Towards a Collaborative Portal Environment for Earthquake Engineering

S. Krishnan, K. Fowler, K. Bhatia, C. Youn, L. Van Den Einde, C. Baru

NEES Cyberinfrastructure Center, 9500 Gilman Drive, La Jolla, CA 92093-0444, USA
Email: {sriram, kfowler, karan, cyoun, lellivde, baru}@sdsc.edu

ABSTRACT:
A "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. The NEES Cyberinfrastructure Center (NEESit) has been created to deliver tools and infrastructure to enable earthquake engineers to remotely participate in experiments, perform hybrid simulations, organize and share data, and collaborate with colleagues. One of the key challenges in building such cyberinfrastructures is how to expose the back-end functionality to the end-users in an easy to use manner, in such a way that it is easy to implement, maintain, and manage. With the advent of Web 2.0 technologies, the trend has been for "Web portals" to aggregate information content from diverse sources that are implemented as a Services Oriented Architecture (SOA), and present them in a unified way. We are proposing an SOA for NEESit, along with a Web portal environment for accessing back-end computational and data resources. In this paper, we describe the current and proposed NEESit architectures, and present our initial prototype called "NEESsphere". NEESsphere features include the ability to contribute and share files and databases, access to back-end computational resources for running simulations, earthquake engineering examples and tutorials utilizing scientific workflow tools, and an online educational hybrid simulation test environment. We strongly believe that portals and SOAs are the future of all cyberinfrastructures, due to flexibility, easy of implementation and use, and the ability to leverage a diverse set of information sources at the back-end.

KEYWORDS: Portals, Cyberinfrastructures, Structural Engineering, SOA, and Collaboration

1. INTRODUCTION

Grid computing is defined as the ability, using a set of open standards and protocols, to gain access to applications and data, processing power, storage capacity and a vast array of other computing resources over the Internet (Foster, 1999). Under the umbrella of Grid computing, mechanisms are provided for single sign-on, job submission and data transfer, in order to allow the coupling of distributed resources in a seamless manner. The concept of a "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 (Atkins, 2003). Cyberinfrastructures represent the evolution of high performance and Grid computing, making these technologies truly usable by all the nation's scientists, engineers, scholars, and citizens. It is also often referred to as "e-Science", especially in the UK, which has a major e-Science initiative. Science communities are increasingly becoming dependent upon such cyberinfrastructures for their research and education. To achieve the goal of increasing research productivity and effectiveness of education, the cyberinfrastructures 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. Over the past decade, an extensive amount of time and effort has been invested in building such cyberinfrastructures for various scientific and engineering communities, such as:

- Life sciences: the National Biomedical Computation Resource (NBCR), the Biomedical Informatics Research Network (BIRN), the cancer Biomedical Informatics Grid (caBIG);
- Geo-sciences: the Geosciences Network (GEON));

- Metagenomics: the Community Cyberinfrastructure for Advanced Marine Microbial Ecology Research and Analysis (CAMERA));
- Earthquake engineering: the Network for Earthquake Engineering Simulation (NEES);
- Meteorological research: (the Linked Environments for Atmospheric Discovery (LEAD) project.

One of the key challenges of building cyberinfrastructures is how to expose all the functionality supported at the back-end to the scientific end-users in an easy to use manner, in such a way that it is easy to implement, maintain, and manage given the dynamic nature of scientific research. This can be broken up into two distinct, yet important, issues. Firstly, from the perspective of an end-user, the cyberinfrastructure should provide an intuitive user interface for leveraging all the applications of interest, preferably from a single point of entry, and without the need to install complex client tools. Secondly, from the perspective of a software architect or a developer, the cyberinfrastructure should allow the addition of new applications, data sources, and collaboration tools, without having to make large-scale changes to the software stack.

Historically, most of the information on the Internet has been disseminated through traditional Web sites. These have tended to be fairly monolithic because all the information sources and applications on most Web sites used to be co-located. The disadvantage of such monolithic Web sites is that it is very hard to retrofit newer applications into the framework, and also to share common features that may be useful across multiple projects. Recently, with the advent of Web 2.0 technologies (O'Reilly, 2005), the trend has been for Web sites to aggregate information content from diverse sources, and present them in a unified way. These have come to be referred to as "Web portals", since they provide a single point of entry or access to a set of information sources at the back-end.

Typically, the information sources at the back-end are implemented within a Services Oriented Architecture (SOA), with the help of Web service technologies such as SOAP (http://www.w3.org/TR/soap/) and REST (Fielding, 2000). The computer science benefits of service orientation are well known. Services reduce complexity by encapsulating the service implementation, and providing simple well-defined interfaces. With the help of open standards, they ensure interoperability across systems and architectures – in other words, a client from one system can easily access a service implemented within another system, as long as it understands the protocols and the interfaces used by the service implementation. Web portals can be built as thin "shells" that can access services implemented elsewhere, and provide a unified interface to the scientific end-user.

The NEES Cyberinfrastructure Center (NEESit) at the San Diego Supercomputer Center (SDSC) is a service-focused organization created to deliver information technology tools and infrastructure to enable earthquake engineers to remotely participate in experiments, perform hybrid simulations, organize and share data, and collaborate with colleagues. The requirements of our community and the dynamic nature of the collaborations lend themselves very well to be implemented as an SOA, with access to the information technology tools to be provided via thin portal interfaces. New applications can be developed independently by the community and exposed as Web services – and aggregated easily, in theory, into the portal environment. In this paper, we describe the state-of-the-art NEESit architecture, and discuss how we are moving towards a SOA/Portal framework.

The paper is organized as follows. In Section 2, we describe the current NEESit architecture, and present its various features and functionalities. In Section 3, we present the proposed architecture for the NEES Portal, and discuss our future plans for NEESit looking. In Section 4, we discuss our initial prototypical implementation of a portal architecture for NEESit called "NEESsphere", and describe how we have re-factored the various pieces to fit within this architecture. Finally, we present our conclusions in Section 5.

2. NEES IT ARCHITECTURE: STATE OF THE ART

The current NEESit end-to-end architecture is shown in Figure 1. In particular, it can be divided into the following tiers:

Infrastructure: This tier consists of the raw hardware resources that support the NEESit system. In particular, it consists of the Web servers hosting the Web sites, the application servers that host the services, the database servers hosting NEES data, a set of development clusters, and distributed Grid resources like the TeraGrid (http://www.teragrid.org/) for running computationally intensive simulations. In the past, we used a MySQL relational database, but this has been replaced by a production quality Oracle system since mid 2008.

Services: This tier consists of a set of services that provide easier access to the infrastructure layer. GridAuth (Warnock, 2005) provides the authentication services, and other services are provided for authorization and job management. Access to the back-end databases is provided by a set of data-oriented services, e.g. a set of Web services for uploading and downloading data from the database.

NEES Core Service APIs: The NEES core service APIs provide a layer of abstraction over the more generic services layer, and expose NEES-specific functionality that is visible to end-user clients. In particular, it provides a NEES authentication API that leverages GridAuth, and a REST-based data access API that leverages the underlying Web services to query metadata about projects and experiments, and upload and download data into the back-end database.

Applications: The applications tier leverages the NEES core service APIs to interact with the back-end data and computational layer, and enables a scientific end-user to access and visualize data, and run experiments. The two key components in the applications tier are NEEScentral, which is a PHP-based Web site that provides features such as user registration, and secure access to the projects and experiments inside the back-end database, and a set of NEES applications distributed separately. Some of the key applications are Data Turbine, which works behind the scenes to store streaming data for later playback, RDV (the Real-time Data Viewer), which provides an interface for viewing real-time, synchronized, streaming data from an equipment site, flexTPS (the flexible TelePresence System), which enables remote viewing and robotic control of telepresence cameras via a web browser, iSeismograph, which enables recording, viewing and saving real-time streaming of acceleration data synchronized with video with an Apple Macintosh laptop computer, NTCP for Matlab, which provides the necessary programs to run a three-site hybrid simulation, Kepler, which provides a platform for scientific workflows, and WebEx, which provides online meeting management tools capable of presentations, application sharing, and shared document editing.

The current NEEScentral data repository does not qualify as a "portal" as per our definition in Section 1. Even though it does leverage some services from the bottom tiers, it doesn't aggregate content from diverse sources – it simply provides access to the Oracle database back-end in a secure fashion. It is not capable of providing access to data from other remote locations, and requires all data to be uploaded and downloaded into the central NEES database. Furthermore, it does not provide a single point of entry to the various applications described above. If a user has to use any of the said applications, they have to download and access them individually at this point. This makes it difficult for end-users to keep track of the various software tools available to them, and for the developers to manage and maintain these individual components and make sure that they work harmoniously with each other. This is the reason why we are moving towards a true portal architecture for NEESit, which is capable of providing access to data hosted at different locations, and applications being developed independently by and for the NEES community.

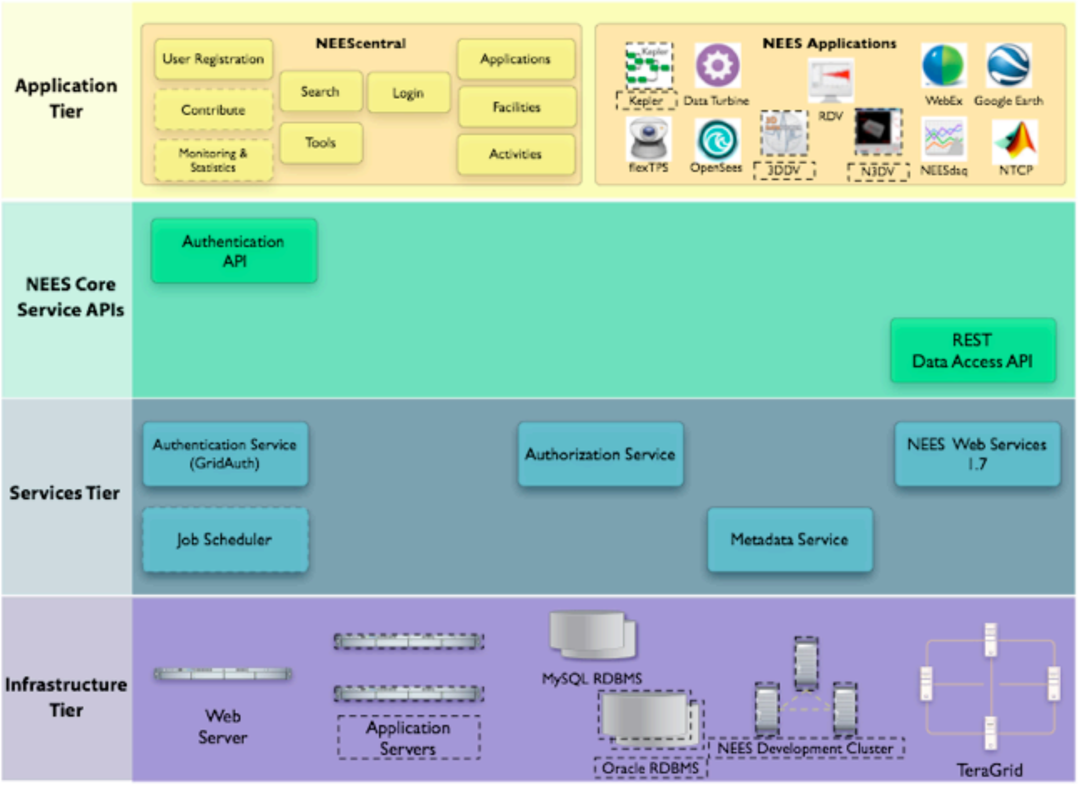

Figure 1: Current NEESit Architecture

3. THE NEES PORTAL ARCHITECTURE

The proposed NEESit portal architecture is show in Figure 2. The different tiers of the architecture stay the same; however, many new pieces are introduced at the various tiers.

The infrastructure tier is almost the same as before, except that the MySQL database back-end is formally deprecated, and replaced by the Oracle RDBMS system. The services tier has several new members – the GridAuth authentication services are replaced by the Grid Account Management Architecture (GAMA) 2.0 services, which provide an easy-to-use abstraction around standard Grid security mechanisms. Prior versions of GAMA (Mueller, 2005) already allowed the creation of Grid security credentials and Web service APIs for the retrieval of limited lifetime X.509 certificates (as is standard in Grid computing). GAMA 2.0 improves on this by providing alternate mechanisms for authentication, e.g. via the LDAP protocol. LDAP is used as a standard mechanism for authenticating users on clusters, Content Management Systems (CMS) such as Drupal (http://drupal.org/), Wikis, etc., and hence can be used by a wider variety of applications. GAMA 2.0 also provides mechanisms for Single Sign On (SSO) across multiple portals, and role-based authorization. Other new services introduced are for resource brokering in order to leverage a larger set of back-end computational resources for simulation purposes, and services for cataloging and management of metadata. Data mediation services are also planned, and this will be discussed in Section 3.1.

The NEES Core Service APIs tier is also augmented to include several new interfaces. These APIs mirror the more generic lower level services, which may actually be implemented and hosted elsewhere. The service APIs can be thought of as thin local abstractions for remote or local service implementations. The applications tier provides practically the same functionality as before. However, rather than implementing everything inside a Web interface in a monolithic fashion, it simply integrates the various features provided by the underlying APIs.

In this way, a user is still able to access a set of information tools, irrespective of whether it is implemented locally or remotely.

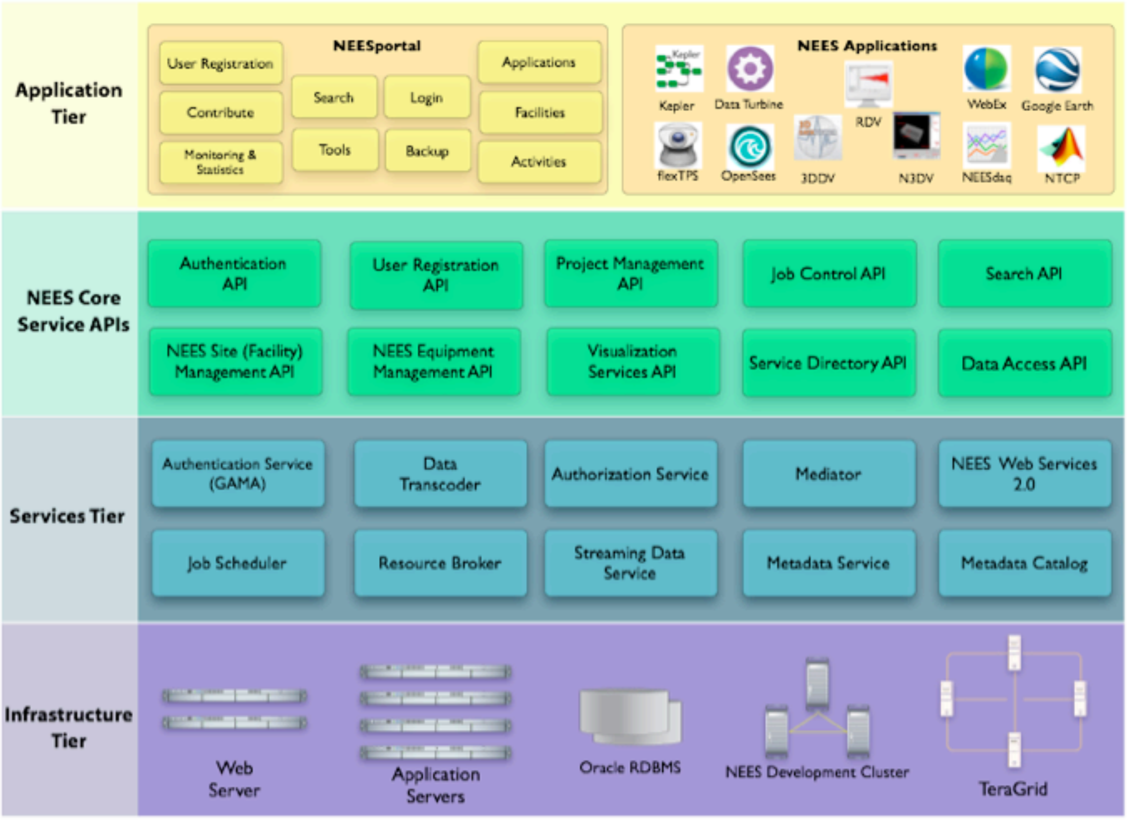

Figure 2: Proposed NEES Portal Architecture

The NEES Portal provides a single point of entry to all the other NEES applications described in Section 2, using Web 2.0 technologies. Tools such as Ajax and Adobe Flex can be used to build what are called “rich internet applications”. These applications can bring the interactivity and visualization capabilities via expressive Web based applications, which were hitherto possible only by building “thick desktop clients” that had to be downloaded by the end-users. In theory, several of the applications can be rewritten using the above tools, and exposed via the NEES Portal. For applications that can not be built this way, the NEES Portal can provide alternate mechanisms for access such as Java Web Start, which enables standalone Java software applications to be deployed with a single click over the network.

Another concept that has been popularized by the Web 2.0 is that of “social network services”, which focus on building online communities of people who share interests and activities, or who are interested in exploring the interests and activities of others. Social networks, such as the ones provided by Facebook (http://www.facebook.com), can be conceptually extended to form collaborations and communities of scientific end-users. Scientific end-users can leverage this to share data, experiment set-ups, publications, etc. and manage access to a set of users that are part of their social network. The myExperiment project (http://www.myexperiment.org/) provides a portal to find, use and share scientific workflows and other files, and to build communities, and has been one of the pioneering social networking sites for scientific users. The NEES Portal will also provide a mechanism to interact with their colleagues using such techniques.

3.1. Federated Data Access

The current data model in NEES enables only the use of a central database repository. In the future, we plan to allow access to a federated set of databases via the NEES Portal (see Figure 3). This is beneficial because it is not always practical to store all data at a central location – in some cases, contributors would rather store their data themselves, and just enable remote access to it. However, providing access to data from various sources has its set of issues. First, metadata about the various data sources still has to be stored at some location so that the various data sources can be discovered – a metadata catalog service can be used for this purpose. Secondly, remote data has to be accessible programmatically in a secure and high-speed manner – Web services and high-speed networks and protocols alleviate this issue. Finally, it may not be possible to enforce a common database schema across all the data sources. In this case, there is a need to build data "mediation" middleware and earthquake engineering ontologies. We plan to address all these requirements via the NEES Portal. We are currently working with researchers at UC Santa Barbara and IRIS (http://www.iris.edu/) to investigate storing strong motion at IRIS and providing access to it through the NEES Portal, as a pilot example.

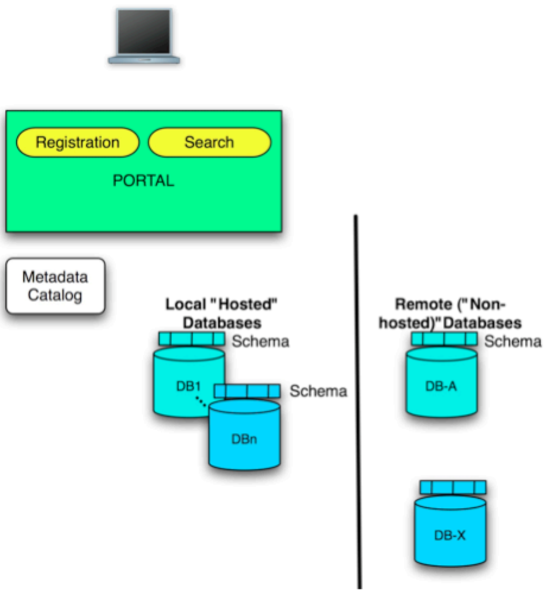

Figure 3: Federated Data Architecture

4. PROTOTYPICAL IMPLEMENTATION

As our first attempt to build a portal for the NEES community, we have a built a prototype of the NEES Portal called "NEESsphere". NEESsphere is built on top of an SOA, and generally follows the architecture described in Section 3. The user interfaces for the services are provided via "portlets" implemented within the GridSphere portal environment (Novotny, 2004), which is a Java Portlet Specification (JSR168) compliant open source portal framework. Portlets are pluggable user interface components that are managed and displayed in a web portal. Portlets produce fragments of markup code that are aggregated into a portal page by the portal framework. These pluggable portal components process user requests and generate dynamic content, and can be also be packaged deployed on remote sites, thus promoting code reuse. Indeed, several components of NEESsphere have been reused from the portal built for the GEON project (http://www.geongrid.org/).

Figure 4 provides an overview of the features provided by NEESsphere. It provides a one-stop shop for the various individual information tools supported by NEESit. This can be categorized into four general areas:

Telepresence: Telepresence refers to the set of IT-based technologies that allow a member of the earthquake engineering community (that includes researchers, practitioners, students, social scientists and policy makers) to remotely engage in an experiment at a NEES equipment site and contribute to all aspects of the experiment within his/her abilities in real-time as if the participant were present at the test location. In this context, the expected contribution includes, but not limited to, collaboration with on-site and remote researchers, monitoring the progress of the experiment and status of the equipment, providing input to decision-making, and operation of the equipment where appropriate. Several IT-based technologies for Telepresence such as the Webshaker for live shake table experiments (http://webshaker.ucsd.edu/), Data Turbine, RDV and iSeismograph are aggregated through NEESsphere.

Education, Outreach and Training (EOT): NEESsphere aggregates content that provides EOT activities – in particular, by providing access to shared community content, online experiments, technical publications, etc.

Data Repository: Just like the NEEScentral application described in Section 2, NEESsphere enables access to metadata and data stored in the back-end database, and allows users to upload and download projects and experiments.

Computational Tools: NEESsphere provides a set of computations tools to enable a user to run simulations on distributed Grid resources. Portlets of particular interest are the HPC simulation portlets, e.g. the one for the OpenSees application (http://opensees.berkeley.edu/), which provides a gateway to the TeraGrid for "high-end" users to directly submit jobs on TeraGrid systems, the SimPortal portlet, which "bootstraps" the users into the OpenSees environment and provides a collaboration environment in the portal enabling them to publish their simulation outputs as datasets, and scientific workflow portlets leveraging the Kepler workflow environment to create more complex scientific pipelines.

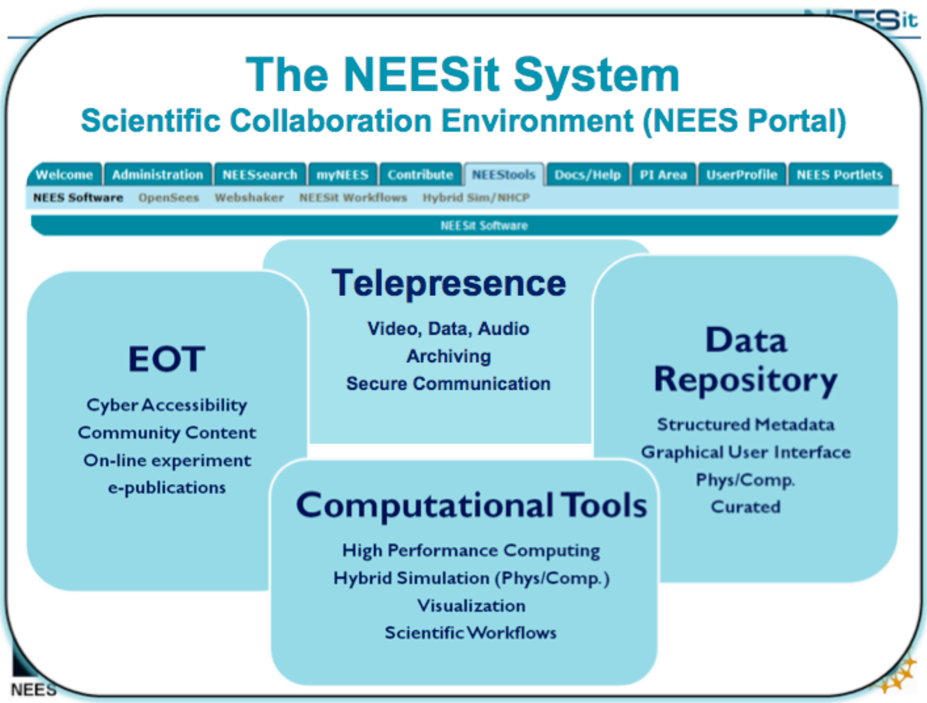

Figure 4: Component Overview of NEESsphere

The Grid Account Management Architecture (GAMA) provides account management and authentication for all users who wish to access the above-mentioned services. Authorization is provided on a per-resource level by comparing a user's Distinguished Name (DN) to a set of users allowed to access the resource.

NEESsphere represents the first prototype of a portal for the NEES community. However, several improvements remain to be made. In terms of access to data, NEESsphere only provides access to a single data source, i.e. the central NEES database. Like we discussed in Section 3, we would like to enable access to multiple data sources via the NEES Portal. If the diverse data sources employ different schemas for their data and metadata, the NEES Portal should provide tools for syntactic and semantic mediation of such data. In terms of computation, it should provide access to other simulation packages that are currently available on TeraGrid resources, such as Abaqus, ANSYS, LS-Dyna, and Matlab that may be useful to the earthquake engineering community. Furthermore, we need to make it easy for new applications to be contributed by the community, and deployed on distributed resources. These applications should be easily accessible to the rest of the NEES community. In terms of outreach and collaboration, we must make it easier for researchers to form online communities with the help of social networking features provided by the recent Web 2.0 tools, and to share experiment data with their colleagues securely.

5. CONCLUSIONS

Science and engineering communities are becoming increasingly dependent upon cyberinfrastructures for their research and education. NEESit is a cyberinfrastructure created to deliver IT tools and infrastructure to enable earthquake engineers to remotely participate in experiments, perform hybrid simulations, organize and share data, and collaborate with colleagues. In this paper, we discussed the current NEESit architecture, and the architecture of a new portal-based system for providing access to the NEESit tools. We also presented our initial prototype of the NEES Portal, which is based on portlet interfaces implemented within the GridSphere framework – a standards-compliant portlet framework for implementing Web portals. We discussed the various components of the framework, and the functionalities that we currently support and plan to support in the future. We strongly believe that portals and services oriented architectures (SOA) are the future of NEESit, and all cyberinfrastructures in general, due to flexibility, easy of implementation and use, and the ability to leverage a diverse set of information sources at the back-end.

REFERENCES

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 NSF Blue-Ribbon Advisory Panel on Cyberinfrastructure.

Fielding, R., (2000), Architectural Styles and the Design of Network-based Software Architectures, Ph.D. thesis, University of California.

Foster, I., & Kesselman, C. (Ed.) (1999). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, Inc.

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

Mueller, K., Bhatia, K., and Chandra, S. (2005). GAMA: Grid Account Management Architecture. *IEEE e-Science 2005*.

Novotny, J., Russell, M., & Wehrens, O. (2004). GridSphere: a protal framework for building collaborations. *Concurrency and Computation: Practice and Experience*, **16** (pp 503-513).

O'Reilly, T. (2005). What is Web 2.0: Design patterns and business models for the next generation of software. In O'Reilly.com, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

Warnock, T., Deng, W., Miller, L., and Lathers, A., (2005), The GridAuth Credential Management System, http://it.nees.org/documentation/pdf/gridauth_credential_management.pdf.