

*Building the National Virtual Collaboratory
for Earthquake Engineering Research*

NEESgrid

www.neesgrid.org

Last modified: March 3, 2003

NEESgrid System Baseline Description

SBD Version 0.9.5

Acknowledgment: This work was supported primarily by the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Program of the National Science Foundation under Award Number CMS-0117853.

Introduction

NEESgrid Overview

This document provides an overview of the software components under development within the NEESgrid Systems Integration Project. The particular decomposition of NEESgrid presented here illustrates the technical scope of the NEESgrid system to stakeholders in the NEES program.

This document provides a high-level introduction to NEESgrid's various components and capabilities, so that stakeholders can learn about system components that will be delivered on or before Oct 1, 2004.

The role of the NEESgrid system is to provide the software fabric that enables collaborative research within the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES). This fabric binds together the various components of the NEES project, as shown in Figure 1 below. Interactions among the NEES equipment sites, various computational and information technology providers, and earthquake engineering stakeholders are facilitated using the capabilities developed within the NEESgrid project.

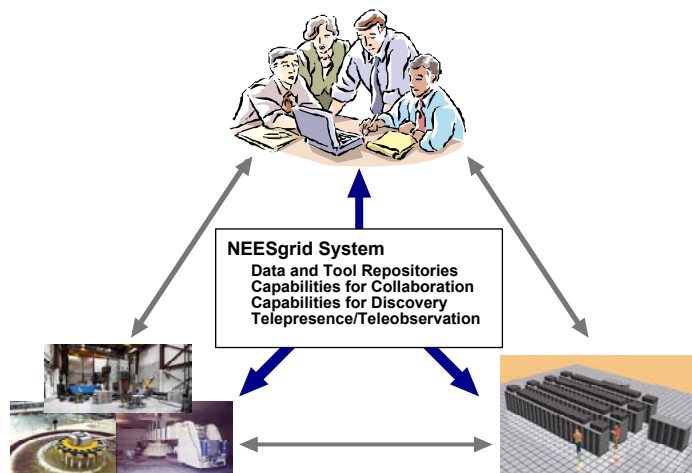


Figure 1: Role of NEESgrid system within the NEES project

The NEESgrid system must accommodate technological and cultural changes over the ten-year lifespan of the NEES consortium, so a flexible grid-based architecture forms the basis for the NEESgrid system. Besides promising users a consistently high level of performance, Grid technology is adaptable and highly extensible, permitting the continual addition of new sites and features to the system over the duration of the NEES consortium. Additionally, grid services software such as the Globus Toolkit provides dependable and secure access to a wide variety of resources. Finally, Grid infrastructure is robust because its development is for the most part open-

source; it is subject to continued collaboration and improvement. [For further information see Appendix A.]

Scope

NEESgrid is an IT system that connects physical experimental equipment sites around the country via software infrastructure. It also provides access to local and centralized earthquake engineering data repositories. Because of this unique part-real/part-virtual character of the NEESgrid system, it is essential to delineate the boundaries of the NEESgrid system in both the real world of physical networks and equipment sites, as well as within the virtual realm of software development and information content.

Network Scope

The network boundaries of the NEESgrid system are diagrammed in Figure 2 below. NEESgrid includes centralized facilities, distributed resources, and tools to manage connectivity for experiments and computational simulations. The NEESpop found at each experimental site mediates between the local network environment (including any proprietary hardware or software included within the experimental facilities) and the grid-oriented global network standards that form the foundation of the NEESgrid system. The NEESpop thus serves as a natural termination point for the network scope of NEESgrid, as it delineates the boundary between the local network resources of a NEES site and the larger network world of the Internet.

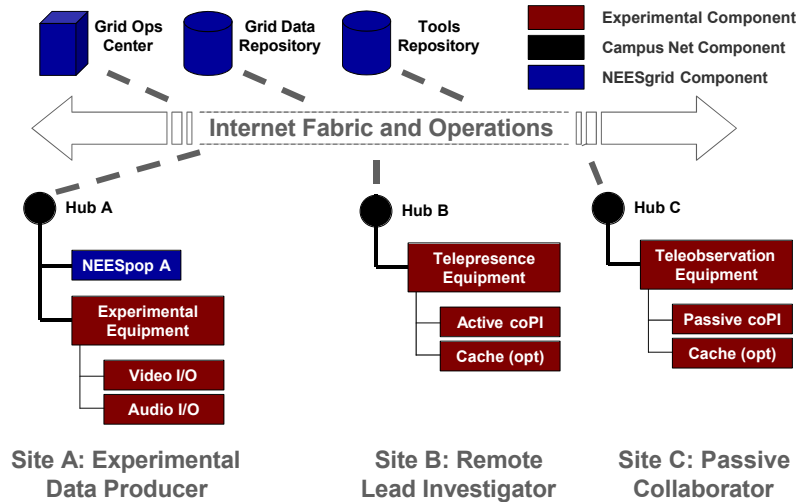


Figure 2: Network scope of the NEESgrid system

Software Scope

The software architecture of the NEESgrid system is idealized in Figure 3 below. The layered architecture of the grid middleware components is idealized into two architectural abstractions of interest to NEES stakeholders, namely (a) those grid components with functional requirements specific to

NEES, and (b) other more generic grid components. Software for the latter components are funded by various non-NEES projects (e.g., the National Middleware Initiative, the Globus Project), while software development for the former is a natural part of the NEESgrid systems integration effort.

Middleware services are abstracted for use in earthquake engineering applications via the Applications Programming Interface (API) layer shown. APIs provide a way to communicate between high-level application functions desired by end-users (e.g., desktop-resident tools to analyze a set of sensor data) and the lower-level functions required to make the NEESgrid system work. In order to facilitate development of end-user tools that aid community stakeholders in utilizing grid services, the NEESgrid system includes sample software implementations that demonstrate each software component. These tools are provided in either stand-alone form (generally written in C++ or Java), or as collaborative applications running within the CHEF software framework.

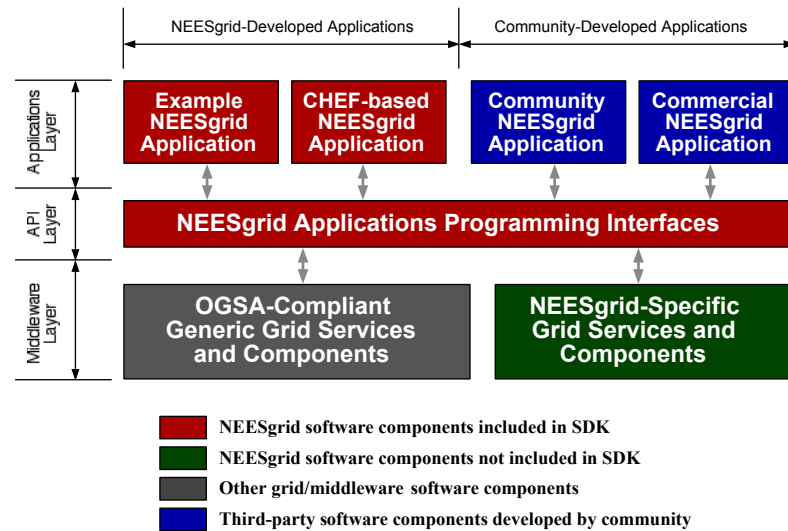


Figure 3: Software architecture and scope of NEESgrid

It should be noted that the idealization presented in Figure 3 is intended to show the architectural relations among the components developed under the NEESgrid project. The idealization is not intended to discourage software developers from using OGSA APIs directly, or to demonstrate exact architectural details of NEESgrid or NMI-funded software components.

NEESgrid SDK

The various APIs, documentation for these programming interfaces, and the sample NEESgrid applications collectively form the NEESgrid Software Developers' Kit (SDK). The use of domain-specific SDK's to promote

NEESgrid
Network
Operation Center

productive development of community codes is well-established in software engineering practice in both the open-source and commercial realms of software development, so this distribution mechanism provides a time-tested way to insure that innovations in NEESgrid software can be realized by the NEES community over the lifespan of the NEES project.

In the fully deployed system, software will be physically implemented at the NEES Equipment Sites and at the NEESgrid Network Operations Center (NOC). The NOC is located at NCSA for the duration of the NEES System Integration Cooperative Agreement. All NEESgrid users will be able to access capabilities, tools and services through the CHEF-enabled web portal. Software supporting security, teleobservation, teleoperation and data transport will be enabled at the Equipment Sites. In addition to core grid services, software supporting CHEF services, data management/curation services and data discovery services will be deployed to Equipment Sites. Core grid services, CHEF services, data management/curation services and data discovery services will be mirrored at the NEES NOC. In addition, the central data repository, numerical simulation code and simulation data repositories will be implemented at the NEES NOC. User interfaces and portals will be maintained at the NOC along with other repositories containing, e.g., documents and source code for NEESgrid components.

[For further information see Appendix A.]

NEESgrid
Repositories

The repositories supplied under the aegis of NEESgrid will provide for ready access by earthquake engineers to relevant data, tools, and technical documents designed to facilitate research, development, and practice. One of the most important benefits of a grid architecture is that replication of information is transparent to the users of NEESgrid, so that the specific topology of these repositories (i.e., whether they are distributed or centralized) is not necessarily exposed to the users of the system. In short, engineers can find their data and tools without having to worry about where that content resides, or how it needs to be transported for their ultimate use.

The tools developed for the NEESgrid system can facilitate setting up and maintaining additional repositories for earthquake engineering data, for use in collaborative research, education, outreach and practice. Community data may be made accessible via the application of NEESgrid data services and NEESgrid metadata standards that aid in discovery of community data content.

[For further information see Appendix B.]

Metadata Catalog

An essential element of the NEES data and community simulation code repositories are the metadata catalogs. Using simple search tools, end-users will be able to search the catalogs against any of their field to help them locate any testing information or information about simulation codes or modeling tools relating to the research problem. Locating published testing and numerical simulation data and all associated metadata allows researchers planning new projects to validate proposed designs or research programs via comparison with past work of similar nature. The system will also allow for discovery of simulation codes or modeling tools (from the community simulation code repository), and other NEESgrid participants with similar research interests (from the www.nees.org member database, and from the metadata catalog).

Data Repositories

There are two types of data repositories supported by NEESgrid, one central and the others located at the NEES equipment sites. NEESgrid discovery capabilities can be used transparently to access either type of data repository. Beside their location, the fundamental difference between these repositories lies in the processes used to support the quality and persistence of these data sets.

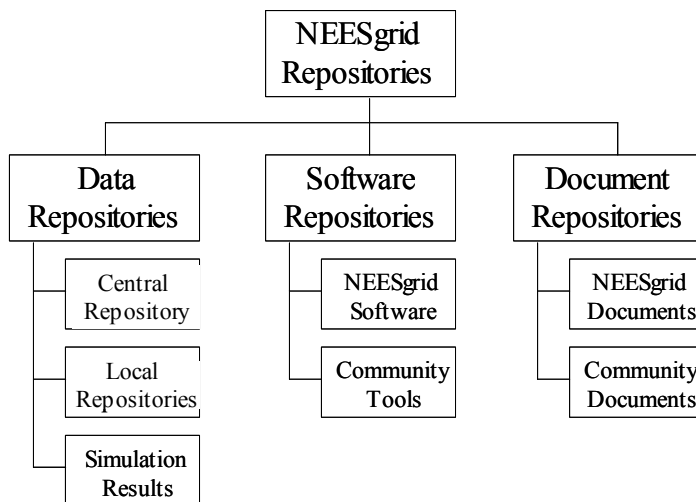


Figure 4: Structure of NEESgrid data repositories.

Central Repository

The central NEESgrid data repository includes all the experimental and simulation data produced as a by-product of shared-use processes at participating NEES sites. Curation permits management, organization, classification, archival, versioning, and various forms of quality-control processes that can operate on the data stored in the repository. This repository is part of the NEESgrid Network Operations Center.

Local Repositories

The users will also have the option to store their data locally at the NEES equipment site where their experiment is taking place. Typically, local repositories will have smaller capacity and not all the extensive services provided by the curated data repository. The local repositories will also act as backup storage for data during the tests, thus providing redundancy in case of network failure.

Simulation Results

NEESgrid data repositories will not only include experimental results, but will also provide for archival and discovery of results from computational tools used by the earthquake engineering community. Computational simulation results may include datasets generated by numerical simulations as well as derived results, such as visualizations, animations, or other representations of computed data. These results can be used in research, education, or public relations efforts on behalf of the earthquake engineering community. Metadata relevant to computational simulation will be an essential aspect of NEESgrid data models. NEES computational capabilities can also be extended by the community to include analysis software optimized for earthquake engineering applications and the utility tools that support problem setup and solution interpretation functions (e.g., visualization).

Software Repositories

In addition to the data archived and retrieved using models and processes developed as part of the NEESgrid project, analogous repositories of software tools and related software libraries will be developed for use by earthquake engineers. These will include applications specifically developed for use within NEESgrid, analysis tools of value to earthquake engineering researchers and practitioners, and example applications intended to help community members develop their own custom NEESgrid software. Earthquake engineers will be able to ascertain the target platform, function, and software quality of software in the repository via metadata standards developed within the data components of the NEESgrid project.

NEESgrid Software

All NEESgrid software will be accessible via the NEESgrid software repository. This collection of tools will include the specific software deliverables developed as part of the NEESgrid effort (e.g., example implementations of data ingestion and discovery tools, collaborative tools built within the CHEF framework, or other elements of the NEESgrid SDK), as well as community-driven enhancements to software in common use within the earthquake engineering community (e.g., portal interfaces for community analysis frameworks). Tools located within the NEESgrid software repository will include all those software applications officially sanctioned by the NEESgrid project.

Community Tools

The NEESgrid system is intended to help community members develop their own tools over the lifespan of the NEES project. For example, researchers working on transient soil-structure interaction problems can create a NEESgrid-accessible repository of tools that provide compatible analysis functions and data for this important multidisciplinary research topic. Subscription by community researchers to NEESgrid data and metadata standards and formats will permit earthquake engineers to discover software resources appropriate to their individual and collective needs, evaluate the function and quality of those software resources, and then to develop novel ways to solve the complex multidisciplinary problems that characterize the discipline of earthquake engineering.

Document Repositories

Document repositories are also an important deliverable of NEESgrid, and the design of these repositories will facilitate document versioning (i.e., the evolution of working documents will be preserved as a sequence of separate documents) and discovery of content via metadata searches. As in all the other repositories developed under NEESgrid, these document archives may be distributed in nature, but accessible to researchers as if all document content were centralized in a single location.

NEESgrid Documents

All documentation developed as part of the NEESgrid project will be accessible via the NEESgrid documentation repository. The content available via this document archive includes all project documents (e.g., project plans, requirements, architecture, etc.) as well as all training materials required for establishing a NEES site as a peer on the NEESgrid system. Documentation supporting the NEESgrid SDK will also be located in the NEESgrid document archive, so that all of the document content underlying the NEESgrid project can be found in a single unified repository.

Community Documents

The architecture of NEESgrid facilitates the collection of documents by individual researchers or teams of community stakeholders. The resulting distributed archive of earthquake engineering documents will include technical reports pertinent to NEES, documentation on community tools developed to support the NEES project, and other technical content intended to raise community awareness of the utility of the NEES MRE.

NEESgrid Capabilities

Overview

The NEESgrid system's unique capabilities are intended to facilitate novel applications in earthquake engineering via the promotion of new capabilities for discovery, collaboration, and remote participation in earthquake engineering research and practice.

Grid Services

The NEESgrid system is scalable, which permits the NEES community of laboratory sites to grow over time to include a wide range of different experimental capabilities and laboratory size scales. A scalable distributed grid computing network, however, must be highly stable, providing consistent and reliable access to remote users. It must also provide a secure environment for communications and data access and transfer. One important component of the NEESgrid system is the NEESgrid equipment site Point of Presence system (NEESpop), which performs a variety of "middleware" services: data management and caching services, resource management services, and security services.

The Globus Toolkit, considered the standard for the deployment of secure, robust Grid infrastructure worldwide, plays a significant role in NEES-POP operation security. Crucial elements include the Grid Security Interface (GSI), which addresses authentication and authorization issues; GridFTP, which enables secure data transfer, and Globus Gatekeeper, a secure resource manager. MyProxy service allows users to store secure credentials online in order to access the complex resources of the NEESgrid system via a single sign-on interface.

Another important NEES-POP component is the Meta Directory Server (MDS), which allows Grid resource discovery and monitors resource configuration and status, including machine load, network load, free disk space, and software versions, as well as site-specific information. In addition, the Network Weather Service (NWS), monitors machine load and external network connectivity and performance.

[For further information see Appendix C.]

Collaboration

Collaborative capabilities in NEESgrid are enabled using the CHEF framework under development at the University of Michigan. CHEF builds on the success of Michigan's long-standing efforts in the development and formation of research laboratories, and supports the development of innovative software that will enhance the collaborative capabilities of earthquake engineering researchers.

CHEF Framework

CHEF (CompreHensive collaboratiVE Framework) from the University of Michigan is an open source collaborative portal framework that provides a flexible and extensible standard software-development environment that will facilitate geographically-distributed educational, practice, and research collaborations. CHEF provides the unified web-based environment for users to access all NEESgrid tools (e.g. access to data repositories, telepresence, grid authentication/security). The extensible nature of CHEF facilitates its use as a tool for developing a broad range of applications, and its flexibility permits software developers and CHEF users to present various organizations of content within a single unified application (e.g., showing various representations of an experiment within contiguous tiled windows in a remote viewing application). The use of CHEF permits innovation within the NEESgrid software development enterprise, while facilitating reuse of existing tools developed by the earthquake engineering community.

At the same time that CHEF permits innovation, it is also easy to use, in large part because it allows users to access NEESgrid tools and resources through a single, intuitive Web browser interface. For example, using CHEF, a researcher new to NEESgrid will be easily able to sign on and create a customized workspace, read about specific relevant projects and general earthquake engineering community news, participate in forums, and establish valuable contacts using email, text chat, and message boards. Using the same interface, he or she will more easily be able to initiate partnerships with researchers at other institutions, set up experiments at remote earthquake test sites, and conduct and observe the experiments via streaming video and data servers. The user will also be able to access, browse, and download both metadata and experimental data.

CHEF Applications

Because CHEF is customizable, teamlets addressing needs as diverse as analysis and document versioning can be developed and tailored to the needs of the community. Additionally, many of the sample community applications developed as part of NEESgrid will be deployed as web-based CHEF applications. These include tools for telepresence, software applications that ingest data to various repositories for use by remote colleagues, and other functions that are collaborative in nature. These representative CHEF applications will thus form a solid foundation for the development of new and improved community software, and hence they form an essential component of the NEESgrid Software Developers' Kit.

[For further information see Appendix D.]

Telepresence

Telepresence capabilities are among the most important application drivers for the NEES community. These resources include support of remote observation, remote operation, and hybrid computational/experimental

capabilities that couple the real and virtual worlds of engineering simulation. The full range of experimental functions that will be enabled by these various telepresence resources are not known a priori, and so the discovery of suitable remote-participation capabilities is one of the key scientific goals of the NEES project.

Teleobservation

Teleobservation capabilities include passive remote participation in experiments via utilization of real-time video and experimental data streaming that permits observation by remote collaborators. Teleobservation protocols, hardware requirements, and remote-participation software permit NEESgrid system users to create unique collaborations that effectively remove distance as a barrier to collaborative examination of experimental data.

Telecontrol

Telecontrol capabilities extend the resources of remote viewing to permit active remote participation in experiments, including the control of experimental facilities from a distance. While this capability must be utilized with due consideration for site-specific personnel and equipment safety concerns, it is required for many of the unique experimental capabilities that can be developed using NEESgrid (e.g., for multi-site experiments).

Coupled Capabilities

Coupled resources of NEESgrid include hybrid capabilities for joining experimental sites, or for connecting experiments with computational simulations (e.g., for pseudodynamic testing of structures). The flexible NEESgrid information technology architecture promises to provide a broad spectrum of coupled and hybrid capabilities that can be extended by community developers over time, as the potential of a networked collaborative environment is discovered by the earthquake engineering community.

[For further information see Appendix E.]

NEESgrid Documentation

The capabilities of the NEES system need to evolve and grow over the lifespan of NEES consortium as new information technologies become available for use by NEES stakeholders. This iterative improvement of NEES capabilities can only occur in the presence of supporting technical documentation that explains NEES functions and permits researchers to learn about NEES information resources so that these collaborative capabilities can be understood, maintained, and extended.

Development

Software development documentation is essential for the transition of NEESgrid capabilities from a specific systems-integration function to a general-purpose set of community capabilities. The software resources that

will be required by NEES users over the ten-year duration of the NEES consortium are impossible to predict, so that the only reliable means to insure they are realized is to support community efforts to design, develop, and deploy software tools that address near- and long-term community needs. The documents required to insure that these needs are met include example implementations of NEESgrid software, documentation for development tools, annotated APIs, and the collection of sample source code and documents within the NEESgrid SDK.

Annotated APIs

The NEESgrid Applications Programming Interfaces facilitate the abstraction of underlying grid services into well-defined interfaces that support development of NEESgrid-resident applications. In addition to good interface design, effective utilization of such computational interfaces hinges on the availability of high-quality documentation that fully explains the data structures and behaviors of the methods included in the various APIs.

NEESgrid SDK

The NEESgrid Software Developers' Kit provides a single distribution of all the relevant software implementation, applications interfaces, and documentation required for developers to write new software tools that utilize NEESgrid information services. The NEESgrid SDK thus represents a convenient and coherent packaging mechanism that encapsulates the appropriate sample code, tools, and annotated interfaces required by NEESgrid developers to write applications software using NEESgrid APIs.

Example Tools

The current release of NEESgrid Tools and Services includes tools that are intended to demonstrate the software's capabilities but are not required. For instance, LABView is included in the current release as the example Data Acquisition System (DAQ), but sites will be able to develop interfaces for other DAQs.

These example implementations are provided to permit NEESgrid software developers (including community developers and other NEES stakeholders) to grasp the fundamentals of NEESgrid systems programming using well-defined applications that are limited in scope, but that amply demonstrate individual NEESgrid system functions (e.g., data ingestion, collaborative functions, etc.)

[A more detailed description of the current example DAQ implementation may be found in Appendix F.]

Training and Support Materials

NEESgrid is a complex distributed system, and while grid technology facilitates hiding much of this complexity from the users of NEESgrid, there is a considerable amount of complicated work that is concomitantly the responsibility of systems administrators at NEESgrid sites. Systems administration support materials permit those IT professionals who deploy

and maintain NEESgrid sites to learn the details of NEESgrid services, and to develop their skills as capable NEESgrid systems administrators. Developers wishing to create application interfaces for their respective sites will also require adequate orientation and support.

To this end, a series of training workshops will be offered at NCSA targeting the following groups:

- Earthquake engineers who will use NEESgrid for their research. The training will focus on the CHEF interface and the functionality of the NEESgrid system. They will only need a basic understanding of the NEESgrid architecture.
- System administrators from NEES sites, who will learn how to install, configure and maintain the standard NEESgrid software packages.
- Applications developers from NEES sites who need to interface their equipment to the NEESgrid. The training will therefore concentrate on the Application Interfaces (APIs) of the NEESgrid system, with some concrete examples of data acquisition systems. These NEES programmers are also encouraged to attend dedicated CHEF workshops organized by University of Michigan in order to learn how to enhance and customize the end-user interface to NEESgrid.

The format of these seminars will be a combination of lecture, discussion, and hands-on experience. Participants will be encourage to bring and work on their own projects during the workshops and will have an opportunity to evaluate their progress both during and in the months following the workshop. All presentation and other training materials will be made available on the Web after the workshops.

Migration Tools

The original NEESgrid system involves only the NSF-funded NEES equipment sites, yet many other experimental and computational sites are expected to join the NEES collaboratory over the lifespan of the NEES consortium. Migration tools and documentation will facilitate the deployment of these new members of the NEES community, by helping local system administrators and earthquake engineers develop and implement feasible strategies to migrate their local resources into the larger networked world of the NEES collaboratory.

Administrative Documents

Administrative documents accessible to the NEES community include system integration work plans, user requirements, technical whitepapers, and other content prepared for the design, development, and deployment of the NEESgrid system. To be fully effective as a working collaboratory, it is essential for all relevant NEESgrid administrative documents to be made available to the NEES community, so that earthquake engineers interested in joining the NEES enterprise can plan for the work involved in becoming full stakeholders in the NEES project. The versioning of documents in the administrative repository will permit the NEES community to understand and appreciate the iterative nature of information-technology systems design.

Internal SI Documents

All relevant internal systems integration documents, including the project execution plan, the user requirements content, the systems architecture description, and supporting documents such as risk management guidelines, will be made available (including versioning) on appropriate NEESgrid document repositories. In this manner, all appropriate NEES stakeholders can examine NEESgrid planning and scope documents, which facilitate responsiveness on the part of both the earthquake engineering community and the systems integration team.

SI Whitepapers

The collection of NEESgrid whitepapers will be available (including versioning, where applicable) within an appropriate NEESgrid document repository. These whitepapers include technical content on NEESgrid components (e.g., the NEESpop, various other servers/services, etc.) that are relevant for NEES community members interested in developing new NEESgrid capabilities, administering new NEES sites, or understanding existing NEESgrid software tools.

Further Information

This document presents only a skeletal description of NEESgrid resources and capabilities. More detail can be found in the various NEESgrid administrative documents, including the NEESgrid Project Execution Plan, the User Requirements Document, the Systems Architecture Document, and the various NEESgrid technical whitepapers.

Refinements to the content in this document may be found on the NEESgrid.org website, and information describing the state of the NEESgrid system's implementation will be found in the NEESgrid configuration management system.

Appendix A: General Information

Revised Project Execution Plan

Revised technical, cost, and schedule baseline document for the development and deployment of NEESgrid.

http://www.neesgrid.org/documents/NEESgrid_PEP_Feb15.2002.pdf

NEESgrid System Overview 2.1 (Tom Prudhomme)

Describes the NEESgrid system as seen, functionally, from the perspective of an earthquake engineering researcher. The paper covers the life cycle of a proposed NEESgrid-based experiment, through NEESgrid-enabled remote collaboration, from initial design through final publication.

http://www.neesgrid.org/documents/NEESSystemOverview2_1.pdf

NEESgrid System Architecture v1.0 (Carl Kesselman, Randy Butler, et al.)

This document, from February 2002, describes system architecture and core capabilities and services provided by the NEESgrid system.

http://www.neesgrid.org/documents/NEESgrid_SA_Feb15.2002.pdf

Technical Report NEESgrid-2002-02 NEESgrid Early Adoption and Implementation Plan (Sridhar Gullapalli, Carl Kesselman, et al.)

This document lays out a roadmap for the planned development and early adoption phase of the NEESgrid project. It covers the period from April 1 to October 1, 2002. The results of these efforts will be the deployment of an early NEESgrid infrastructure with limited functionalities, with a view to demonstrate an initial set of capabilities.

http://www.neesgrid.org/documents/NEESgrid_TR.2002-02.pdf

Other links:

NEES project at the National Science Foundation:

<http://www.eng.nsf.gov/nees/>

NEES Consortium, Inc.

<http://www.nees.org>

Globus Project

<http://www.globus.org>

National Science Foundation Middleware Initiative

<http://www.nsf-middleware.org>

Appendix B: Data Repository

Technical Report NEESgrid-2002-04 NEESgrid Data and Metadata Harvesting Protocol Whitepaper (Joe Futrelle, Jeff Gaynor, NCSA)

Provides an overview of the NEESgrid Data and Metadata Harvesting service, which will provide NEES users with the ability to transfer data and metadata to the central repository. It will also enable users to manage the data and metadata in a variety of ways, including organizing data and metadata objects into projects, updating and removing data and metadata objects, and controlling who has access to the objects.

http://www.neesgrid.org/repository/NMHPwhitepaper_v1_0_2.pdf

Technical Report NEESgrid-2002-05 The NEESgrid Metadata Service API: Overview (Joe Futrelle, Jeff Gaynor, NCSA)

Describes the architecture and client API of the NEESgrid Metadata service API. The client API is a Java class library providing remote access to the Metadata Service. The API enables clients to do several things such as authenticate on behalf of a user, request full or partial sets of attributes of metadata objects, retrieve multiple versions of the same object, resolve inter-object references, update objects and request upload or download of data files.

http://neesgrid.org/repository/MetadataService_v1_0.pdf

NEESgrid Strawman Metadata Model v1.0 (Joe Futrelle, NCSA)

This document describes an initial, "straw man" metadata model specification intended as a point of departure for broad discussion of the NEESgrid data and metadata models and formats, and for prototype implementation in the Early Adopter program.

http://www.neesgrid.org/repository/StorageTechnologies_v1_0.pdf

Appendix C: Grid Services

Technical Report 2001-04: NEESpop White Paper (Randal Butler, Ian Foster, and Carl Kesselman)

This white paper describes the functions of and the requirements for the NEESPOP, a point-of-presence server at each equipment site for accessing and communicating with the NEESgrid.

http://www.neesgrid.org/documents/NEESgrid_TR.2001-04.pdf

Technical Report NEESgrid-2002-01 NEESgrid Local Area Network Requirements (Laura Pearlman, ISI)

This draft document is intended to help NEES equipment sites to ensure that they have sufficient network and computing equipment and personnel to support NEESgrid services.

http://www.neesgrid.org/documents/NEESgrid_TR.2002-01.doc

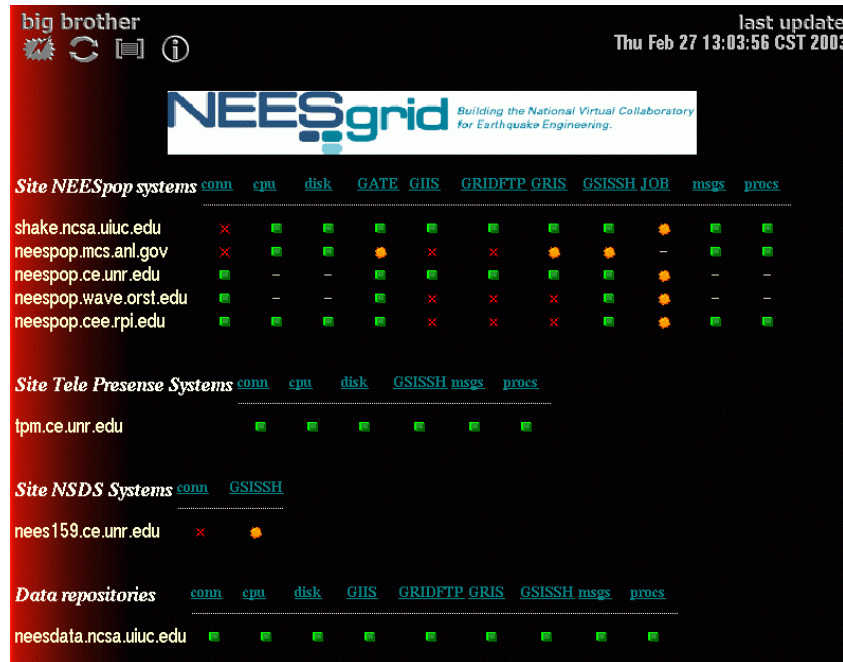


Figure C-1: The NEESgrid Systems Status page, provided by the network monitoring application Big Brother, gives viewers a live, updated view of the entire NEESgrid network. (<http://neespop.ncsa.uiuc.edu/bb/>)

Technical Report NEESgrid-2003-01: Integrating Grid Capabilities into the CHEF Collaborative Portal Framework (Chuck Severance, School of Information, University of Michigan)

This paper discusses the design for integrating support for the grid into CHEF, the web-based Comprehensive Collaborative Framework for remote collaboration. The goal is to allow grid-oriented tools to be built into the CHEF framework and operate within a grid security context.

http://www.neesgrid.org/documents/Integrating_Grid_into_CHEF-v01.pdf

Description of the CHEF/Jetspeed API included in the NEESgrid alpha software release

Major system dependencies for this component (i.e. third-party software, LabVIEW, Tomcat, GTK)

- Tomcat
- Ant
- Java JDK 1.4

Services provided:

- NEESgrid Portal Framework
- Single Sign-on to the Grid for the NEESgrid Web portal

Interacts directly with:

- Data repository code as provided by NCSA

Language:

- Java

Additional features:

- Extensible. Sites may choose to develop local software, which they can deploy in CHEF.

Documentation:

<http://chefproject.org/> includes API documentation and other supporting documentation for using/developing CHEF.

Future modifications:

The primary new feature for the final release will be a version of CHEF, which allows the federation of multiple CHEF servers into a single image. This will include distributed single sign on and distributed group membership. This will require no changes to any of the APIs – it is simply improved infrastructure within CHEF.

Appendix E: Telepresence

NEESgrid TelePresence System Overview: White Paper (Nestor Zaluzec, ANL)

The NEESgrid TelePresence System Overview: White Paper (PDF) is intended to describe to Equipment Sites the NEESgrid TelePresence Mode (TPM). It describes the basic system components and hardware requirements for data and video streaming and capture. This is a draft document.

<http://www.neesgrid.org/documents/TPMOverview20020306.pdf>

NEESgrid Imaging Overview: White Paper (Nestor Zaluzec, ANL)

The NEESgrid Imaging Overview: White Paper is a general discussion of the issues involved with streaming video technology, such as resolution and frame rate, as related to the TeleObservation component of the NEESgrid project.

<http://www.neesgrid.org/documents/ImagingOverview20020313.pdf>

Appendix F: Data Acquisition Components (DAQ)

LabVIEW Code Installation Instructions (Paul Hubbard, ANL)

Covers downloading, installation and configuration of the NEESgrid LabVIEW code for data acquisition.

<http://www.mcs.anl.gov/neesgrid/daq-install-instructions.pdf>

Driver installation instructions (Paul Hubbard, ANL)

Covers the download, installation and configuration of the NSDS driver. This is a small piece of C code that mediates between the DAQ and the NSDS, and is required for all NEESgrid sites.

<http://www.mcs.anl.gov/neesgrid/driver-install-instructions.pdf>

NSDS / driver / DAQ protocol specification and system design (Paul Hubbard, ANL)

Covers the protocol used between the DAQ, driver and NSDS. This is primarily of interest to sites wanting to write their own driver for non-LabVIEW data acquisition systems. It might also be of interest to persons curious about the system design and internals.

<http://www.mcs.anl.gov/neesgrid/driver-install-instructions.pdf>

Description of the DAQ API included in the NEESgrid alpha software release

DAQ Library

Major system dependencies for this component

- LabVIEW 6.1
- LabVIEW Internet Toolkit

Services provided:

- This is the main code library that the users will call from their DAQ applications.
- Metadata read / write / upload / download
- FTP upload and download, with repository protocol
- Save to disk
- Data streaming to driver
- ISO 8601 timestamps

nsds-driver

Interacts with:

- DAQ daemon
- NSDS driver

Development environment:

- LabVIEW

Documentation:

<http://www.mcs.anl.gov/neesgrid> includes install and customization documentation, as well as NEESgrid-specific data protocol.

Planned changes in future releases:

Components for active control. More information, including details about configuration and additional APIs, will be available in the finished protocol specification

Major system dependencies for this component:

- **To build:** make, C compiler, libgetopt, POSIX API
- **To run:** working DAQ and NSDS servers

Services provided:

This is the DAQ abstraction layer, intended to allow sites to run non-LabVIEW-based DAQ systems. This version is written both as a template for them and as a driver for our DAQ. It is intended to be customizable.

Interacts with:

- DAQ daemon
- NSDS

Language/development environment for the component:

- ANSI C, roughly C9X level, POSIX API.
- Tested on Linux (Redhat 7.2, 7.3, 8.0 and Debian) and Mac OS X 10.2.

Documentation

<http://www.mcs.anl.gov/neesgrid/> includes the protocol specifications.

Documentation of the code itself (using Doxygen) is both distributed with the source and available at the abovementioned URL.

Users can also access help via command line: ./driver --help

Future modifications:

nsds-driver will be modified to support the control protocol presently being written at ISI.

Currently, you need to know the name or IP of your DAQ and NSDS server machines. If they've been modified to use non-standard ports, you need the

DAQ daemon

port numbers as well. This may be extended, depending on how the control system is defined.

Major system dependencies for this component:

- LabVIEW 6.1, on Windows, Mac or Linux
- **To run:** Same as above
- **To stream data:** Working driver and NSDS

Services provided:

- Listens on two TCP ports and talks to the NSDS driver.
- Handles the list of open channels, connection management, metadata such as sample rate and channel units, and all other control channel tasks.

Interacts with:

- DAQ library code, NSDS driver

Development environment:

- LabVIEW

Documentation:

<http://www.mcs.anl.gov/neesgrid/>

Future modifications:

Presently the daemon handles DAQ; it may also be extended to handle active control as well.

Miscellaneous DAQ code

Major system dependencies for this component:

- LabVIEW 6.1
- LabVIEW Internet Toolkit

Services provided:

- NSDS simulator, DAQ hardware simulator
- NSDS stress tester, Fake DAQ stress tester
- Example DAQ code (DMA-driven)
- Fake DAQ (useful on systems w/out DAQ hardware)

Interacts with:

- DAQ daemon
- NSDS driver

Development environment:

- LabVIEW

Documentation:

<http://www.mcs.anl.gov/neesgrid/>

Future modifications:

Internal testing and example code will be expanded as new capabilities are added to the codebase.