BioCoRE: A Collaboratory for Structural Biology

Milind Bhandarkar[1,2], Gila Budescu[2], William F. Humphrey[5],
Jesus A. Izaguirre[1,2], Sergei Izrailev[2,3], Laxmikant V. Kale[1,2],
Dorina Kosztin[2,4], Ferenc Molnar[2], James C. Phillips[2,3],
and Klaus Schulten[2,3]
[1]Department of Computer Science, [2]Beckman Institute,
and Departments of [3]Physics and [4]Chemistry
University of Illinois at Urbana-Champaign,
Urbana, IL 61801
[5]Advanced Computing Lab
Los Alamos National Laboratory
Los Alamos, NM 87545
email: kschulte@ks.uiuc.edu

Biology, Network, Software Engineering, Interactive Simulation, Collaboratory

ABSTRACT

Modern computational structural biology requires scientists to employ a wide range of tools and techniques to solve complex problems while keeping accurate records of research activities. Additional complications are introduced by the need to effectively engage in interdisciplinary collaborations with geographically dispersed colleagues. The software BioCoRE, a collaborative research environment for molecular modeling and simulations, addresses these challenges. Initial design work has led to a web-based architecture focused on four primary interface paradigms: a workbench allows diverse computational tools to be applied to the problem at hand in a consistent manner, a notebook automates recording of research activities, electronic conferences held with collaborators can be saved and replayed, and multi-author documents can be prepared in a cross-platform revision control system. When complete, it is expected that the BioCoRE meta-application will drastically reduce the effort and expense presently associated with structural biology distance collaborations.

INTRODUCTION

Structural biology investigates the molecular basis of life in its healthy and diseased states. Initially through x-ray diffraction (Perutz *et al.*, 1960; Kendrew *et al.*, 1960), later through NMR (Wuethrich, 1986), and lately also through electron microscopy (Nogales *et al.*, 1998), ever more medically relevant and larger biomolecular structures have been discovered.

The blossoming field of genomics rivals structural biology with an outpouring of genomes. The vast amount of information available and the complexity of the information units, gene sequences and protein structures, have made computers indispensable tools in biomedical research to an extent which in 1991 was characterized as a paradigm shift by W. Gilbert (Gilbert, 1991). Today about 200 different programs serve researchers well, in particular those coupled with web-based tools such as the Biology Workbench*, PROPSEARCH† (Hobohm and Sander, 1995), and BLAST ‡ (Altschul *et al.*, 1990). Programs for structural biology have been mostly single-user oriented and not integrated with web-based tools. Here we suggest to develop a "Biological Collaborative Research Environment" (BioCoRE), an integrated, web-based, and tool-oriented

*URL: `http://biology.ncsa.uiuc.edu/`
†URL: `http://www.embl-heidelberg.de/prs.html`
‡URL: `http://www.ncbi.nlm.nih.gov/BLAST/`

computing and communication system for biomolecular modeling and simulation.

Structural biology, similar to its molecular biology parent, traditionally existed in small, independent laboratories. The need for expensive instrumentation (e.g. x-ray sources), the scale of tasks such as sequencing the human genome, the sharing of large data bases, the requirement of broad expertise and of cutting-edge technology have all led to many trans-mural groups ranging from collaborations between two laboratories to national consortia. Fortunately, this development coincides with the great performance increase of the US research network so that communication, data sharing, and joint use of instrumentation and computing facilities can be realized on the necessary scale (Schooler, 1996; Kouzes *et al.*, 1996; Shortliffe *et al.*, 1996; Clutter, 1996; Finholt and Olson, 1997). The wide bandwidth network connects the structural biology community today to massively parallel powerful computers, "pumping data back to their local site for immediate visualization" (UCAID, 1997). An adequate use of this resource for analysis and modeling of increasingly complex structures or for real time interactive modeling, requires a new type of web-based modeling which exploits distributed computing as well as a unification of graphics and molecular dynamics (MD) simulation.

Collaboratory software tools

Collaborative projects in structural biology demand new software tools which can be classified into four categories:

- distributed resource utilization: transparent allocation of resources, sharing of data, information and disk space
- distributed simulation and visualization: remote simulation and visualization control through web-interfaces and interactive molecular dynamics
- analysis and postprocessing: web-based analysis tools interfaces, monitoring capabilities, reporting and publication tools
- other web-based collaboratory tools for: communication (including audio/video capabilities), mentoring, record keeping, and program repositories

Currently, no software package which encompasses all these properties exists. Implementing the aforementioned capabilities in a single application will significantly advance the research environment in structural biology.

BioCoRE design supports four basic types of activities pertinent to most research projects: utilizing a wide range of computational tools, keeping records, communicating with collaborators, and writing articles and reports. This functionality will be implemented in four main interfaces of BioCoRE, called *Workbench*, *Notebook*, *Conferences*, and *Documents* (Fig. 1).

The *Workbench* interface of BioCoRE includes features for controlling molecular modeling, simulation, and bioinformatics tools with convenient and uniform access to collaboratory data. The *Notebook* interface furnishes the tools for logging, locating, and reviewing methodology, data, results, and annotations related to the ongoing projects. Scientists are able to discuss their research in real time or time-delayed sessions via the *Conferences* interface, which spawns software for teleconferencing and synchronized visualization of shared data at distant sites. The *Documents* interface of BioCoRE supplies collaborators with a convenient front end for preparing multi-author documents for publication.

BioCoRE is a general web-based collaboratory tool that leads to accelerated development and dissemination of basic biomedical knowledge. This network-centered meta-application improves the collaboration between biomedical researchers located at either the same institution or at geographically distant places. It facilitates the transparent use of and communication between existing programs, tools, and databases. BioCoRE allows researchers to share information and resources. Scientists interact in both synchronous and asynchronous fashion with each other or with the modeling tools via a common infrastructure. BioCoRE enables scientists to initiate new collaborations through its communication interface and reduces the need for travel between the participating research groups.

EXISTING COMPONENTS OF BioCoRE

The development of BioCoRE is based on successful efforts in advanced software, novel algorithms and state-of-the-art hardware. The software developed is freely available to the biomedical community and is widely distributed. We have taken a broad approach, reflected in the modeling package MDScope (Nelson *et al.*, 1995) which includes our popular molecular graphics program VMD* (Humphrey *et al.*, 1996).

Efforts have also been invested to apply parallel computer architectures for molecular dynamics simulations. To this end the program NAMD† (Nelson *et al.*, 1996; Kalé *et al.*, 1997; Kale *et al.*, 1998; Kalé *et al.*, 1998) has been developed. NAMD includes state-of-the-art algorithms, which are applicable in both serial and parallel computing environments. NAMD has been designed as a flexible program, incorporating many options to control integration methods, force field parameters, etc. It is user friendly and is easy to modify and extend to serve goals we have not foreseen (Phillips *et al.*, 1998). This modifiability is accomplished in NAMD through extensive user and programmer documentation, and free availability of source code written with an object-oriented design. NAMD is unique in the extent to which other researchers can transform it to perform specific tasks.

VMD is a mature visualization environment for structural biology. Many research groups have adopted VMD for plain visualization, docking studies, structure refinement, trajectory analysis, and interactive simulations (Leech *et al.*, 1996). Other capabilities include enhanced flexibility in display options, extensions to the scripting language, static docking, and structure alignment. Through the enhanced Tcl-based scripting language, new analysis routines may now be written by end-users without recompilation of VMD. The scripting language allows nearly all mouse actions to be logged to a script file, edited, and replayed, making it easy to produce on-line tutorials or high-quality movies. Synchronous VMD sessions on two workstations have already been achieved. VMD is web-aware and can function as a helper application for web tools.

BioCoRE is described from two perspectives in the following sections. Section illustrates BioCoRE from the user's perspective, whereas section presents the architecture of BioCoRE from the perspective of developers and implementors. For concreteness, the locally developed programs NAMD and VMD are assumed to represent the molecular graphics and dynamics components of BioCoRE although we intend to encompass other programs, e.g., RASMOL (Sayle and Milner-White, 1995) or X-PLOR (Brünger, 1996).

*URL: `http://www.ks.uiuc.edu/Research/vmd/`

†URL: `http://www.ks.uiuc.edu/Research/namd/`

BioCoRE DESIGN AND CAPABILITIES

Collaborations between structural biology researchers, in general, focus on specific biopolymer systems, e.g., on a particular class of proteins. In the course of such investigations, researchers collect information as they carry out experiments, run simulations, perform analyses, and discuss results. Accordingly, an intuitive paradigm for collaboratory software is the *project*. Hence, BioCoRE represents the information contributed by collaborators as a project, and makes projects the fundamental units of organization. Each project contains all the documents, results, history, and communications for a specified set of collaborating scientists.

BioCoRE will be accessible via a Java-enabled web browser. The user will log into the system via a browser and will be presented with a menu containing all projects for which access is authorized. The user will then select a project, which is created and stored in a BioCoRE *collaboratory server* at a selected web site. At this point the user has access to the main BioCoRE user environment, summarized in Figure 1, which supports four main types of user activities:

- controlling computational (simulation) tools with convenient and uniform access to collaboratory data at a *Workbench*;
- entering, locating, and reviewing procedures, data, results, and annotations in a shared collaboratory *Notebook*;
- discussing ongoing research via real time or time-delayed *Conferences*;
- preparing multi-author *Documents* for publication.

The common user interface will be able to launch the *Workbench*, *Notebook*, *Conferences*, and *Documents interfaces* for the selected project. The functionality of these interfaces, as will be implemented in BioCoRE, is described below.

The Workbench Interface

A key portion of any computational structural biology project is the use of simulation, analysis, and

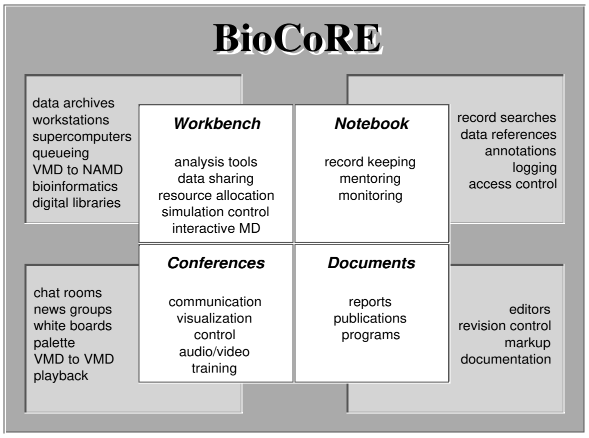

Figure 1: BioCoRE Overview; light areas describe core functions; shaded areas describe sample applications. VMD and NAMD represent any visualization and molecular dynamics simulation software such as QUANTA, Insight or Look, and X-PLOR, AMBER or CHARMm, respectively.

visualization software. Quite often it is desirable to access these programs during a collaborative session, to perform such tasks as analysis of existing data, investigations with new data or parameters, or visualization of results. The *Workbench* interface will give the user access to modeling software tools, such as the simulation or visualization programs of preference, and will support data sharing, resource allocation, simulation control, and interactive simulation. The collaboratory *Workbench* will provide the user with transparent access to all collaboratory data and computational tools without regard to where data are stored or where programs run while automating much of the log-keeping required for simulations.

As an example, consider two researchers working together online who wish to access the results of a protein simulation currently running with the *Workbench* control environment. Using the *Workbench* controls, both researchers can start separate visualization applications that will connect to a running simulation and display the same data; the two researchers may actually utilize different molecular graphics programs, e.g. VMD and Quanta* (Polygen, 1988).

Many different structural biology simulation methods, such as molecular dynamics simulations, generate large amounts of data to be managed and stored. Large files will be kept in the *collaboratory archives* that will store all the project data available in the system (aside from *Notebook* entries) and will be presented as an unified directory structure. Data added to the archive will be logged in the *Notebook*, together with comments. These *Notebook* entries will allow searching of all collaboratory data by topic without burdening the *Notebook* with large files. Data in the archive will be accessed by either manual download or automatically by *collaboratory-enabled software* or *collaboratory software wrappers*.

Third-party interactive software packages are an important tool in the modern scientific process and their use will be simplified by the collaboratory environment. Interactive software launched from within the environment will be wrapped to allow access to data in the archive which will be automatically downloaded into a special working directory, while the results will be automatically archived and logged.

BioCoRE will simplify running jobs on remote resources, as part of the *Workbench* capabilities. After selecting an available tool, such as a simulation program, the user will specify input and output files in the archive and enter configuration information. The *Workbench* will then present the user a choice of suitable resources for fulfilling the request such as the user's workstation, a remote workstation, or a dedicated compute server. The job will be executed as directed by the user and logged in the *Notebook* along with comments and accounting information.

The *Workbench* will also provide interactive steering interfaces. A primary example of this is the interface between VMD and NAMD. Selecting from a list of running jobs, the user will be presented with an appropriate monitoring and manipulation interface for the selected job. Any modifications to the progress of a simulation will be recorded in the *Notebook*, ensuring that all inputs to the simulation are saved.

Some specific tools to be incorporated into the *Workbench* include VMD, NAMD, MDTools , and BLAST and FASTA searches, or interfaces to SWISS-PROT (Appel *et al.*, 1994), PIR (Sidman *et al.*, 1988), and PDB (Bernstein *et al.*, 1977), available through the Biology Workbench.

*URL: `http://www.msi.com/info/products/QUANTA.html`

The Notebook Interface

The *Notebook* interface will provide the record keeping, mentoring, and monitoring capabilities of the collaboratory. The interactive collaboratory *Notebook* will present the user with a chronological view of the entire project. The *Notebook* will be organized into *entries*. A *Notebook* entry will comprise a series of references to *components*. Any form of data recognized by the collaboratory environment will be accepted as a component, that is handled by the *Notebook* according to its type. For example, rather than pasting a molecular coordinate file or an image in the *Notebook*, a visualization program configuration is included which will launch an appropriate viewer with a preselected representation and orientation. Component types recognized by the collaboratory environment will include text or HTML documents, images, tables of data displayed as interactive graphs, visualization program configurations displayed as a thumbnail, simulation trajectories displayed as a summary, and lists of *data references* to other entries or their components.

It will be possible to search for entries in the *Notebook* by title, author, date, subproject, topic, and type as well as component properties such as content, or entries referred to. The user will also be presented with ready access to several standard reference lists such as calendar views by day, week, and month, unread and recently visited entries, and a *palette* of references chosen by the user.

The Conferences Interface

The *Conferences* interface will meet requirements for communication, visualization control, audio/video capabilities, and training. This interface will allow a user to review ongoing *Conferences* which this user is eligible to join, or to create a new conference and invite others to join it. Upon joining a conference, the user will be presented with a menu of third-party *conference tools* which can be launched from within the environment. These tools will automatically connect the user to other conference participants. The user will also be presented with the *conference palette* which will contain references to *Notebook* entries being discussed and to the *Notebook* entry containing the ongoing *conference log*. Users may dynamically enter and leave a conference, and a conference may continue asynchronously through posting of messages and replies.

A major conferencing tool to be developed will be the ability to slave one VMD session to another such that their displays are identical. This will enable one collaborator to control visualization on other collaborators' displays during a conference. The control of the visualization can be passed from one collaborator to another.

The Documents Interface

The *Documents* interface will meet requirements for preparing reports, publications, and other forms of dissemination. These features can also aid in software development. The key requirement is multi-author revision control services. This will most likely be accomplished by wrapping the existing CVS (Concurrent Versions System) package in such a way that existing tools with CVS interfaces can be used without modification. Added value will be provided by incorporating CVS logs into the *Notebook*. Additional features will distribute and archive commentary and markups of drafts.

IMPLEMENTATION STRATEGY

The list of tools and capabilities which will be made available to the users of BioCoRE is extensive and reflects the broad range of modes in which people work together. To implement the collaboratory server and components described in Section , the following strategy will be used:

1. Within the individual components, existing packages and technology available from other research groups and commercial vendors will be employed.

2. To link these components together, and to provide flexibility in configuring and extending the environment, standardized protocols and methods for component interoperability and communication will be used.

3. A small number of custom applications and components will be implemented through adaptation of existing programs and libraries.

This strategy will yield collaboratory software that will be familiar in its interface and concepts to both users and new developers. Our goal is to integrate capabilities from a disparate set of applications and tools into a cohesive environment in which structural biologists can work more efficiently.

To support the collaborative user experience detailed in Sections and , BioCoRE sessions must be fast and convenient to join, clearly beneficial to use, and able to incorporate many different tools. For this purpose, BioCoRE will have the following general features and capabilities:

- **User Access Security.** Since BioCoRE will rely on a web-based user interface on the Internet, it is important that only authorized users can access project data. Within a project, information stored by certain users needs to be protected from access by others.

- **Fast Data Access.** When researchers are geographically far apart, network latencies and limited bandwidth can limit the usability of collaborative software. BioCoRE will provide local data caching to help alleviate this problem.

- **Diversity in Communication Methods.** Personal communication components including video, audio, and white-board conferencing will all be available within the BioCoRE interface.

- **Attaching to Running Applications.** An important part of this user environment will be the ability to access simulation applications that are currently running or starting new applications. Applications will often need to be dynamically linked together in order to share data.

- **Modifiability and Programmability.** BioCoRE will support various paradigms of collaborative activities. A framework of this type, intended to support a multitude of different types of research activities in a constantly evolving environment, must be extensible to new tools and mechanisms. Emerging trends in the paradigms of collaborative activities, availability of better software, and evolution of hardware systems should be easily supported with minimal changes required to the existing systems. Component integration using high level scripting and coordination languages is an important capability which will be part of BioCoRE.

BioCoRE will be implemented using a component approach, which has become a powerful and popular software development method in industry today. The primary components of BioCoRE correspond to the elements of the user interface as discussed in the earlier sections. The basic BioCoRE components are summarized in Figure 2.

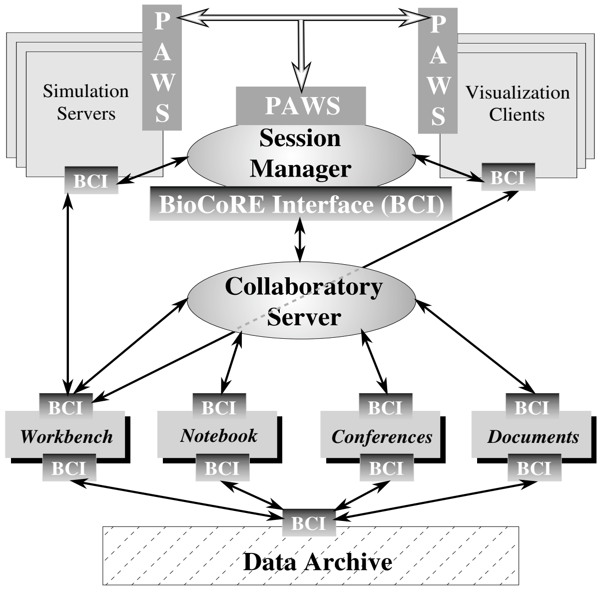

Figure 2: Components of BioCoRE. Components for the user interface and application programs interoperate through a BioCoRE component interface (BCI). Parallel clients communicate via a PAWS interface.

All of the BioCoRE components detailed below must communicate with each other and know how to initialize their working environment. For example, the user interfaces, such as *Notebook* and *Workbench* must know how to contact the collaboratory server. For these reasons, using a component model requires a standard component architecture and interface mechanism. To match the particular needs of this structural biology collaborative environment, we will first specify a BioCoRE Component Interface (BCI). The BCI will state how BioCoRE components can communicate with other components, find out information about the other components, and exchange data. Each component of BioCoRE such as the *Notebook*, *Workbench*, and any new components, will need to implement BCI. Once a component is configured to use the BCI, it will be able to interoperate with any of the other BioCoRE elements. A number of component architectures exist already, such as CORBA (CORBA, 10 December 1991), ActiveX (Denning, 1997), and JavaBeans(Arnold and Gosling, 1998; Englander, 1997), which can be used to implement the BCI and to provide the network protocol for component communication. JavaBeans is a compelling option for the BCI, since it will make it easy to implement several BioCoRE components as Java applets, servlets, or standalone Java applications in a portable manner.

Collaboratory Server

The Collaboratory Server has been previously mentioned as the point of origin for the user in any BioCoRE collaborative interaction. This server will maintain a database with information about established projects, access control, and active *sessions*. This server does not need to store all data files related to a project, since BioCoRE will also provide archival data storage components for large, persistent data elements separately.

Users will be provided with a standard web-based interface to connect to the Collaboratory Server the first time they access the BioCoRE environment. Access to the server will be performed through the BCI, and can be implemented with a simple Java applet running in a web browser. Information about available projects and active sessions must be retrieved from the server via the BCI; users must also be able to initiate new projects or sessions. We advocate to use the Lightweight Directory Access Protocol (LDAP) (Howes and Smith, 1997) to access the project and session information, that will naturally follow a directory-like organizational structure.

The collaboratory server will be implemented as a distributed and replicated collection of components to provide fast and reliable access for geographically dispersed users. Since BioCoRE will use the Internet as its backbone, security issues must be addressed by the Collaboratory Server. Techniques such as public key cryptography should be adequate for this purpose. Certificates will be used as a mechanism to identify users to the Collaboratory Server and to join a session.

Session Manager

The Collaboratory Server will only be a repository for the existing collaborative projects. When one or more researchers are actually working at a terminal within the BioCoRE environment, a *session* will exist. Multiple sessions can exist for a single project, and a user can be associated with more than one session at a time. Each session will have a Session Manager application, that will be responsible for the following tasks: (i) Keeping track of users and projects involved; (ii) Managing connections between user-interface components, data archives, applications active within a session, and collaboraory server.

Most of these activities can be accomplished through the BCI, which must be able to find and communicate with other users' BioCoRE environments. However, management of running applications is complicated for two reasons: (i) the applications may be running in parallel, yet need to link together to share data; (ii) the applications may need to run within specialized computing resources, such as a supercomputer batch queue. Accessing parallel applications as components within technologies such as CORBA (CORBA, 10 December 1991) is not feasible, since CORBA and other popular component architectures do not support the notion of parallel data structures.

To handle the problem of working with parallel applications, the Session Manager will be implemented using a modified version of the Parallel Application WorkSpace (PAWS)* controller application. PAWS is a user Application Program Interface (API) and associated controller process being developed at Los Alamos National Laboratory as part of the DOE 2000 initiative†. It provides a mechanism to allow two parallel applications, possibly running on different numbers of processors and possibly using different strategies for parallel data distribution, to link together to share data. Using this mechanism, applications such as NAMD or VMD can be made PAWS components, and can exchange data with any other PAWS component.

Within BioCoRE, we will have an important distinction between two types of components: those implementing the BCI interface, and those implementing the PAWS interface. The Session Manager will act as an overseer for several users to work on a particular project, and as a manager for the applications to be run and linked together. The Session Manager will need to implement both the BCI interface, and the PAWS interface. This will be accomplished by enhancing the existing PAWS controller to include the needed BioCoRE Session Manager capabilities (including a BCI). Resource allocation, scheduling, and user authentication mechanisms will need to be part of BCI and PAWS in order to launch the par-

*URL: `http://www.acl.lanl.gov/paws/`

†URL: `http://www.acl.lanl.gov/SciTL/`

allel applications. For this purpose, PAWS is being modified to use the Globus (Foster and Kesselman, 1998) metacomputing infrastructure in order to include compute resource allocation and user authentication support.

Application Clients

A number of components will serve as applications that users run as part of the collaboration, e.g., to visualize simulation results or to compute molecular dynamics trajectories. BioCoRE will incorporate two types of client applications: *thin* applications and *fat* applications.

Thin visualization clients will incorporate basic essential functionalities of popular molecular visualization software such as VMD and will run as applets inside a browser window. Fat clients support the PAWS interface themselves and, thus, can participate in distributed data sharing with simulation servers directly. An important distinction from the implementation point of view between the thin and fat clients is that the thin clients do not have BCI integrated within them, but are wrapped by BCI supported by the *Notebook*. Fat visualization clients such as VMD, RASMOL[‡] or Quanta can be launched and controlled by the *Workbench* in BioCoRE providing extremely flexible uses.

Simulation applications, such as NAMD, will also form an important part of the list of fat clients accessible within BioCoRE. Large computations will take place on machines designated as *compute (simulation) servers*, such as local high-performance workstations or remote supercomputers. The *Workbench* component of BioCoRE will provide the mechanism for interacting with the job-scheduling protocols on these machines through the use of PAWS controller within the BioCoRE session manager. Needed data files will be automatically retrieved from the collaboratory data archive servers. NAMD will be modified to make use of the PAWS API to participate in distributed data sharing, as will VMD and selected other fat clients. Other simulation programs such as X-PLOR (Brünger, 1996) and CHARMM (Brooks *et al.*, 1983) will be provided with BCI wrappers for incorporation into BioCoRE.

User Interfaces

Notebook, *Conferences* and *Workbench* are the primary user interface platforms of BioCoRE. They embed all the other objects and, thus, act as canvases for visual programming. They combine the capability to link the related objects to each other (hypermedia) with extensibility provided through scripts. In particular, the *Notebook* is an abstraction for a domain-specific extensible hypermedia browser. These user interface components will understand commonly used protocols and interfaces and will be enriched with applets, plug-ins and scripts to handle previously unknown types of objects. *Notebook* will be implemented as a set of the eXtensible Markup Language (XML) (Connolly, 1997) documents and Java applets that use JavaBeans as a component model to communicate with each other within standard web-browsers that support LDAP and Internet Inter-ORB Protocol (IIOP) (Orfali and Harkey, 1997). An example of electronic notebook software that can be adapted for the *Notebook* component is the DOE 2000 Electronic Notebook[§] project at Oak Ridge National Laboratory. The *Workbench* component will provide a simple visual interface to the PAWS controller capabilities within the session manager that will control connecting these PAWS clients together. This will make it possible, e.g, for the collaborating users, to connect visualization clients to running simulation clients from different remote sites in user-selectable patterns. Complex data connection networks will be possible in this environment.

Data Archive Servers

Large data sets will not be stored on the collaboratory servers, but will instead be distributed to a number of *archive servers*. Although these data sets will be managed and cataloged by the collaboratory servers, they do not need to be replicated except for temporary caching. A wide variety of machines will be capable of acting as archive servers with access methods varying from LDAP or IIOP to remote procedure calls, as determined by a mediating collaboratory server. Databases will be used as a repository of molecular structures and parameters.

CONCLUSIONS

The web-based collaboratory concept entails a new way of conducting scientific research and implies a new way of interacting with software tools and other

[‡]URL: `http://www.umass.edu/microbio/rasmol/`

[§]URL: `http://www.epm.ornl.gov/enote/`

researchers (Star and Ruhleder, 1996; Schatz, 1992). Unlike traditional research settings, the new problem-solving environment relies on computer-mediated rather than in-person interaction. Design and implementation of BioCoRE depend on a fast and efficient high bandwidth network technology as backbone for distributed web applications. Computational structural biology research involving molecular modeling and simulations, as well as bioinformatics studies will be greatly facilitated by the web-based approach.

ACKNOWLEDGEMENTS

This work was supported by the National Institutes of Health (grant: PHS 5 P41 RR05969-04). JCP was supported by a Computational Science Graduate Fellowship from the US Department of Energy.

REFERENCES

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. 1990. Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Appel, R. D., Bairoch, A., and Hochstrasser, D. F. 1994. A new generation of information retrieval tools for biologists: the example of the ExPASy WWW server. *Trends Biochem. Sci.*, **19**, 258–260.

Arnold, K., and Gosling, J. 1998. *The Java Programming Language*. Second edn. Addison-Wesley.

Bernstein, F. C., Koetzle, T. F., Williams, G. J., Meyer, E. F., Brice, M. D., Rogers, J. R., Kennard, O., Shimanouchi, T., and Tasumi, M. 1977. The Protein Data Bank: A Computer-Based Archival File for Macromolecular Structures. *J. Mol. Biol.*, **112**, 535–542.

Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., and Karplus, M. 1983. CHARMM: A Program for Macromolecular Energy Minimization, and Dynamics Calculations. *J. Comp. Chem.*, **4**, 187.

Brünger, A. T. 1996. *X-PLOR, Version 3.8: A System for X-ray Crystallography and NMR.* The Howard Hughes Medical Institute and Department of Molecular Biophysics and Biochemistry, Yale University.

Clutter, M. 1996. *Hearing on Computational Biology.* Statement of Dr. Mary Clutter, Assistant Director for Biological Sciences, National Science Foundation, before the Subcommittee on Science, Technology and Space, Committee on Commerce, Science, and Transportation, U.S. Senate September 17, 1996. http://www.nsf.gov:80/od/lpa/congress/cluttes2.htm.

Connolly, D. 1997. *XML : Principles, Tools, and Techniques.* Travelers' Tales Inc.

CORBA. 10 December 1991. *The Common Object Request Broker: Architecture and Specification (Draft).* Revision 1.1.

Denning, A. 1997. *ActiveX Controls Inside Out.* Second edn. Microsoft Press.

Englander, R. 1997. *Developing Java Beans.* O'Reilly & Associates.

Finholt, T. A., and Olson, G. M. 1997. *From Laboratories to Collaboratories: A New Organizational Form for Scientific Collaboration.* Preprint, University of Michigan, Ann Arbor, January 1997. http://www.si.umich.edu/UARC/Labtocollab.html.

Foster, I., and Kesselman, C. 1998. Globus: A Metacomputing Infrastructure Toolkit. *J. Supercomputing App.* In press.

Gilbert, W. 1991. Towards a Paradigm Shift in Biology. *Nature*, **349**, 99.

Hobohm, U., and Sander, C. 1995. A sequence property approach to searching protein databases. *J. Mol. Biol.*, **251**, 390–399.

Howes, T., and Smith, M. 1997. *LDAP : Programming Directory-Enabled Applications With Lightweight Directory Access Protocol.* Macmillan Technical Publishing.

Humphrey, W. F., Dalke, A., and Schulten, K. 1996. VMD – Visual Molecular Dynamics. *J. Mol. Graphics*, **14**, 33–38.

Kalé, L., Skeel, R., Bhandarkar, M., Brunner, R., Gursoy, A., Krawetz, N., Phillips, J., Shinozaki, A., Varadarajan, K., and Schulten, K. 1998. NAMD2: Greater Scalability for Parallel Molecular Dynamics. *J. Comp. Phys.* Submitted.

Kalé, L. V., Bhandarkar, M., Brunner, R., Krawetz, N., Phillips, J., and Shinozaki, A. 1997 (August). NAMD: A Case Study in Multilingual Parallel Programming. *Pages 367–381 of: Proc. 10th International Workshop on Languages and Compilers for Parallel Computing.*

Kale, L. V., Bhandarkar, M., and Brunner, R. 1998 (August). Load Balancing in Parallel Molecular Dynamics. *In: Fifth International Symposium on Solving Irregularly Structured Problems in Parallel.*

Kendrew, J. C., Dickerson, R. E., Strandberg, B. E., Hart, R. G., Davies, D. R., Phillips, D. C., and Shore, V. C. 1960. Structure of myoglobin: A three-dimensional Fourier synthesis at 2 Angstrom resolution. *Nature*, **185**, 422–427.

Kouzes, R. T., Myers, J. D., and Wulf, W. A. 1996. Collaboratories: Doing Science On The Internet. *IEEE Computer*, **29**(8), 40–46.

Leech, J., Prins, J., and Hermans, J. 1996. SMD: Visual Steering of Molecular Dynamics for Protein Design. *IEEE Comp. Sci. & Eng.*, **3**(4), 38–45.

Nelson, M., Humphrey, W., Gursoy, A., Dalke, A., Kalé, L., Skeel, R., Schulten, K., and Kufrin, R. 1995. MDScope – A Visual Computing Environment for Structural Biology. *Comput. Phys. Commun.*, **91**(1, 2 and 3), 111–134.

Nelson, M., Humphrey, W., Gursoy, A., Dalke, A., Kalé, L., Skeel, R. D., and Schulten, K. 1996. NAMD – A Parallel, Object-Oriented Molecular Dynamics Program. *J. Supercomputing App.*, **10**, 251–268.

Nogales, E., Wolf, S. G., and Downing, K. H. 1998. Structure of the $\alpha\beta$ tubulin dimer by electron crystallography. *Nature*, **391**, 199–206.

Orfali, R., and Harkey, D. 1997. *Client/Server Programming with Java and CORBA.* John Wiley & Sons.

Perutz, M. F., Rossmann, M. G., Cullis, A. F., Muirhead, H., Will, G., and North, A. C. T. 1960. Structure of myoglobin: A three-dimensional Fourier synthesis at 5.5 Angstrom resolution, obtained by X-ray analysis. *Nature*, **185**, 416–422.

Phillips, J. C., Brunner, R., Shinozaki, A., Bhandarkar, M., Krawetz, N., Kalé, L., Skeel, R. D., and Schulten, K. 1998. Avoiding Algorithmic Obfuscation in a Message-Driven Parallel MD Code. *In:* Deuflhard, P., Hermans, J., Leimkuhler, B., Mark, A., Skeel, R. D., and Reich, S. (eds), *Algorithms for Macromolecular Modelling.* Lecture Notes in Computational Science and Engineering. Springer-Verlag. In press.

Polygen. 1988. *Quanta.* Polygen Corporation, 200 Fifth Av., Waltham, MA 02254.

Sayle, R. A., and Milner-White, E. J. 1995. RasMol: Biomolecular graphics for all. *Trends in Biochemical Sciences*, **20**(Sept.), 374–376.

Schatz, B. 1992. Building an Electronic Community System. *Journal of Management Information Systems*, **8**, 87–107.

Schooler, E. M. 1996. Conferencing and Collaborative Computing. *Multimedia Systems*, **4**, 210–225.

Shortliffe, E. H., Barnett, O., Cimino, J. J., Greenes, R. A., and Patel, V. L. 1996. *InterMed: An Internet-Based Medical Collaboratory.* Presented at INET '96, the Annual Meeting of the Internet Society, 25-28 June 1996, Montreal, Canada.

Sidman, K. E., George, D. G., Barker, W. C., and Hunt, L. T. 1988. The protein identification resource (PIR). *Nucl. Acids Res.*, **16**, 1869–1871.

Star, S. L., and Ruhleder, K. 1996. Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research*, **7**, 111–133.

UCAID. 1997. *New national networking organization established by 112 U.S. research universities.* University Corporation for Advanced Internet Development. Press release, 1 October 1997. http://www.ucaid.edu/html/ucaidi_established.html.

Wuethrich, K. 1986. *NMR of Proteins and Nucleic Acids.* John Wiley & Sons.