A Science Collaboration Environment for the Network for Earthquake Engineering Simulation

Choonhan Youn, Chaitan Baru, Ahmed Elgamal, and Dogan Seber
San Diego Supercomputer Center, University of California at San Diego
9500 Gilman Drive La Jolla, CA 92093-0505, U.S.A
{cyoun, baru, seber}@sdsc.edu, elgamal@ucsd.edu

ABSTRACT
The vision of cyberinfrastructure is to provide "the comprehensive infrastructure needed to capitalize on dramatic advances in information technology," in support of science and engineering applications. The development of collaboration environments based on "science portals" plays an important role in achieving this cyberinfrastructure vision. While online, discipline-specific problem solving environments have been in use for many years, the attempt to create a common cyberinfrastructure for this purpose is a more recent development. In this paper, we address the current effort for building such a science collaboration portal as a joint effort between the GEON (GEOsciences Network) and NEES (Network for Earthquake Engineering Simulation) projects. In particular, we present recent work in developing portlets for providing access to computational simulation tools.

Keywords
Grid portal; service-oriented models; Web services; cyberinfrastructure; collaboratory.

1. INTRODUCTION
The concept of *cyberinfrastructure* encompasses advanced scientific computing as well as a more comprehensive infrastructure for research and education based upon distributed, federated networks of computers, information resources, on-line instruments, and human interfaces [1]. Science communities are increasingly becoming dependent upon such cyberinfrastrcuture for their research and education. To achieve the goal of increasing research productivity and effectiveness of education, the cyberinfrastructure must provide effective tools to end users for online collaborations, access to computing resources and ability to launch computational tasks, and sharing of data and other resources with others in a given community. The ultimate power of cyberinfrastructure arises from the ability to leverage capabilities developed for one community for use by another community resulting in rapid deployment of key functionality. Collaborations among projects—such as the one described here between GEON and NEES—serve this purpose, and are facilitated at a technical level by using the principles of a service-oriented architecture and a common set of core Grid/Web services [2, 3].

The GEOsciences Network (GEON, www.geongrid.org) is a large-scale collaborative cyberinfrastructure project involving information technology and geoscience researchers from multiple institutions [4]. The focus is on building data-sharing frameworks, developing tools and services, and identifying best practices with the objective of dramatically advancing geoscience research and education. GEON has adopted a service-oriented approach and a portlet-based approach—both of which are applicable to other science grid projects as well—that has led to the development of a number of reusable portal services [5]. The GEON portal framework is being leveraged and reused by a number of GEON collaborators including, the CUAHSI (Consortium of Universities for the Advancement of Hydrologic Science) [6] Hydrologic Information System (HIS), for hydrologic science; the Chesapeake Bay Environmental Observatory (CBEO) [7], for environmental science and engineering; the NEON (National Ecological Observatory Network) Single-string Testbed [8], and the TEAM (Tropical Ecology Assessment and Monitoring Network) [9], for ecological science; the NEES (Network for Earthquake Engineering Simulation) [10], for earthquake engineering; and, TDAR (The Digital Archaeological Record) project [11], for archaeology.

In this paper, we present our initial efforts in developing and deploying services and tools for the NEES community, based on the GEON software infrastructure. The NEESit component of NEES is a service-focused organization that plays a leadership role in providing information technology infrastructure for the NEES project. This includes providing the capability to perform large-scale computational and physical hybrid simulations, organize and share data, and collaborate with colleagues in worldwide knowledge extraction efforts [12]. The NEES Portal is the access point for all online resources, services and tools provided by the NEES community.

2. PORTAL-BASED SERVICES
Similar to some other science projects, GEON is based on a service-oriented architecture [13], where the user interfaces to services are provided via *portlets* implemented within a portal framework, and using a standard portlet API (JSR 168). These pluggable portal components process user requests and generate dynamic content, and can be deployed on remote sites. Using a standard software stack and API, partner sites are able to develop and deploy software that is compatible with the rest of the system and can be reused by others in the network in a seamless fashion [5]. The NEES Portal has also been implemented using this approach.

Similar to GEON, the NEES community also consists of a wide variety of stakeholders, including 15 large-scale NEES equipment sites, approximately 40 NSF-funded research grants (referred to as "NEESR projects"), and the broad earthquake engineering researcher and practitioner community. The role of NEESit is to provide access to experiment and other databases, visualization tools and environments, large-scale grid-based supercomputing, and telepresence capability for geographically distributed, synchronized, large-scale experiments to the research and education communities [12]. Given the similarity in the community structure and the implementation approach, we were able to leverage the already existing GEON software infrastructure to support the NEES environment. The service model naturally supports reusability, modularity, and sharing as well as the need to support a multi-disciplinary community interested in building complex applications. For example, a developer can build a simulation application by utilizing a library of preexisting services, and substitute different components based on the algorithms or approaches required for the particular domain. Services can also be composed in a hierarchical (nested) fashion to deal with complex problem solving environments.

The NEES Portal, based on the GEON implementation, provides web-based user interfaces for accessing a variety of distributed services that act as middleware wrappers to allow a heterogeneous collection of resources to be accessed in a relatively uniform fashion, as illustrated in Figure 1. The core services are as following:

- **Registration:** the registration service enables registration of metadata for data, tools, applications, and ontologies according to a predefined metadata schema. In the case of flat files, the physical data file may be stored in a Storage Resource Broker (SRB) data repository [14] for later use and/or download. GEON supports registration of a variety of standard data formats (e.g. relational databases, Excel spreadsheets), including some that are specific to geospatial data (e.g. shapefiles, geoTIFFs. The registration system has been developed for easy extensibility for new data types. However, the validation scheme for each new data type must be provided by the developer. A few data types were added specifically for NEES. Registering a resource allows users to subsequently use and share that resource with other users.

- **Search:** the portal provides a Basic and an Advanced Search capability. Basic Search is a keyword-based search across data, tools, services, and ontologies. The search is applied across certain metadata fields (Description, Title, and Keywords). The Advanced Search allows users to search for resources based on various metadata fields including spatial coverage (using a Google Map API), and temporal coverage, as well as using ontologies, or any combination of these criteria.

- **MyWorkspace:** this service provides access to a secure workplace where data, input files and computational resources may be stored, e.g. for a research project. It provides a private space where users may store results from search operations, output from mapping programs, and results from computations. Data in the workspace can be integrated. For example, multiple shapefiles obtained using the Search service can be integrated to create a new map; multiple databases and shapefiles can be integrated into a *virtual relational resource*. The workspace also provides tools to manage submitted resources.

- **Account Management:** NEES employs the Grid Security Infrastructure (GSI) [15]. Account management services are provided by the Grid Account Management Architecture or GAMA [16], which is used by GEON and a number of other cyber communities. GAMA provides a central GSI certificate management system where “grid” users are approved and credentials are created for them based on global policies. Grid users have a single login credential and password that can be used at any portal site that has that account enabled. In addition to grid accounts, in a federation of portals, each local portal can create purely local-user accounts with no GSI credentials. These accounts may be used by developers to build the local services, or by users who are not members of NEES community that wish to access the local resources.

  GAMA provides a central GSI certificate management system. The GSI for the mutual authentication across Grid services is based on the public key infrastructure which is used for X.509 certificates, and SSL communication protocol. The user portal authentication is performed by the login portlet which accesses the GAMA server. An identity-based authorization system is used for accessing Web services. This simple authorization mechanism is performed through the gridmap file which contains the user distinguished name and the local user name. Upon the authenticated call that arrives at a certain service, the identity of the caller is checked against the owner of the application. If both identities match, the authorization for the Web service access is granted.

- **Simulation:** access to the computational environment is built around interacting Grid/Web services, hiding the complexity of grid technologies. Typical services provided by this computing environment include information services describing available host computers and applications, job submission and monitoring, file transfer, and remote file access and manipulation. We have developed a simple job management service based on the NEES application code for NEES community, and have worked closely with UC Berkeley’s application development team to test and fine tune the job management steps. Users who have applied for the NEES supercomputing allocation can submit the jobs to the TeraGrid HPC resources [17] through a web browser interface. This is described in section 3 in more details. The steps include parameter specification for application inputs, job execution, job status etc. The SimPortal application [18], which is one of the simulation services, has been integrated into the NEES portal environment, based on the Open System for Earthquake Engineering Simulation (OpenSees) [19].

- **Workflow:** Performing computational tasks can involve multiple steps. Typically, input files for the scientific application must be preprocessed, often the problem must be partitioned correctly to run efficiently in parallel, the job must be submitted and run on an HPC, perhaps the code’s output should serve as input for an another code after filtering, and finally some visualization and analysis must be done. This entire process can be automated using a scientific workflow system such as Kepler [20]. The NEESit workflows provide a framework for the design, execution, and deployment of scientific workflows enabling seamless integration of and access to high performance computing resources, large scale data sets, and instruments. For example, an open source R-package (http://www.r-project.org/) for the earthquake engineering community has been integrated with the Kepler platform with the added capability to easily create new user need-based composite actors for specific applications [21]. These examples are currently being integrated into the NEES portal.

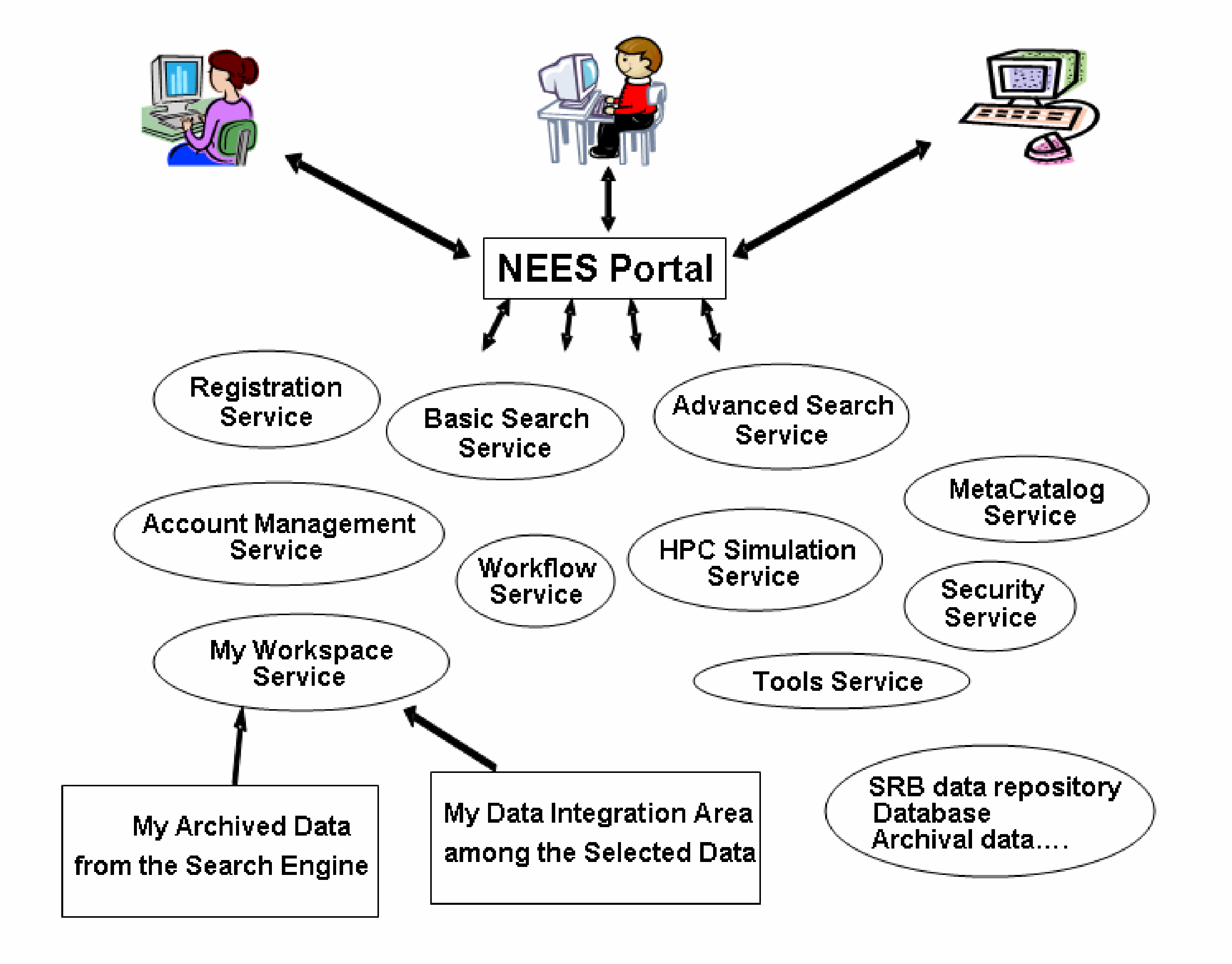

Figure 1. A Service Model for the NEES Portal Environment

Using the above set of services, the NEES portal now provides access to a variety of tools, such as computational simulation, scientific workflows, *Webshaker*, for a live shake table experiment [22], and a web-based NHCP (NEES Hybrid Simulation Communications) for the hybrid simulation [23].

By building the NEES portal upon the portal infrastructure developed by GEON, we have demonstrated that common services such as those related to search, document and tool sharing, and access to computation simulation tools can be easily leveraged across projects. This was made easier by employing a general-purpose service model with modular software components for the portal infrastructure. In conclusion, since each science domain utilizing cyberinfrastructure will require their own portal, the challenge is to find a way to develop common portal components that can be integrated into different domain portals. This has the larger benefit that the portals maintain some level of compatibility with each other and with the broader base of cyberinfrastructure.

3. COMPUTATIONAL SIMULATION

SYNSEIS – the SYNthetic SEISmogram generation tool – is a GEON domain-specific application tool that provides access to a computational environment via a portlet-based mechanism [24]. Authorized users are able to access high-end computational resources, including TeraGrid clusters. Similar to the SYNSEIS tool, we have developed a job management portlet for NEES community that provides authorized NEES users the ability to submit computational simulation jobs either to the SDSC-based TeraGrid systems or to a NEESit cluster in order to execute a parallelized version of the OpenSees code.

3.1 The Architecture of the Job Management Portlet

Figure 2 depicts a traditional three-tired distributed software architecture consisting of user interfaces in the front-end, services in the middleware, and computing resources in the backend. The user interfaces are deployed within portlets and access a variety of services implemented as part of the middle tier. These services allow a heterogeneous collection of resources to be accessed remotely in a uniform fashion.

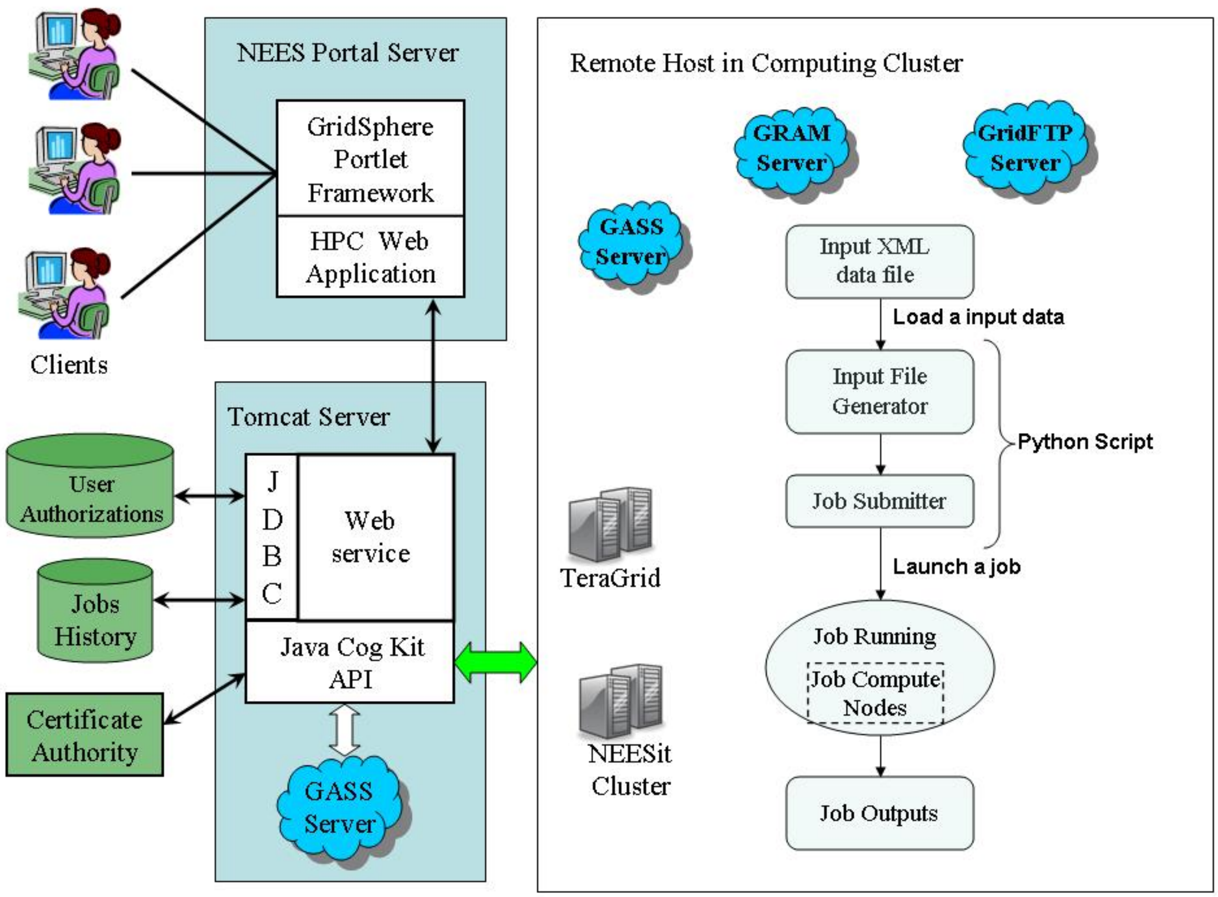

Figure 2. A conceptual architecture of the job management system.

When clients submit their simulation jobs, the job management portlet connects to the job submission Web service. This Web service (a) validates the XML input file before it is saved into the data repository for archival reference, for modification and re-use, (b) checks the user identity and authorization, and (c) creates the proxy credential for interactions with Grid services in the computational cluster. If the input has been successfully validated and the user has the authorizations, then it sends the user's XML input data to the selected computing cluster which executes the real application code, using a set of GridFTP client interfaces. To run a job remotely, the Globus (www.globus.org) Resource Allocation Manager (GRAM) gatekeeper, which is at the core of the Globus remote program execution infrastructure, is run on the remote resource. The execution begins when a GRAM client sends a job request to the remote resource. A job request is written in the Resource Specification Language (RSL). The generated job request script specifies the location of the python script, which will be executed in the remote resource environment, and the location of the GASS server, for storing the job execution messages on the remote resource. Once the job is submitted, the job request script is handled by the GRAM gatekeeper, which forks a job manager to handle the execution of the new job as well as for communications with the user.

On the backend, the python script consists of two modules: (1) An Input file generator, and (2) A Job submitter. The input file generator takes and parses user's input XML data file so that it creates several input files needed to run the code, and the queue script, in a specific directory for this experiment, which also contains the input files and batch scripts. In the second step, a Job submitter submits the batch script to the queuing system. Eventually, job outputs are generated after the application is executed on the remote resource.

After submitting a job, users are able to check the job status on the portal by accessing a job monitoring Web service. The job monitoring service is invoked using the same authorization and credential process as that for the job submission service. The job submission service returns a unique job identifier that can be used for enquiry of job status in the job monitoring service. If the job is submitted to a batch scheduler, then it is in the pending state while sitting in the queue waiting to be executed. The job may become suspended due to pre-emption mechanisms. In the case of a normal completion, the job status is “Done”, otherwise the job status is “Failed”. In our case, once the user’s job is done, job outputs are transferred into the local directory and the user’s MyWorkspace, using GridFTP clients.

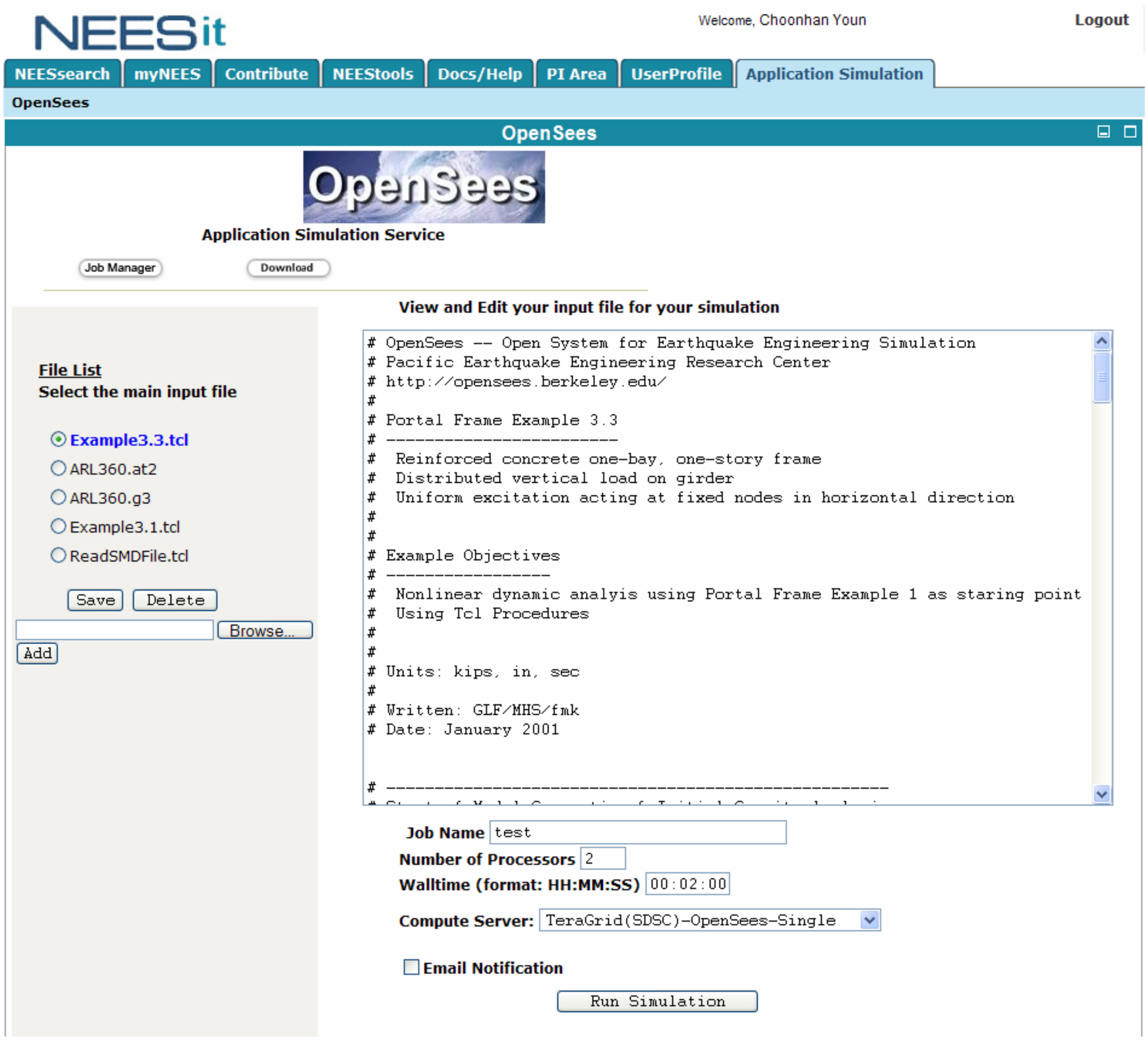

Figure 3. A User interface for managing input files before the simulation is started.

3.2 Metadata and User Interface

OpenSees (Open System for Earthquake Engineering Simulation) is the computational tool that NEES users can currently access from the Portal. OpenSees was originally sponsored by the Pacific Earthquake Engineering Research Center (PEER). It is a software framework for developing applications to simulate the performance of structural and geotechnical systems subjected to earthquakes. It is a community-based open-source software that allows earthquake engineering researchers to build upon each other’s contributions and developments. It incorporates advanced capabilities for modeling and analyzing the nonlinear response of systems using a wide range of material models, elements, and solution algorithms, and is designed for a parallel computing to allow scalable simulations on high-end computers, or parameter studies [19].

The OpenSees application reads one or more tcl scripts and data files. In case of multiple input files, the application requires a “main" input file from which it links to other input files to read the information automatically into the application. In our implementation, the XML schema used to represent the input for SYNSEIS is replaced with an implementation where the multiple input data files used by OpenSees are added into elements of the XML tag representing the program input. When users reuse this input XML file for the job submission, these tcl scripts and data files can be edited on the web browser. We structure our XML dialect definitions as being composed of the following:

- Application Information: involves the name, version, and description of the application about this input.
- Computing resource information: includes the number of processors and the wall clock time.
- Input Source information: includes various input types such as the script, local or remote path, URI, and the data file.
- Job Information: includes the job notification, and job name.

Note that we are not modifying the code to directly read the XML input data. Rather, we use an XML-based description to generate input files for the code in the proper legacy format. This XML schema simply defines the information necessary to implement the input files in a particular application.

Unlike the SYNSEIS user interface, which generates only one input data file, OpenSees may typically have more than one input file for a given run. Therefore, we had to redesign the user interface to support this capability of having multiple input files for a run. Thus, the functions provided in the OpenSees portlet include the ability to (1) allow users to upload multiple input files, (2) choose the main input file from the file list, (3) view and edit the content of each input file, and (4) add, delete, and replace an input file.

Figure 3 shows a screen snapshot of the OpenSees portlet, which illustrates how to set up the variables needed for an OpenSees simulation on the portal. Before getting to this page, users are able to specify the type of file for uploading, and whether uploading multiple files or one zipped file. The user interface, as shown in Figure 3, provides a user-friendly view of the OpenSees application, and shows the input data file in the text window, without requiring any modifications to the actual simulation code. Users are presented with a basic screen that allows multiple input data files to be entered for the simulation. After users specify parameters of the computing resources and input data files, the generated XML file is sent to the selected computational resource through the job submission Web service. The script on the remote host generates input data files in the job directory and submits the job.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have shown how we have developed services needed by the NEES community by taking advantage of the infrastructure already developed in the GEON project. We believe that such efforts leading to leveraging of common infrastructure by diverse user communities are important—they provide timely, more effective, and less expensive solutions—and create synergies among diverse communities.

Some extensions that we are considering include a job management portlet that will provide access to other simulation packages that are currently available on TeraGrid resources, such as Abaqus, ANSYS, LS-Dyna, and Matlab. Also, we are reviewing Web 2.0 technologies (namely, REST services, rich internet applications, online communities) that can be used and combined with this environment. While Web 2.0 is not a specific set of technologies, it is best characterized as a *movement* towards a network programming approach. The blurring of distinctions between web applications and web tools means even non-programmers are able to create sophisticated web applications. Combining Web 2.0 concepts with conventional cyberinfrastructure holds the promise of creating *virtual scientific communities* [25]. For instance, the GEON Resource Registration and Search portlets can be converted to Web 2.0-like services to allow scientists to develop social networks around datasets and services and lets users review and vote on submitted contents in order to capture the community's perceived value of data and services.

5. REFERENCES

[1] Atkins, D. E., Droegemeier, K. K., Feldman, S. I., Garcia-Molina, H., Klein, M. L., Messerschmitt, D. G., Messina, P., Ostriker, J. P., and Wright, M. H. 2003. Revolutionizing Science and Engineering through Cyberinfrastructure. Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure, January 2003.

[2] Berman, F., Fox, G., Hey, T. (ed.). 2003. Grid Computing: Making the Global Infrastructure a Reality. John Wiley & Sons: Chichester, England, 2003.

[3] Foster, I., Kesselman, C. 2004. The Grid 2: Blueprint for a new Computing Infrastructure. Morgan Kaufmann, 2004.

[4] Cyberinfrastructure for the Geosciences: http://www.geongrid.org.

[5] Youn, C., Baru, C., Bhatia, K., Chandra, S., Lin, K., Memon, A., Memon, G., Seber, D. 2007. GEONGrid portal: design and implementations, Concurrency and Computation: Practice and Experience, Vol. 19, Issue 12, pp 1597-1607 (2007), DOI: 10.1002/cpe.1129.

[6] Consortium of Universities for the Advancement of Hydrologic Science: http://www.cuahsi.org/.

[7] Chesapeake Bay Environmental Observatory: http://ccmp.chesapeake.org/CBEO/.

[8] National Ecological Observatory Network: http://www.neoninc.org/.

[9] Tropical Ecology Assessment and Monitoring Network: http://www.teaminitiative.org/.

[10] Network for Earthquake Engineering Simulation: http://www.nees.org/.

[11] The Digital Archaeological Record: http://cadi.asu.edu/.

[12] Network for Earthquake Engineering Simulation IT: http://it.nees.org/.

[13] Foster, I. 2005. Service-Oriented Science. *Science*, Vol 308, 6 May 2005.

[14] SRB (Storage Resource Broker): http://www.sdsc.edu/srb/.

[15] Foster, I., Kesselman, C., Tsudik, G., Tuecke, S. 1998. A Security Architecture for Computational Grids. Proc. 5th ACM Conference on Computer and Communications Security Conference, pp. 83-92, 1998.

[16] Mueller, K., Bhatia, K., Chandra, S. 2005. GAMA: Grid Account Management Architecture. IEEE e-Science 2005, Melbourne, Australia, Dec 2005.

[17] TeraGrid Project: http://www.teragrid.org/.

[18] Haupt, T., Kalyanasundaram, A., Ammari, N., Chandra, K., Das, K., & Durvasula, S. 2005. SPURport: Grid Portal for

Earthquake Engineering Simulations. International Conference on Computational Science, 2005, Atlanta, USA: Spriger Verlag, Lecture Notes in Computer Science, *3514*, 493-500, 2005.

[19] Open System for Earthquake Engineering Simulation: http://opensees.berkeley.edu/.

[20] Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao, J., Zhao, Y. 2006. Scientific workflow management and the Kepler system. Concurrency and Computation: Practice and Experience 18(10): 1039-1065 (2006).

[21] NEESit workflow documentations: http://it.nees.org/software/kepler/index.php.

[22] Webshaker experiment: http://webshaker.ucsd.edu/.

[23] NHCP (NEES Hybrid Simulation Communications): http://neestpm.sdsc.edu:8888/mm1demo/.

[24] Youn, C., Kaiser, T., Santini, C. and Seber, D. 2005. Design and Implementation of Services for a Synthetic Seismogram Calculation Tool on the Grid. ICCS 2005: 5th International Conference, Atlanta, GA, USA, May 22-25, 2005, Proceedings, Part 1, LNCS 3514, pp. 469-476, 2005.

[25] Pierce, M. E., Fox, G., Yuan, H. and Deng, Y. 2006. Cyberinfrastructure and Web 2.0. Proceedings of HPC2006, July 4 2006, Cetraro Italy.