A Web-based Collaboratory for Supporting Environmental Science Research

Xiaorong Xiang Yingping Huang Gregory Madey
Department of Computer Science & Engineering
University of Notre Dame
Notre Dame, IN 46556
{xxiang1, yhuang3, gmadey}@nd.edu

Steve Cabaniss
Department of Chemistry
University of New Mexico
Albuquerque, NM 67131
cabaniss@unm.edu

Abstract

A scientific collaboratory for supporting research in the field of environmental science is presented in this paper. The purpose for building this Web-based research support system is to promote collaboration among a geographically separated group of NSF sponsored scientists from different research areas and allow them to share their data and information across distributed sites. An XML-based Markup Language, NOML, is provided to build the XML-based Web components and facilitate Web services development in the future. This collaboratory also consists of a set of on-line simulators with an intelligent interface for guiding simulation configuration and various electronic communication tools. The development of this collaboratory takes advantage of the J2EE and RDBMS technologies to provide scientists a robust and flexible environment to do science on the Web.

1. Introduction

Prior research into Web-based research support systems has had two orientations: 1) system that support the individual researcher [25], and 2) collaboratories that support group of researchers.

A collaboratory is defined as "an open meta-laboratory that spans multiple geographical areas with collaborators interacting via electronic means" [11]. It is a merger of the words "collaboration" and "laboratory" first coined by Wulf (1996) [12] who was elected as President of the National Academy of Engineering in 1997. Collaboratories are intended to promote collaborations among scientists in various research areas across geographic boundaries. They allow scientists to share expensive research instruments as well as data and information stored at distributed sites, to exchange personal experiences, and to accelerate the development and dissemination of knowledge. Several collaboratories have been developed for various scientific purposes, including the Diesel Collaboratory [20] in Combustion Science, BioCoRE [2] in Biology Science, and the EMSL Collaboratory [9][16] in Environmental Science. Sonnenwald et al. (2003) [22] did an evaluation of scientific collaboratories and investigated their capabilities and disadvantages. They concluded that "there is positive potential for the development of scientific collaboratory systems."

Creating a collaboratory involves integrating existing software and hardware tools to build a seamless environment where scientists can work together virtually. A number of commercial and research tools have been developed to allow Web-based collaboration. These tools can be separated into two categories, synchronous tools and asynchronous tools. Electronic mail, discussion boards, and electronic notebooks are typical asynchronous tools, while audio/video conferencing, chat boxes, and white boards are synchronous tools. However, some collaboration tools, such as audio/video conferencing, rely on high performance hardware support. Audio/video conferencing allows scientists to see and hear each other and to communicate with each other, as if face-to-face. Although audio/video conferencing greatly supports scientific interaction, it is adopted in very few collaboratories due to the cost, hardware restrictions, low network bandwidth, and poor quality. Therefore, determining which tools are suitable for building into a collaboratory depends on whether the tools add value to the scientific interaction process or not. It also heavily depends on the requirements of end-users, the collaborators.

Besides these generic communication tools that can be integrated into the collaboratory, the software packages that scientists use to do science should be developed separately and integrated in such a way that they correspond to different scientific topics. The design and implementation of collaboratories are important to create a flexible collaboratory.

The NOM project involves a group of NSF sponsored researchers from different research areas, including chemists, ecologists, biologists, and computer scientists. As is often the case, these researchers are also geographically sepa-

rated. In order to enable scientists from different geographical areas and research areas to work together virtually, a Web-based collaboratory for supporting scientific collaborations in the NOM (Natural Organic Matter) research community is built based on the Java 2 Enterprise Edition (J2EE) [8] platform from Sun Microsystem. The NOM collaboratory provides capabilities for scientists to share computational resources (including large-scale databases and a high performance simulation cluster), analysis tools, simulation results, and data and information. It also allows scientists to communicate with each other through a wide range of tools. The NOM collaboratory also provides a XML-based Markup Language, NOML, which is used to manage molecule information and simulation configurations. Additionally, NOML is used to build the XML-based Web components to support the collaboration. Using the universal XML-based data representations enhances component reuse, enables data sharing, and facilitates Web services development. The NOM collaboratory provides several online simulation models to allow scientists to do various experiments via the Web. This collaboratory has been built, evaluated, and used by several environment research groups [1][24][4][14][13].

In this paper, an overview of the NOM collaboratory is introduced in section 2. NOML, a XML-based format for describing the molecule structure and the simulation configuration, is presented in section 3. The implementation of several collaboration components is described in section 4, and conclusions are drawn and future work are described in section 5.

2. The NOM collaboratory overview

The NOM collaboratory provides the following functions:

- Distributed computational resource utilization: Users can configure and invoke their simulations through a Web interface, and computational resources on remote sites are allocated transparently by a job manager.
- Data analysis: Users can view their simulation data, generated by the NOM simulator, from a Web-based interface. These data and information are represented in various types of graphs (bar charts, pie charts, line charts) and statistical reports by employing data query and data mining technologies.
- Information sharing: Users can share the results of their simulation, the molecule definitions, and the simulation configurations through web interfaces and a search engine.
- Data repository: Oracle databases are used to store the internal data that are generated from the NOM simulator. Additionally, external data, including publications, technical reports, and other forms of dissemination, which are uploaded by scientists, are also stored in the database.
- Secure access: Users do not have the same level of access privileges to all the tools in the NOM collaboratory. Some particular tools, such as the "Molecule validator" and the "NOM simulator", can only be accessed by authorized persons. Users have access to their own simulations, and other users can not access data which have not been authorized for public usage.
- Communication tools integration: A discussion board and a chat room are integrated into the collaboratory in order to facilitate communications between users.

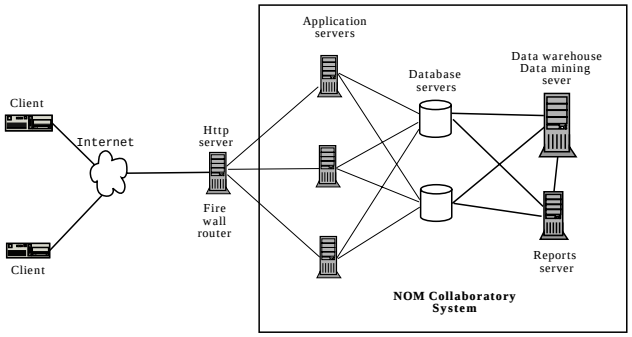

Figure 1. The architecture of NOM collaboratory

The NOM collaboratory is built upon the J2EE architecture running on a distributed cluster. This cluster has multiple dual processor PCs running Redhat Linux 8.0 and Windows 2000 operating systems. These machines in the cluster include a HTTP server, application servers, database servers, a reports server, and a data mining server. The architecture is showed as Figure 1.

The number of servers can be scaled to meet our requirements for better performance and reliability.

The implementation of the NOM collaboratory is supported by the J2EE architecture and a relational database management system (RDBMS). The Oracle RDBMS [18] was chosen for this purpose. OC4J (Oracle9iAS container for J2EE) was selected for the application servers. In order to overcome the disadvantages of traditional alternatives, such as Common Gateway Interface (CGI), for the dynamice creation of Web pages, Java Server Pages (JSPs), Servlets, JavaBeans, and Java Database Connection (JDBC) technology are applied. In order to easily maintain the programs, add new components, and reuse existing components, the design of the collaboratory follows the Model-View-Controller (MVC) design diagram [10]. Figure 2 illustrates the Web-based interfaces and components in the NOM collaboratory. Users can access these tools and ser-

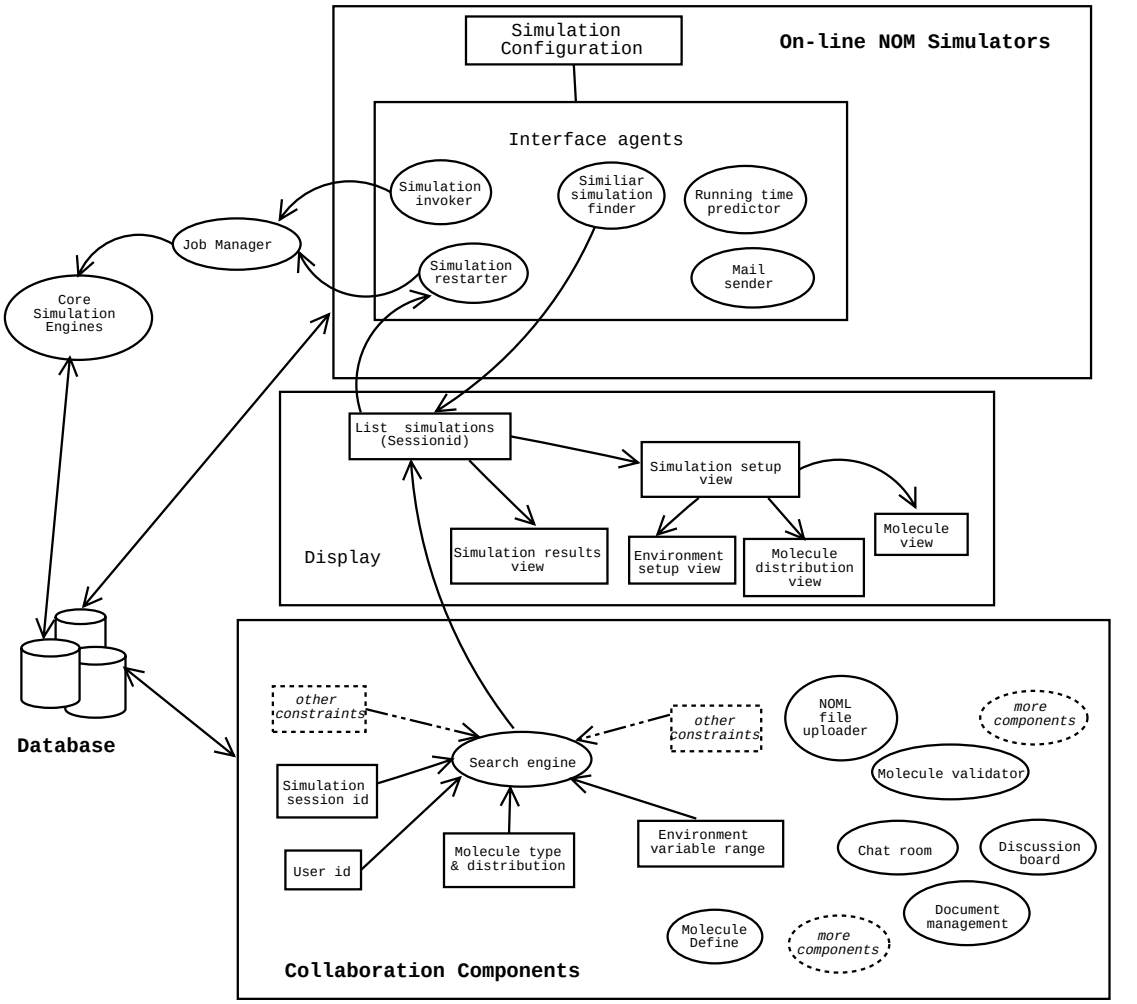

Figure 2. Components of the NOM collaboratory environment for supporting cooperation among remotely located scientists.

vices in the collaboratory environment through multiple Web-based interfaces from a standard Web browser.

The NOM collaboratory includes the following components:

- Web-based NOM simulators: NOM simulators are core parts of this collaboratory. Users can access these on-line NOM simulators through an intelligent Web interface using a standard Web browser. Several simulation models can be chosen to do their experiments. They can configure their simulations with multiple HTML and JSP pages and invoke their simulations on the remote computer cluster. The intelligent web interface provides a set of intelligent agents to find similar simulations, stop and resume simulations, predict running time of simulations, and send email to users after simulations are finished. These agents guide users to use the computational resources more efficiently. A user can submit one or more simulations and several users can submit their simulations simultaneously. A simple job manager assigns tasks on several simulation servers to achieve load balancing.

- Search engine: An ad-hoc query-based search engine provides users maximum flexibilities to search any information in the data repositories.

- NOML data format and file uploader: This Web service is built to support uploading of NOML format files. These files are parsed on the fly and the parsed data pieces are stored in the database.

- Molecule editor: Users can access the Molecule editor to define new molecule structures through a Web-based interface or upload a NOML format file.

- Molecule validator: Authorized users can access this tool from the Web interface to validate the newly cre-

ated molecule types or add new molecule types for public usage.

- Chat room and discussion board: These two collaboration tools are integrated into the collaboratory in order to facilitate the communication between users.

- Document management: This tool is integrated to exchange and disseminate information.

3. XML-based NOM Markup Language (NOML)

The XML-based model ensures uniformity across components and helps to abstract the component structure and implementation from the component interface. McLaughlin (2001)[15] and Ray (2001) [21] provide more details about XML.

The NOM collaboratory provides a set of standardized XML DTD definitions that form an ontology for the simulation configuration and the molecular structure definition. This standard format can reduce efforts of information interchange between users. Users can create XML documents, conforming to the standardized NOML DTD definitions, to describe simulation configurations and molecular structures. Once users have stored the information in NOML format files, they can attach these files into email or documents and share the data with others. Users can also upload files using the NOML uploader service that is embedded in the NOM collaboratory. Without using the Web-based configuration interface, a simulation can also be invoked after a NOML-based configuration file is uploaded. Three DTDs are defined in NOML to describe various data information.

(1) *environment.dtd* describes the format of the environment information. Users can define a set of environment parameters using tags that are defined in the DTD and used to indicate the owner as well as the accessing privilege.

(2) *molecules.dtd* describes the format of the molecular information. Within a XML file, users can define more than one new molecule structure and specify the accessing privilege.

(3) *setup.dtd* describes the format of the simulation configuration. Users can define one set of parameters, including environment parameters, molecule types, and molecule distributions, to configure a simulation.

Users can view the DTDs and download the NOML examples from the Web. The simple tree structure of the sample documents in NOML format are shown in Figure 3.

NOML is used to accomplish data exchange between users and applications. It can be extended by defining more DTDs to support various Web services.

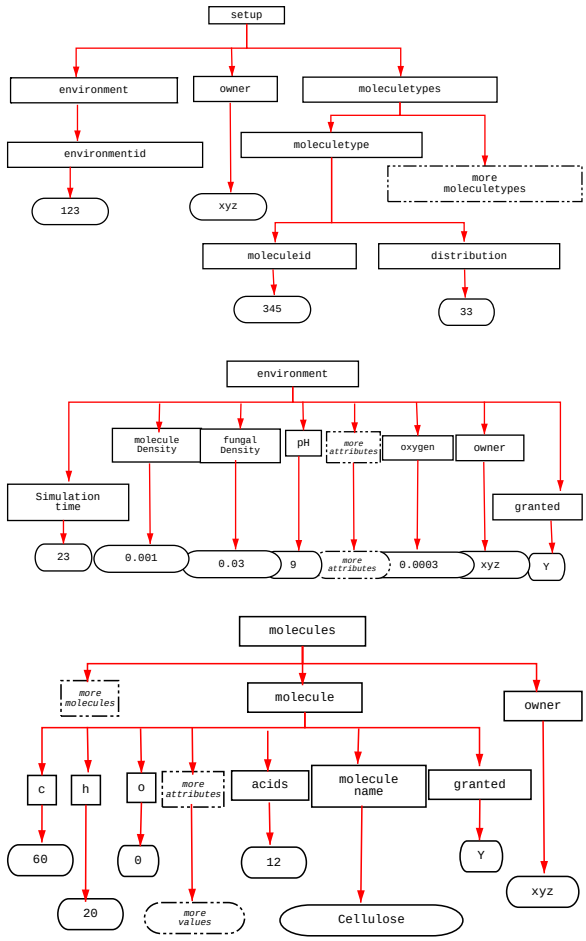

Figure 3. Tree structured view of three types of documents in NOML format. Note: not all the elements are shown in the graph.

4. Collaboratory components

All components of the NOM collaboratory are implemented within a thin-client, three-tier architecture by following the MVC design pattern. Components can be easily added, removed, and modified without changing any other features.

A client tier interacts with end users and displays information from the server to end users. In general, HTML and Java applets in a client container implement this tier. For the design of our Web interface, we use standard HTML, XML/XSL style sheets, and JSPs instead of Java applets for the performance concern.

A Web tier accepts users' responses from the client tier and generates the presentation logic. We used JSP pages for presentation logic and Servlets for session management.

An application tier handles the core scientific logic of the application. EJB components in an EJB container and Jav-

aBeans are used for implementing the application tier. In the NOM system, we used JavaBeans.

An Oracle database has been designed as the backend to store all the data input and output.

4.1. Web-based NOM Simulators

Natural organic matter (NOM) is a mixture of molecular compounds with different types of structures, compositions, functional group concentrations, molecular weights, and different degree of reactivity. NOM comes from animal and plant material in the natural environment. It exists everywhere in the world, from terrestrial ecosystems to aquatic environments. NOM plays a crucial role in ecological and bio-geochemical processes such as the evolution of soils, the transport of pollutants, and the global biochemical and geochemical cycling of elements [3]. The evolution of NOM over time from precursor molecules to mineralization is an important research area in a wide range of disciplines, including biology, geochemistry, ecology, soil science, and water resources. NOM, a prevalent constituent of natural waters, is highly reactive with mineral surfaces [3]. While NOM is transported through soil pores by water, it can be adsorbed onto or desorbed from mineral surfaces. Sorption of NOM is an important consideration in the treatment of drinking water.

NOM is present in all surface and soil waters. The amount and composition of NOM differ with climate and physical location, as well as a number of other environmental factors. NOM represents a significant fraction of the solute present in fresh water and influences all chemical and biological processes in the aquatic environment [17].

NOM, micro-organisms, and their environment form a complex system. The global phenomenon of a complex system can often be observed by simulating the dynamic behavior of individual components and their interactions in the system. Currently, in order to meet different requirments from different users, two different NOM simulators have been built into the collaboratory. Each of these NOM simulators is an agent-based stochastic model with an intelligent Web interface that can model NOM, mineral surfaces, and microbial interactions near the surface of the soil. More details on the modeling and the implementation of core simulation engines are described in Xiang(2003) [23] and Huang(2003) [7]. By simulating the behaviors of individual molecules in the system, the distributions of physical, chemical, and biological properties of NOM can be predicted. Additionally, such simulations can provide scientists in biology, geochemistry, and ecology valuable information for their studies.

The purpose of these NOM simulators is to provide scientists a testbed for their theoretical analysis and experimental results for NOM study. As the simulation system is refined, we expect that the system will help many scientists better understand the NOM complex system by providing them with information for predicting the properties of the NOM system over time.

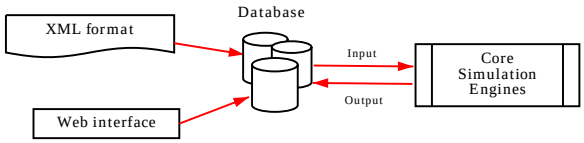

Figure 4. Simulation Input and Output

4.1.1. NOM simulation model overview The NOM simulation model includes three parts: an intelligent Web interface that assists users in configuring simulation parameters, core simulation engines that do the computations and simulate the complex behavior of a large number molecules, and a data analysis package that allows users to view their simulation results through a standard Web browser. Users can configure a simulation by inputing the simulation parameters either from the Web interface or from a NOML file. These inputs are stored in the backend Oracle database and a unique identification number is assigned to this particular simulation. The output data is stored into the database while the simulation is running. Figure 4 shows this process.

4.1.2. Intelligent agent components The Web interface provides remote users with a configuration tool. Users can specify the simulation parameters and invoke the simulation from a standard Web browser using the HTTP protocol. The intelligent configuration interface can also guide users in setting up their simulations step-by-step. Intelligent agents in the Web interface can not only provide a novice user with guidance during the configuration process, but can also provide the experienced user more information on how to better use the simulator. These intelligent agents are components based on JavaBean technology, and every component is a separate module. A summary of these intelligent agents is described as follows:

- The sending email agent

  The agent automatically sends an email to the user after the simulation is done. This email message gives the user information about the status of the simulation, either it was successfully completed or it was terminated due to any exception that was caused by hardware malfunction or software problem.

- The running time prediction agent

  A user can ask this agent to statically predict how long it will take to run the simulation according to the user's configuration inputs before the user invokes the simulation. According to the time returned by the

agent, the user can decide if the simulation configuration should be adjusted.

This agent also provides the capability for dynamically predicting the running time while the simulation is running. Users can submit a request that asks how much time remains before the finish of their simulation by simply clicking a button.

- The similar simulation finder

  Since executing a simulation is a time-consuming task, especially when the problem size is large and the running time is long, we provide users the option to take advantage of simulation results that were generated earlier by the user or by other users. A similar-simulation-finder-agent helps users find finished simulations that have some similarities with the current setup in the database. Users can either retrieve the simulation results from the database without running their simulations or submit the simulation for execution according to their own preference.

  By employing the "Euclidean distance," the distance from the current input data set to other data sets retrieved from the database can be computed. The data sets that have the smallest distance have the highest degree of similarity with the current setup.

- Automatic restarting agent

  The agent helps users extend their simulations for longer time steps by restarting their simulation from the point where it stopped. While one simulation is running, the data information about reactions is stored in the database every time step for further data analysis. In addition to this, the state of the simulation system is stored in the database at different checkpoints. The insertion and updating of the state at each checkpoint are treated as a data transaction process to guarantee data integrity.

4.2. Data analysis and visualization

One of the primary goals of this collaboratory is to provide an explanation of the simulation results, to help users better understand their data, and to predict the global properties of NOM over time. Large-scale scientific simulations take days to run and produce massive amounts of data. A user typically needs to run several variations and compare many sets of results with their experimental data. It is very difficult for users to do this with only raw data. Gray (2002) [5] shows how important new data mining algorithms are in helping scientists to access their on-line data more quickly. The Oracle9i tools are used to build the data warehousing and data mining and to incorporate them into the simulation model in order to organize, visualize, and analyze multitudinous data [6]. We enabled the display of simulation

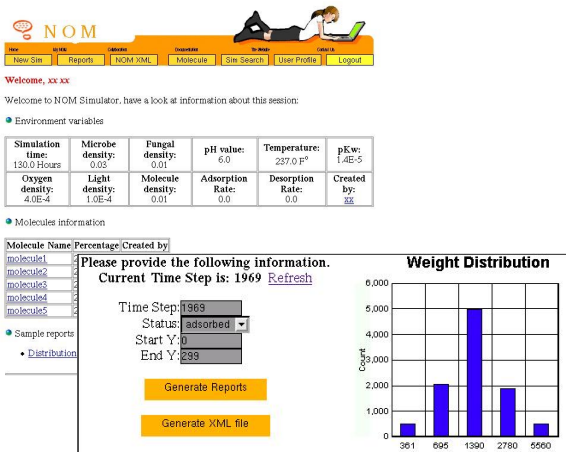

Figure 5. Two screenshots of the configuration and sample output of one simulation

results independently from the processing. Users can examine the results and download the data in XML format from the Web site during and after the simulation completes running. Figure 5 shows example of both input information and of sample output from a simulation executed by a scientist.

4.3. Search engine

Collaborative research demands sharing information and experiments to reduce the time it takes to produce new results. Executing a large-scale scientific simulation is a time-consuming task. The NOM collaboratory offers the capability of allowing users to share the configurations and results of their simulations by providing a search engine. Users can take advantage of the simulation's stored results from previous executions instead of doing the same simulation over and over again. By achieving this capability, the instruments and computational resources can be used more effectively.

Users can access the search engine through a Web interface by providing several search conditions. Users can either leave all the fields blank or make reasonable combinations and submit the request form to the server. The remote server processes the request and returns a list of simulations associated with the simulation configurations that meet with users' requests.

Users can view data information corresponding to each simulation in more depth by clicking on links. These information include the simulation configuration and result for a particular simulation. If the simulation results are valuable for users, they can extend the simulation by entering extra simulation time and resuming the simulation from the Web interface.

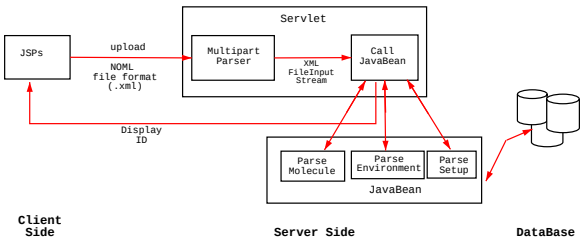

Figure 6. The design model for NOML file uploader

4.4. NOML file uploader

NOML file uploader provides an interface for users to upload a set of NOML format files. Users issue a Multipart Request by submitting a HTML form from the client side. On the server side, instead of saving files in the disk on the server, a Servlet reads incoming files and parameters and parses the incoming file's information to a XML FileInputStream. The FileInputStream is fed into a corresponding JavaBean according to the content of the incoming files. JavaBeans are implemented to validate and parse the NOML documents, establish a connection to the Oracle database, and write the data in the database. The Servlet returns corresponding ID to users for further reference. The design model is shown in Figure 6.

Users can access the NOML file uploader by providing a file name. The data validation is done by the server side JavaBeans. No information is stored to the database if the data is invalide and corresponding error messages are returned to users.

4.5. Molecule editor

Several simple rules are encoded to automatically validate the newly defined molecule in order to prevent the specification of chemically impossible molecule structures. Besides providing a Web interface for the molecule definition, users can also define new molecule types by uploading a NOML format file to the server. The uploaded file is parsed on the fly and the data is stored in the database through the NOML file uploader.

Users can also search the corresponding molecules according to the molecule name and owner of the molecules.

4.6. Molecule validator

Before one molecule type definition is ready to be shared by the NOM community, this molecule definition must be validated. However, the molecular validation is not only an objective task that can be accomplished by a piece of software, but also a subjective task that needs human involvement.

General rules that can be encoded in software are not sufficient for validating a newly created molecule. Therefore, an administrator role is provided to an authorized person who can validate newly defined molecules manually. The authorized user, a chemist in the general case, can access the validator. Only validated molecules can be shared by every user who participates in the collaboratory. Non-validated molecules can only be used by their owners.

An authorized user can also define the molecule type from the separate molecule editor or upload a NOML format file to the server. This user can remove a particular molecule from the NOM system.

4.7. Chat room and discussion board

In order to facilitate the communications between researchers, a chat room (a synchronous collaboration tool) and a threaded discussion board (an asynchronous tool) are integrated into the collaboratory. Users can enter the chat room through a Web interface by providing a user ID and password. The threaded discussion board helps users to easily trace discussion contents about one topic.

4.8. Document management

Collaborative research often demands exchange of large data files and documentation for faster dissemination of the knowledge. Users can upload research papers and data sets through a Web interface into a shared place residing on the remote servers. These files are accessible to other researchers after they are uploaded.

5. Conclusions and Future Work

In this paper, a description of a Web-based collaboratory environment that supports new ways for conducting scientific research on NOM is presented. The NOM collaboratory deploys collaborative technologies to NOM researchers geographically separated or in different disciplines. This system integrates a number of collaboration tools, including NOM simulators, a molecular validator, a molecule editor, search engine, document management, a discussion board, and a chat room. The architectural design of the NOM collaboratory makes it easy to add, modify, and remove components in the collaboratory according to the requirements of scientists.

In the future, some powerful third-party tools (such as video/audio conferencing) which rely on a fast and efficient high bandwidth network technology and advanced hardware (e.g., Internet2) may be integrated. Oracle Collaboration Suite [19] offers integrated email, voice mail, phone,

fax, scheduling, calendaring, meeting management, and file management. It could also be considered for integration into the NOM collaboratory.

An XML-based NOM Markup Language, NOML, which provides a standard definition for the molecule and simulation configuration is described. The NOML not only supports data communication between users but also give us flexibility in extending the on-line NOM simulation. It also supports building new XML-based Web services in the near future. A set of new simulation models for the NOM study will be built and integrated into the collaboratory. The communication between these models can be achieved by the extension of the NOML. The NOM collaboratory allows users from different places to collaborate by sharing their simulation data and configurations as well as by hosting discussions among the NOM community.

6. Acknowledgments

This research was supported in part by a NSF ITR Grant No. 0112820 and by the Center for Environmental Science & Technology at the University of Notre Dame. We acknowledge the contributions of Patricia Maurice and Leilani Arthurs of the Department of Civil Engineering & Geological Science at the Univeristy of Notre Dame for their respective discussions on the Web interface, input, and testing of the NOM collaboratory.

References

[1] L. Arthurs, P. Maurice, X. Xiang, G. Madey, and Y. Huang. Agent-based simulation of biocomplexity: Effects of adsorption on natural organic mobility through soils. In *American Chemical Society National Meeting*, 2003.

[2] M. Bhandarkar, G. Budescu, W. F. Humphrey, J. A. Izaguirre, S. Izrailev, L. V. Kalé, D. Kosztin, F. Molnar, J. C. Phillips, and K. Schulten. Biocore: A collaboratory for structural biology. In *Proceedings of the SCS International Conference on Web-Based Modeling and Simulation*, pages 242–251, 1999. Also available at: http://www.ks.uiuc.edu/Research/biocore.

[3] S. Cabaniss. Modeling and stochastic simulation of NOM reactions, working paper. http://www.nd.edu/ñom/papers/papers.html, July 2002.

[4] S. Cabaniss, G. Madey, P. Maurice, L. Leff, Y. Huang, X. Xiang, E. Chanowich, and O. Olapade. Stochastic synthesis model for the evolution of natural organic matter. In *225th American Chemical Society National Meeting*, 2003.

[5] J. Gray and A. Szalay. The world-wide telescope. *Communication of the ACM*, 45(11):51–55, November 2002.

[6] Y. Huang. Infrastructure, query optimization, data warehousing and datamining for scientific simulation. Master's thesis, University of Notre Dame, 2002.

[7] Y. Huang, G. Madey, X. Xaing, and E. Chanowich. Web-based molecular modeling using JAVA/SWARM, J2EE, and RDBMS Technologies. In *Seventh Annual Swarm Researchers Meeting (Swarm2003)*, 2003.

[8] Java 2 platform, enterprise edition. http://java.sun.com/j2ee/.

[9] G. C. Jr., J. Myers, and D. Hoyt. Social networks in the virtual science laboratory. *Communications of the ACM*, 45(8):87–92, August 2002. Also available at: http://collaboratory.emsl.pnl.gov/.

[10] N. Kassem and the Enterprise team. *Designing enterprise applications with the Java 2 Platform, Enterprise Edition*. Addison-Wesley, 2000.

[11] S. H. Koslow and M. F. Huerta, editors. *Electronic collaboration in science*. Lawrence Erlbaum Associates, 2000.

[12] R. T. Kouzes, J. D. Myers, and W. A. Wulf. Collaboratories: Doing science on the Internet. *IEEE Computer*, 1996.

[13] G. Madey, Y. Huang, X. Xiang, E. Chanowich, S. Cabaniss, and P. Maurice. Complex system simulation: interactions of nom molecules, mineral surfaces, and microorganism in soils. In *Workshop on Modeling Complex Systems, US Geological Survey*, 2002.

[14] G. Madey, Y. Huang, X. Xiang, E. Chanowich, S. Cabaniss, and P. Maurice. Simulation of biocomplexity: Interactions of nom, mineral surfaces, and microorganism in soils. In *ASLO 2003 Aquatic Sciences Meeting*, 2003.

[15] B. McLaughlin. *Java & XML, 2nd Edition*. O'Reilly & Associates, 2001.

[16] J. D. Myers, A. R. Chappell, M. Elder, A. Geist, and J. Schwidder. Re-integrating the research record. *Computing in Science and Engineering*, pages 44–50, May/June 2003. Also available: http://collaboratory.emsl.pnl.gov/.

[17] Natural organic matter in the nordic countries. http://www.kjemi.uio.no/envir/nominic/documents/background.html.

[18] Oracle database. http://www.oracle.com/ip/deploy/database/oracle9i/index.html.

[19] Oracle collaboration suite. http://otn.oracle.com/products/cs/content.html.

[20] C. M. Pancerella, L. A. Rahn, and C. L. Yang. The diesel combustion collaboratory: combustion researchers collaborating over the Internet. In *Proceedings of the 1999 ACM/IEEE conference on Supercomputing*. ACM Press, 1999.

[21] E. T. Ray. *Learning XML*. O'Reilly & Associates, 2001.

[22] D. H. Sonnenwald, M. C. Whitton, and K. L. Maglaughlin. Evaluating a scientific collaboratory: Results of a controlled experiment. *ACM Transactions on Computer-Human Interaction*, 10(2):150–176, June 2003.

[23] X. Xiang. Agent-based scientific applications and collaboration using java. Master's thesis, University of Notre Dame, 2003.

[24] X. Xiang, G. Madey, Y. Huang, and S. Cabaniss. A stochastic simulation of natural organic matter and microbes in the environment. In *2003 World Conference on Natural Resource Modeling*, 2003.

[25] Y. Yao. A framework for web-based research support system. In *Computer Software and Applications Conference (COMPSAC2003)*, Dallas, TX, Nov. 2003.