

A Web Portal for Environmental Research

Xiaorong Xiang • Gregory Madey • Yingping Huang • Steve Cabaniss

***Abstract.** The main purpose of existing Web portals is to promote information sharing over the Internet. In this paper we present the design and implementation of a science portal, the Natural Organic Matter research portal, which is not only for information sharing but also provides tools to allow scientists to share remote computational resources and do science on the Web. An XML-based Markup Language, NOML, is provided to build the XML-based Web components and facilitate Web services development in the future. This portal 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 portal takes advantage of the J2EE and RDBMS technologies.*

1 Introduction

Scientific portals are intended to promote information sharing and discussion among scientists over the Internet. 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. Warr describes that “The name suggests that a portal is a doorway or gateway to other places” (Warr 2001). The author also states, according to a recent survey, that Internet users see portals as destinations rather than as gateways to the Web. Several scientific portals have been developed for various scientific purposes, including the (Ecoportal 2003) for environmental sustainability, the (Microbes 2003) for microbiology information, the (Eps 2003) for earth science, and the (ESeFDee 2003) for marine sciences. All of these portals include search engines, links to other Web sites, directories, and a set of communication tools. The main purposes of these portals are to disseminate information of regulations, policies, knowledge and news.

In this paper we present the design and implementation of a science portal, the Natural Organic Matter (NOM) research portal, which is not only for information sharing but also providing tools to access remote computational resources and do science on the Web. The purpose of this portal is to build a seamless environment where scientists can work together virtually and support the collaboration between scientists. Electronic mail, discussion boards, and electronic notebooks are typical asynchronous communication tools, while audio/video conferencing, chat boxes, and white boards are synchronous communication tools (Koslow and Huerta 2000). Determining which tools are suitable for building into a portal 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 portal, the software packages that scientists use to do science are developed separately and integrated in such a way that they correspond to different scientific topics. The design and implementation of portals are important to create a flexible portal.

The NOM research portal 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 portal also provides an 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 collaboration. Using the universal XML-based data representations enhances component reuse, enables data sharing, and facilitates Web services development. The NOM portal provides several on-line simulation models to allow scientists to do various experiments via the Web. This portal has been built, evaluated, and used by several environment research groups (Arthurs et al. 2003; Cabaniss et al. 2003; Madey et al. 2003; NOM 2003).

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

2 NOM Research Portal Overview

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. 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 (Cabaniss et al. 2003). 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.

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 separated. A Web-based portal was built to enable scientists from different geographical areas and research areas to work together virtually. The NOM research portal is running on a distributed cluster. This cluster has multiple dual processor PCs running Redhat Linux 8.0 and Windows 2000 server operating system. These machines in the clus-

ter include a HTTP server, application servers, database servers, a reports server, and a data mining server. The architecture is shown as Figure 1. Scientists can access this portal from any standard Web browser.

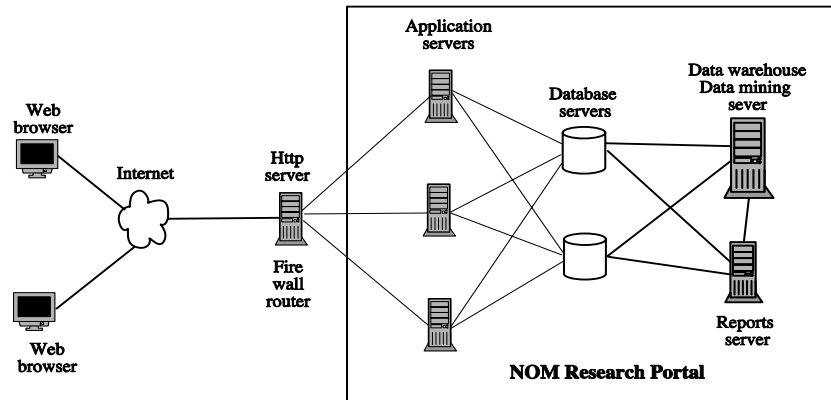


Figure 1. The architecture of the NOM research portal

2.1 Technologies and Platform

With the evolution of Internet technology, especially after Sun Microsystems introduced the J2EE technology (J2EE 2003), there is growing interest in using the Web as a new platform for scientific applications. J2EE provides developers with the development tools and runtime capabilities to build applications that meet security, scalability, and maintenance requirements. Employing the J2EE technology that includes Java Server Pages (JSPs), Servlets, JavaBeans, and Java Database Connection (JDBC) can overcome the disadvantages of traditional alternatives, such as Common Gateway Interface (CGI), for the dynamic creation of Web pages. The Oracle RDBMS was chosen for this purpose due to its popularity, reliability, and powerful functionalities. OC4J (Oracle9iAS container for J2EE) was selected for the application servers. It's a fast, lightweight, highly scalable, and easy-to-use J2EE environment. OC4J is written entirely in Java and executes on the standard Java Development Kit (JDK) virtual machine. It provides all the containers that J2EE specifies and can be downloaded from Oracle Web site (Oracle 2003).

2.2 Design Pattern

In order to easily maintain the programs, add new components, and reuse existing components, the design of the portal follows the Model-View-Controller (MVC) design pattern (Kassem 2000). All components of the NOM portal are implemented within a thin-client, three-tier architecture by following the MVC design pattern. Servlets process the incoming requests and manage the session and then call the corresponding JavaBeans components to handle the application logic. Finally, JSPs

generate the dynamic presentations to users. 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 XHTML, XML/XSL, and JSPs instead of Java applets. Although applets can make the Web page look more attractive, not all Web browsers support Java applets very well. Sometimes, users need to download the Java Runtime Environment (JRE) in order to run the Java applets correctly. Normally, the size of Web pages written in Java applets is bigger than that of Web pages written in XHTML/JSP and have longer response times, especially when the Internet connection is slow.

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.

The application tier handles the core scientific logic of the application. EJB components in an EJB container and JavaBeans are used for implementing the application tier. In the NOM system, we used JavaBeans. An Oracle database was designed as the backend to store all the data input and output.

2.3 Functionalities of the NOM Research Portal

The NOM portal provides the following functions:

- Distributed computational resource utilization: Users can configure and invoke their simulations through a Web interface, and a job manager allocates computational resources on remote sites transparently.
- 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 portal. Some particular tools can only be accessed by authorized persons. Users have access to their own simulations, and other users cannot access data that have not been authorized for public usage.
- Communication tools integration: A discussion board and a chat room are integrated into the portal in order to facilitate communications between users.

3 XML-based NOM Markup Language (NOML)

XML allows user-defined tags to be used to represent any kind of information. Users can develop their own XML documents formatting, content, and specifications by defining a set of tags and attributes to represent their own data (McLaughlin 2001; Ray 2001).

The XML-based model ensures uniformity across components and helps to abstract the component structure and implementation from the component interface. The NOM portal 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 portal. 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 access 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 access 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. The DTD is shown as below.

```
<!-- simulationsetup DTD setup.dtd -->

<!-- Root Element -->
<!ELEMENT setup (environment, moleculetypes, owner)>

<!-- Main Element -->
<!ELEMENT environment (environmentid)>
<!ELEMENT moleculetypes (moleculetype)*>
<!ELEMENT moleculetype (moleculeid, distribution)>

<!-- Basic Element -->
<!ELEMENT environmentid (#PCDATA)>
<!ELEMENT moleculeid (#PCDATA)>
<!ELEMENT distribution (#PCDATA)>
<!ELEMENT owner (#PCDATA)>
```

Users can view the DTDs and download the NOML examples from the Web. The simple tree structures of the sample documents in NOML format are shown in Figure 2.

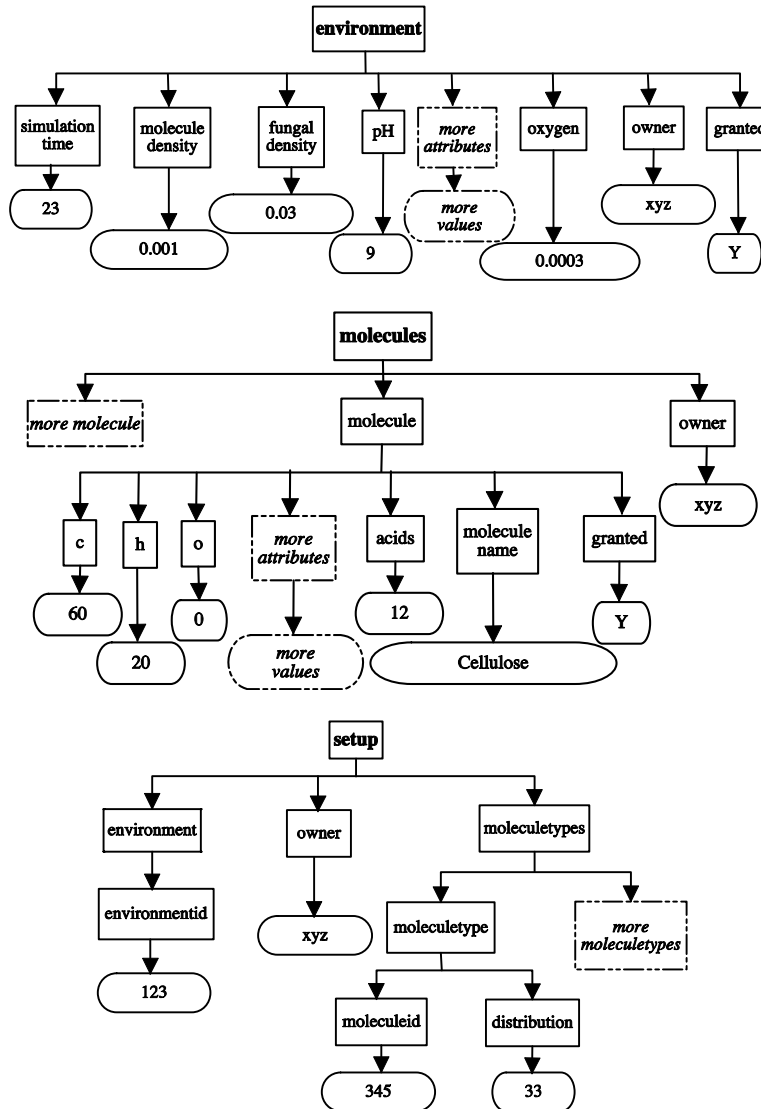


Figure 2. Tree structured view of three types of documents in NOML format. Note: not all the elements are shown in the graph, as indicated by the dash lines.

4 Portal Components

The NOM portal includes Web-based simulators, job manager, search engine, NOML uploader, molecule editor, molecule validator, chat room, discussion board, and document management tool. The details of these components are described in the following sections.

Figure 3 illustrates components in the NOM portal. Users can access these tools and services in the portal environment through multiple Web-based interfaces from a standard Web browser.

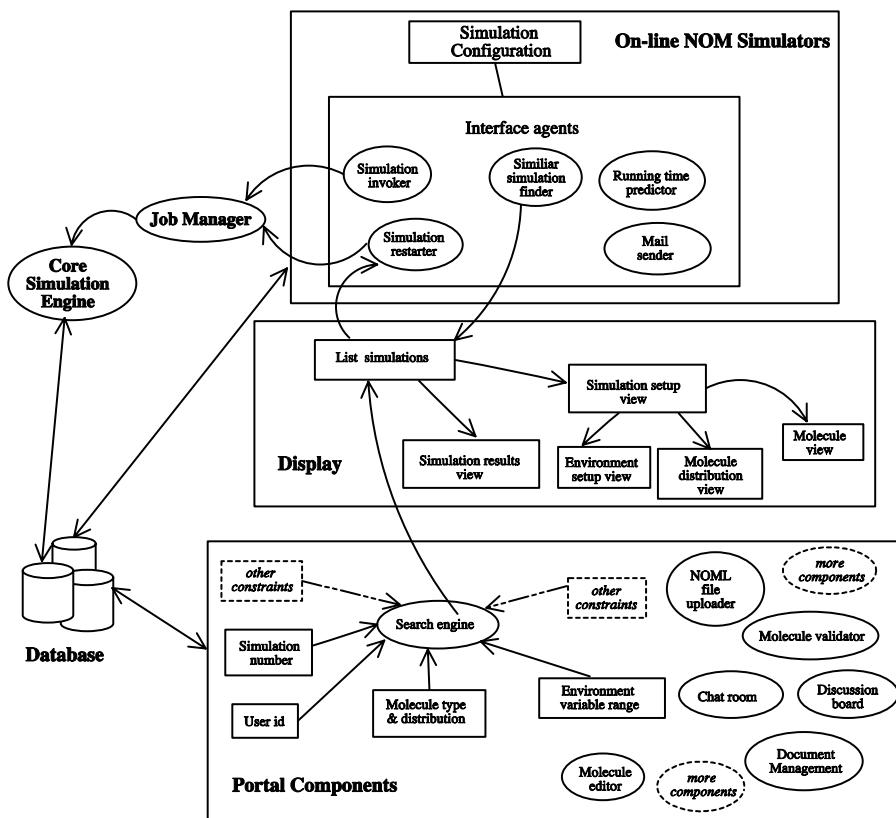


Figure 3. Components of the NOM research portal environment

4.1 Web-based NOM Simulators

NOM simulators are core parts of this portal. 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 XHTML and JSP pages and invoke their simulations on the remote computer cluster. The intelligent web interface provides a set of intel-

ligent 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.

Currently, in order to meet different requirements from different users, two different NOM simulators have been built into the portal. 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. In the stochastic model of the evolution of NOM in discrete time and space, NOM is presented as a large number of discrete molecules with varying chemical and physical properties. Individual molecules can be transported through the soil medium via water flow, adsorb on the soil particle surfaces, and react with other molecules, micro-organisms, and the environment. The chemical and physical attributes, as well as behaviors of molecules are simulated using the Java programming language and Swarm library from Santa Fe Institute (SantaFe 2003). More details on the modeling and the implementation of core simulation engines are described in (Xiang et al. 2003).

The purpose of these NOM simulators is to provide scientists a test bed for their theoretical analysis and experimental results for studying NOM. 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.

4.4.1. NOM Simulation Model Overview

An 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 inputting 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.

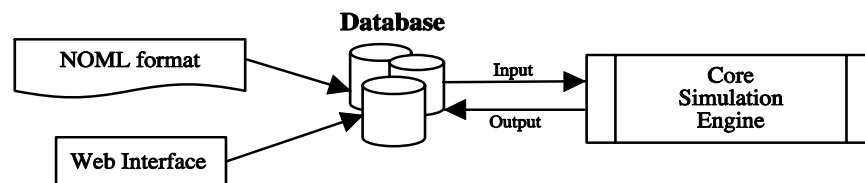


Figure 4. Simulation Input and Output

4.1.3. Intelligent Agents

Intelligent agents built 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. 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.
- 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. The agent can return an ordered list of similar simulations, with the most similar simulation listed first. Users can also view the parameter setup of similar simulations. The basic algorithm for finding the similar simulations is described as follows:
 - I. After users issue a request to search for similar simulations, the agent retrieves the data sets in the database. These data sets can be viewed as a number of points $(a_1, a_2, a_3 \dots a_i \dots a_n)$ in a high-dimensional space with m attributes $(1, 2 \dots j \dots m)$. The task is transformed to compute the distance from the current parameter setup to the other parameter setups.
 - II. Each attribute in these data sets is normalized to the $[0,1]$ interval using the equation.
$$\frac{a_{ij} - \min(a_i)}{\max(a_i) - \min(a_i)}$$
 - III. There are several efficient algorithms for finding the nearest neighbor on a high dimensional plane. For simplicity, 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 reac-

tions 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. A table is created for storing the state. A series of SQL statements is executed through JDBC code. 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 portal 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 produce massive amount of data and take days to run. 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 and Szalay show how important new data mining algorithms are in helping scientists to access their on-line data more quickly (Gray and Szalay 2002). 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 large data sets. Huang describes the detail of building the data warehousing and employing data mining technologies (Huang et al. 2003). We enabled the display of 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 examples of both input information and of sample output from a simulation executed by a scientist.

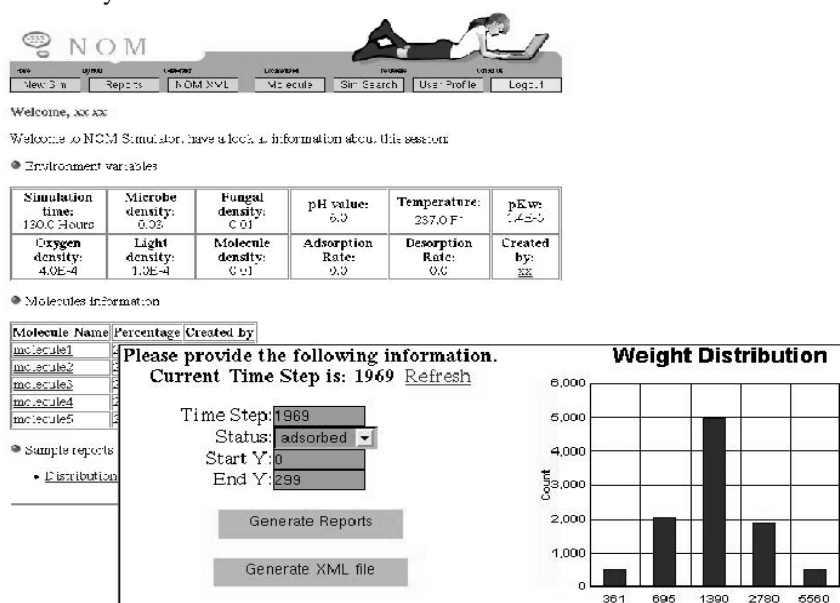


Figure 5. Two screenshots of the configuration and sample output of a simulation

4.3 Job Manager

After users submit jobs, a simple job manager in this portal is invoked and assigns tasks on several simulation servers to achieve load balancing. The goal of load balancing is to distribute simulations evenly among application servers. There are many load-balancing schemes available in the literature. A simple round robin algorithm is used for job manager implementation. New jobs are assigned to the application servers sequentially. For example, suppose we have 6 jobs to be assigned to 5 application servers, then job 1 is assigned to application server 1, job 2 is assigned to application server 2, and so on. Job 6 is assigned to application server 1 again.

4.4 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 portal 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 users' requests.

Users can view data information corresponding to each simulation in more depth by clicking on links. Users can also view the analysis data of 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.

4.5 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, a Servlet was implemented to parse the request. Instead of saving files in the disk on the server, the 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 the information to users for further reference. The design model is shown in Figure 6.

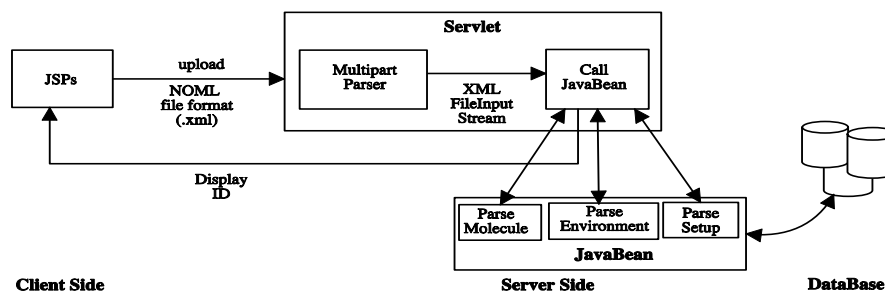


Figure 6. The design model for NOML file uploader

Users can access the NOML file uploader by providing a file name. The server side JavaBeans does the data validation. No information is stored in the database if the data is invalid and corresponding error messages are returned to users. Users can create their NOML files using a text editor. Allowed formats are published on the Web. Users can download the example files and change the values. An example NOML file is shown as follows.

```

<?xml version='1.0'?>
<!DOCTYPE setup SYSTEM "http://tenor.cse.nd.edu/setup.dtd">

<setup>
  <environment>
    <environmentid>102 </environmentid>
  </environment>
  <moleculetypes>
    <moleculetype>
      <moleculeid>1 </moleculeid>
      <distribution>33 </distribution>
    </moleculetype>
    <moleculetype>
      <moleculeid>2 </moleculeid>
      <distribution>67 </distribution>
    </moleculetype>
  </moleculetypes>
  <owner>xyz </owner>
</setup>
  
```

4.6 Molecule Editor

The “molecule editor” tool provides a flexible way for users to define new molecules for their simulations. Several simple rules are encoded to automatically validate the newly defined molecule in order to prevent the specification of chemically impossible molecule structures. For example, a simple rule is defined as follows.

If the “benzene” ring structure, is part of the molecule structure, then the total number of carbons (atom C) in this molecule structure definition must be equal to or larger than 6 times the number of “benzene” rings.

Besides providing a Web interface for the molecule definition, users can define new molecule types by uploading a NOML format file to the server. Users can also

search particular molecules according to the molecule name and owner of the molecules.

4.7 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 expert 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 by clicking a button. A set of molecules that is indicated for public usage by their owner is displayed. Users who participate in the portal can share only validated molecules. 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.

4.8 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 portal. 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.

4.9 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 Evaluation

We conducted a preliminary evaluation and testing of the NOM portal using several approaches: 1) a value analysis to the research scientist users, and 2) formal software testing. Each approach is briefly discussed in turn below.

The potential value of the NOM portal to environmental research scientists can be broken into two categories: 1) support to the actual scientific research and 2) communication support. The NOM portal permits scientists to compare their field and laboratory data with predictions made by the NOM simulators within the simulator. Several simulations, each developed for a different scientific project, are implementations of models and theories developed by the scientists to explain their

experimental data. For each simulation, the proposed explanatory model or theory could not be easily reduced to a system of equations and thus, a simulation was the practical way to compare the predictions of the model or theory against observed data. In all cases, the initial simulation did not fit the data and the underlying model was revised, resulting in a better model or theory of the NOM phenomenon being studied by the environmental scientists. Once the scientists were convinced that their revised explanatory model or theory satisfactorily fit their experimental data, the simulation was made available to other environmental scientists who either compared the predictions of the simulations with their data, or “played around” with the simulation to gain an understanding of the proposed model of the original environmental scientist, then using the communication tools of the portal to exchange ideas and comments between the scientists.

A formal test plan was developed, implemented and used to assess the alignment between the design goals and the implementation of the NOM portal. Standard software engineering methods were employed (Ieeepress 1984; Schach 1993) resulting in minor revisions to the NOM portal.

6 Conclusions and Future Work

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

In the future, some powerful third-party tools (such as video/audio conferencing) that rely on a fast and efficient high bandwidth network technology and advanced hardware (e.g., Internet2) may be integrated. Oracle Collaboration Suite (Ocs 2003) offers integrated email, voice mail, phone, fax, scheduling, calendaring, meeting management, and file management. It could also be considered for integration into the NOM portal.

An XML-based NOM Markup Language, NOML, which provides a standard definition for the molecule and simulation configuration is described in the paper. The NOML not only supports data communication between users but also give us flexibility in extending the on-line NOM simulators. It also supports building new XML-based Web services in the near future. Due to the limitations of DTD for describe the NOML, XML Schema (XMLS) will be used to describe the NOML. A WYSIWYG NOML editor will be created and integrated into the portal.

By requests and suggestions from our users, a set of new simulation models for the NOM study will be built and integrated into the portal. The communication between these models can be achieved by the extension of the NOML. The NOM portal 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 University of Notre Dame for their respective discussions on the Web interface, input, and testing of the NOM portal.

References

- Arthurs, L., Maurice, P., et al. (2003). "Agent-based simulation of biocomplexity: Effects of adsorption on natural organic mobility through soils", *226th American Chemical Society National Meeting*. New York City, NY, USA.
<http://www.chemistry.org/portal/a/c/s/1/acsdisplay.html?DOC=meetings\past.html>.
- Cabaniss, S., Madey, G., et al. (2003). "Stochastic synthesis model for the evolution of Natural Organic Matter", *225th American Chemical Society National Meeting*. New Orleans, LA, USA. <http://www.chemistry.org/portal/a/c/s/1/acsdisplay.html?DOC=meetings\past.htm>.
- Ecoportal (2003). *Eco-Portal, an information gate way empowering the movement for environmental sustainability*. <http://www.eco-portal.com>.
- Eps (2003). *Earth science portal*. <http://esdcd.gsfc.nasa.gov/ESD/portal>.
- ESeFDee (2003). *A marine science portal*. <http://www.dvz.be/Portal>.
- Gray, J. and Szalay, A. (2002). "The world-wide telescope", *Communication of the ACM*, 45(11): 51-55.
- Huang, Y., Madey, G., Xiang, X. and Chanowich, E. (2003). "Web based molecular modeling using Java/Swarm, J2EE, and RDBMS Technologies", *Seventh Annual Swarm Researchers Meeting (Swarm 2003)*. Notre Dame, IN, USA.
<http://www.nd.edu/~swarm03/Program/Abstracts/HuangSwarm2003.pdf>.
- Ieeepress (1984). *Software Engineering Standards*: Institute of Electrical and Electronics Engineers.
- j2ee (2003). *Java 2 Enterprise Edition*. <http://java.sun.com/j2ee>.
- Kassem, N. (2000). *Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition*: Addison-Wesley.
- Koslow, S.H. and Huerta, M.F. (2000). *Electronic collaboration in science*: Lawrence Erlbaum Associates.
- Madey, G., Huang, Y., et al. (2003). "Simulation of biocomplexity: Interactions of NOM, mineral surfaces, and microorganism in soils", *ASLO 2003 Aquatic Science Meeting*. Salt Lake City, Utah, USA.
<http://www.sgmeet.com/aslo/slc2003/viewabstract2.asp?AbstractID=771&SessionID=SS3.08>.
- McLaughlin, B. (2001). *Java & XML, 2nd Edition*: O'Reilly & Associates.
- Microbes (2003). *Microbiology information portal*. <http://www.microbes.info>.
- NOM (2003). *Stochastic Synthesis: Simulating the Environmental Transformations of Natural Organic Matter*. <http://www.nd.edu/~nom/>.
- Ocs (2003). *Oracle Collaboration Suite*. <http://otn.oracle.com/products/cs/content.html>.
- Oracle (2003). *Oracle technology*. <http://www.oracle.com>.
- Ray, E.T. (2001). *Learning XML*: O'Reilly & Associates.
- SantaFe (2003). *Santa Fe Institute research*. <http://www.santafe.edu/sfi/indexResearch.html>.
- Schach, R.S. (1993). *Software Engineering*: Asken Associates.
- Warr, W.A. (2001). *Science portals on the Internet*. <http://www.warr.com/wzarc01.html>.

Xiang, X., Madey, G., Huang, Y. and Cabaniss, S. (2003). "A stochastic simulation of natural organic matter and microbes in the environment", *2003 World Conference on Natural Resource Modeling*. Beaufort, North Carolina, USA.
<http://resourcemodeling.org/conferences/2003carolina.htm>.