requirements. These were largely due to the fact that CTC uses AFS (Andrew File System), it is important to CTC to have a secure interface and once which provided accountability of the resource utilization. Although the access control facility used at NPAC would have allowed a subset of the CTC requirements it could not meet all of them.

These modifications and enhancements included simple changes such as:

- Adding specific information regarding the exercises for which the VPL was to be used.
- Setting environment variables to those required for the CTC system.
- Adding client side editor choices of vi, emacs, or pico.
- Simple changes in the interface to integrate more readily with the VW interfaces.
- Adding information about VPL and some simple facilities such as "clean up directories" button so a user could clean up the files that are automatically generated for him or her.

Figure 5 shows the CTC interface to the VPL for HPF programming. These changes are small and within the range expected to integrate such an environment into a specific set of laboratory materials. However the major VPL changes were those required to fulfill the CTC needs for user accounting and AFS authentication on the CTC system. In order to ensure that users could access their files in AFS and that the usual usage accounting was recorded we changed the architecture to integrate with the CTC secure authentication system. Figure 6 shows the architecture of the prototype VPL as implemented at CTC. The authentication toolkit is a client/server system that allows a user to log into the CTC system using their own *userid* and *password* and allowing access to data files and software as in the regular *telnet* facility. Although the present system uses the NCSA server, the next generation authentication system will use the *Apache/Stronghold* web server resulting in transfer of the user information and data in a more secure fashion than the usual *telnet*.

We have learned a great deal through this experience of integrating the prototype the VPL into a new environment. It has helped to focus our attention and identify those areas in need of further development in order to yield a generally portable tool. It has also helped to identify the supporting tools that are required to make the VPL a fully integrated development system.

We believe this technology could have impact in a broad variety of computing applications. The prototype has been used for educational purposes at CTC and NPAC and we will continue the developments in that framework as we believe at this time the emphasis on usability over performance fits the present day technology.

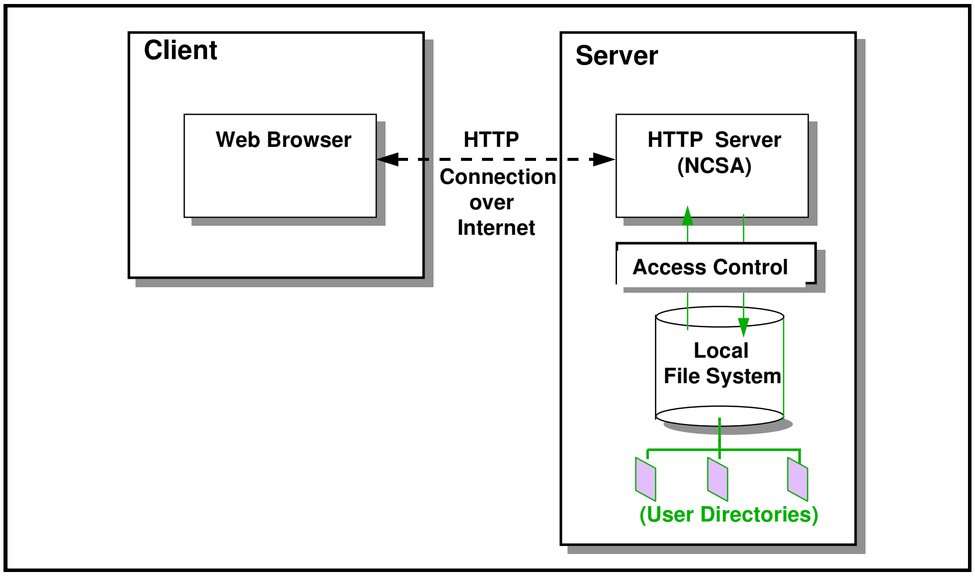

Figure 6. Client-server interaction in VPL as it is implemented in CTC/NPAC VW modules..

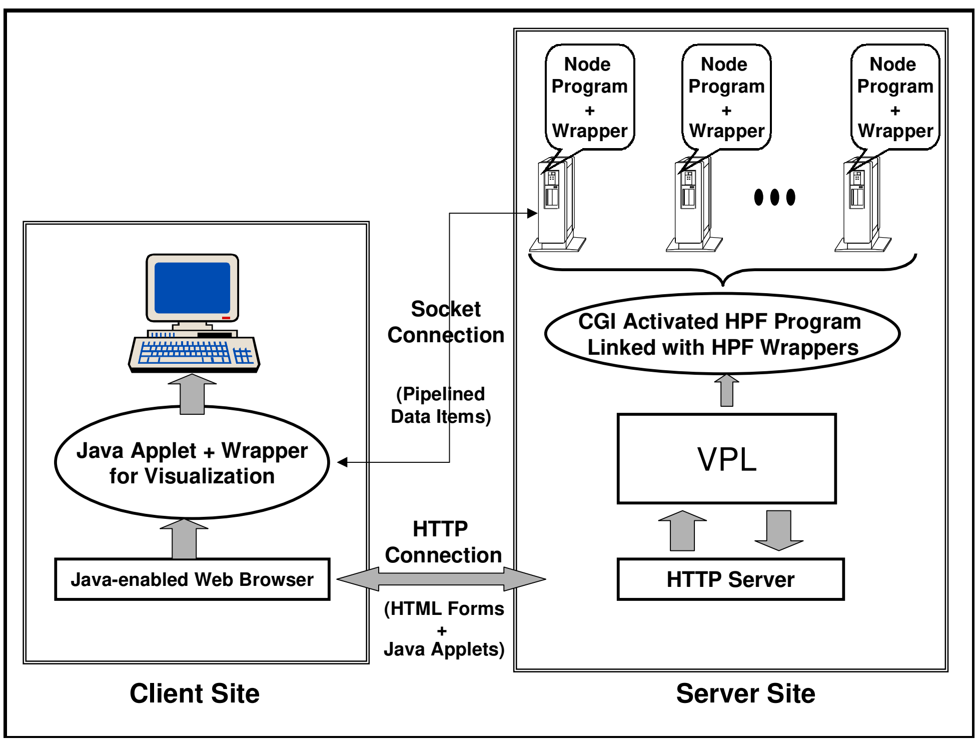

Figure 7 . Communicating data structures and interactive application steering using data wrappers.

2.2 Educational Framework

The Cornell Theory Center (CTC) is committed to providing a wide range of educational services to the national community. Established in 1985 CTC has a long history of education programs reaching audiences ranging from the youngest children to educators and school administrators, undergraduates and graduates. CTC Education and Training programs are intended to make high performance parallel computing accessible to a wide variety of potential users. These users range from researchers who are actively engaged in the exploration of scientific problems to elementary and secondary school teachers and students. To reach a target audience this diverse requires a broad, deep and flexible set of educational programs. Not surprisingly, CTC's Education and Training staff have turned to computers and the World Wide Web, to help users get the right material and training when they most need it.

The Virtual Workshops (VW) are a recent innovation in the use of electronic media as a teaching aid. These hands-on, Web-based courses at this time cover principles of parallel and distributed memory programming, the MPI and PVM message-passing libraries and High Performance FORTRAN. The VW is an entirely over-the-network program, all of the modules are built on web-based technologies and structured to allow easy traversal through a variety of levels of detail and complexity. The VW provides access to CTC's educational programs from an Internet connected machine, while retaining interactive features of a live workshop. The course components are on-line lectures and exercises, communication with CTC consultants and other participants, and logins on CTC's world-class IBM RS/6000 SP2 system. Communication between consultants and participants can take a number of forms: email, web-chat, and a VWMOO, a text-based, multi-user, collaborative object-oriented environment. Evaluation forms of the course material are an integral part of the web-based materials and were submitted as a form by the push of a virtual button. To date over 600 researchers have participated in Virtual Workshops.

The modules of the VW have been developed in a multi-layer fashion with an overview layer linking to more detailed layers. The material is represented in a variety of fashions, including text, graphical interpretations, animations and in some cases video and audio. We provide simulations in which the user can alter parameters and experiment; multiple choice quizzes which are automatically scored and mailed into CTC. With such a large number of participants the VW benefits from a lot of feedback and valuable evaluations of the content and the use of technology within the modules. This has not only allowed us to refine and polish the existing materials but has given us experience on which to build when creating new modules.

NPAC is a research and development center focusing on high performance computing and communications. Established at Syracuse University in 1987, the center is directed by Fox, who assumed NPAC leadership in 1990. Since then the scope of NPAC has broadened to include world class computational science research and education, and most recently research and development in high performance communications with particular focus on support for the National Information Infrastructure (NII). Since 1990, Dr. Fox has led an initiative to develop Computational Science curricula throughout the university. Today undergraduate and graduate programs are available in multi-disciplinary areas such as medicine, aerospace engineering and physics. Further computer science and engineering students can take computational science courses with application specialization in either the simulation or information arenas. NPAC has also partnered with Syracuse University departments to pioneer new interactive forms of education delivered via the Internet's World-Wide Web. This includes “lifelong learning” with a professional certificate in Internet applications offered thorough Syracuse University’s department of continuing education. That also uses a VPL concept but aimed at Java, CGI and Web-linked database programming.

NPAC has significant experience in teaching at Undergraduate and Graduate level including courses ECS400/CPS616 and CPS615 and undergraduate courses Phy105/106. These cover Java JavaScript Perl VRML, Web-Database Integration and a broad range of topics in parallel computing and computational science. They involve language instruction combined with student projects. The prototype VPL has been integrated into the Fortran90 MPI and HPF language sections of the computational science course CPS615. The material for CPS615 is available on the Web in the form of audio augmented foils. They are supported by a sophisticated JavaScript and Perl based system *WebWisdom* which manages some 13000 foils in 330 foilsets. Although this represents a broader range of topics than the current CTC Virtual Workshop, the NPAC material lacks components that have been of critical value to students. We intend to augment CPS615 with VW concepts such as glossaries, multi-level descriptions and Applets as part of the proposed collaborative activity.

In the area of visualization, the Theory Center has already made significant knowledge and expertise available through the Web (http://www.tc.cornell.edu/Visualization/) [**Doerr 96, Land 93, Land 94, LMTK 96**]. Training in visualization techniques has taken place at local workshops, including a series of workshops using IBM's Data Explorer for Scientific Visualization. Bruce Land has taught CS418, a computer graphics course in the Computer Science department at Cornell University since Spring 1995, and this semester taught a section of CS422/522 using the web-based visualization modules under development. CS418 makes extensive use of the Web all of the laboratory exercises are available on the web. This allows the students to preview material, review exercises and set the pace of their own progress through the laboratory, in general. The visualization modules draw heavily on the materials and expertise that already exist. It will provide educators with on-line references, information, interactive examples, and teaching demonstrations.

The VW visualization module that is under development is based on IBM's DataExplore (DX). The current prototype visualization Virtual Workshop modules include Java applets for the design of polygonal objects and parametric surfaces. Visualization is introduced in the modules in the context of general classes of scientific data representation, as currently in use in simulation. The materials introduce the general areas of visualizing particles, fields, and human-designed objects.

- Particle visualization includes systems represented by many similar objects, such as molecules, elementary particles, sand or stars in a galaxy.
- Field visualization includes vector fields such as electromagnetic or gravitational systems, and scalar fields such as 3D medical scanning output and temperature fields.
- Visualization of human-designed objects includes the machines used in science and the geometry in which a computation is embedded (e.g. shape of a machine part undergoing stress analysis).

We believe the integration of these materials with light-weight visualization tools and other commercially available tools will serve the purpose of presenting high quality learning material in the subject of visualization as well as the use of the tools. This is the ideal forum to ensure proper integration of the tools within the VPL environment.

3. Integrated VPL and VW

3.1 Technology Enhancements

Within this project we aim to develop and deliver a generally available Virtual Programming Laboratory for MPI and HPF with further testing in educational environments. Key components of the VPL will be: compilation and execution of MPI, Fortran 90, and HPF programs; visual file manager; file editing capability; performance visualization; data visualization.

As described in Section 2 many of these facilities exist in the prototype but we realize that their functionality in the present system are too limited for a generally useable tool. Considering each key capability in turn:

Compilation and execution of MPI, Fortran 90, and HPF programs: Here we intend to create a more flexible system that is easily manipulated to meet the local system and software requirements. Extensions will include new facilities for choosing lightly-loaded computational servers for running users' programs automatically or semi-automatically, submitting batch jobs from the same interface, detecting and terminating run-away user processes, piping out generated outputs from executed programs directly to visualization components.

Visual file manager: This component of the VPL requires little further development. An automatic file uploading and downloading facility for transferring files between the client and server sites is planned. Java 1.1 should be deployed in mainstream browsers over the next 6 months and we will exploit it's drag-and-drop capability and keyboard action shortcuts to provide a more user-friendly tool for file operations.

File editing capability: In order to make this facility available to CTC users a fairly crude solution was implemented that is rather naive and will not scale well within this architecture. It is clear from user feedback that the ability to use an editor of their choice is a key component with respect to usability. Our intention is to provide an integrated development environment that is alleviating the necessity for the user to learn a lot of detailed information in order to be successful. The last thing they want to do is to learn a new editor in order to use the environment. We will provide a customized visual Java editor in addition to the currently supplied CGI-based and conventional UNIX editors.

Performance visualization: Our initial audience for this tool is the HPCC community where monitoring the performance of a code is an essential part of the development cycle. The prototype system includes the capability to visualize Pablo performance data over the Web. CTC and other HPCC facilities in general support several such tools and users have their own favorite tool. We will extend the capability to include other tools and provide an interface that allows the user to use the tool of their choice.

Data visualization: We will provide a basic set of light-weight visualization Java applets. Beyond this we will also set up interfaces to MATLAB, and VTK [**VTK]** to complement the existing DX framework. The underlying visualization application will run on a server, and the image window, with its controls would sit in a Web page. A user could then view, and manipulate the data set from the document. This would be useful for larger scale visualizations that would require a more powerful rendering engine. However, the user would still be able to handle the controls from the browser. Our aim is to use commercial visualization packages wherever possible for the visualization server. In supporting DX, MATLAB and VTK we feel we are being sufficiently general and covering the range of visualization capabilities computational scientists and engineers are most likely to have available to them.

Data wrappers. Our initial prototype implementation for visualizing replicated and distributed HPF data structures provides the base for our future work. Currently, calls to data wrapper routines are inserted manually, VPL's visual text editor will be extended to facilitate putting appropriate wrapper calls into the code. We currently use HPF's own intrinsics to access distributed data items, portable runtime primitives needs to be developed in order to collect distributed data items from all processors in message-passing codes. We also plan to develop a standard Java applet library for commonly used visualization applets. This library will be used in constructing the application-specific visualization applets in the context of the VW module, and help in interactive steering of applications through the Web interface.

3.2 Extended Integration of VPL into Education and Training

We intend to continue the testing of the enhanced VPL in a variety of arenas, both in remote training through the Virtual Workshop and locally in the classroom setting both at Syracuse and Cornell Universities. We intend also to expand the test environment by expanding the topics taught using the VPL to include modules from a broader computational science curriculum. These new modules will be based on existing material used in CPS615 at Syracuse university, these include Fortran 90, HPF and MPI programming with problem specific modules including Iterative Solvers and a set of simple PDE based application case studies. For example, this would use Cornell's HPF VW module but embed into the more general discussion in CPS615 of data parallel programming so that we get material which can either be used in targeted training (typical in Supercomputer center needs) or education. The current CPS615 Iterative Solver and PDE discussion does not use the VPL (except for student homeworks last semester) but we intend to integrate both VPL and VW techniques including Java Applets into the existing CPS615 material. This is quite complete from a curricula point of view, but generated by exporting HTML and Images from Persuasion foils. Our new approach will generate Web material that exploits its interactive opportunity. CPS615 is taught each year by Fox and so this new material will be thoroughly tested!

CTC has already developed a data visualization Virtual Workshop module. This is being used at this time in a course in the Cornell Computer Science Department CS422/522, "*Software Tools for Computational Science*". The module will be used as a remote VW training module in the summer of '97. In developing this module we have gained a great deal of experience in the restrictions imposed by latency/bandwidth imposed on visualization. We intend to create a lightweight tool to complement the VPL that will make an easier environment for such educational modules. Our goal is to provide the participant with flexible, informative, and exciting methods of interacting with the computing environment and with the instructors.

3.3 Timeline and Deliverables

We are proposing a two year project. The technology advances and integration into educational materials will progress in parallel.

Year 1: - packaging of the prototype system to be used in the CTC VW MPI modules

Year	Activity	Responsible Party
Year 1	Wrappers for MATLAB visualization and integration into VW visualization module	CTC
	Integration of VPL into CTC VW MPI modules	CTC/NPAC
	Integration of VPL into CPS615 Language modules	NPAC

	Further development and test of data wrappers	NPAC
	Integration of multiple editors into VPL	NPAC/CTC
Year 2	Wrappers for VTK visualization and integration into VW module	CTC
	Integration of VPL into, and development of web-based Iterative solver module.	CTC/NPAC
	Completion of data wrappers and integration into PDE case study application. Test with use in Student Homework assignments	NPAC/CTC

Throughout the two year period, while maintaining the above outline for development we anticipate continual development of the user interface and the compilation/execution interface tools. As new technology updates are made they will necessarily impact the user interface. The CTC Virtual Workshops will continue to be presented on a regular basis throughtout the year and the materials will be used in the on-site courses within the university departments. As we receive feedback on the contents and technology usability of the materials we will update the modules.

4. Cornell Theory Center - NPAC Collaboration

The NPAC/CTC collaboration began with the development of the VPL prototype by Kivanc Dincer and the need for such a tool at CTC. An HPF module was under development at CTC for use in the CTC Virtual Workshop, the inclusion of the VPL in that environment meant that the participants could work entirely in the web environment rather than having the browser and an X-terminal window connected to CTC, as had been the model previously. Staff at CTC and NPAC collaborated both in porting the VPL and in the development of new facilities such as multiple editors. NPAC and CTC are also jointly working on other Internet-based projects.