

NEESgrid Simulation Component: OU Subaward Summary Final Report

Kyran D. Mish

School of Civil Engineering and Environmental Science
University of Oklahoma, Norman, Oklahoma

Jin-song Pei

School of Civil Engineering and Environmental Science
University of Oklahoma, Norman, Oklahoma

Lee. M. Taylor

Terascale LLC,
Cedar Crest, New Mexico

Table of Contents

<i>Overview</i>	3
The Role of Simulation in the NEES Project	3
Uses of Simulation in Earthquake Engineering	4
Task Descriptions	5
<i>Report on Task 2.3.1</i>	7
General overview of modeling tools and techniques	7
Sample Prototype Archive Content	8
Task 2.3.1 Path Forward	10
<i>Report on Task 2.3.2</i>	10
The Role of Digital Content	11
Prototype Community Needs	11
Task 2.3.2 Path Forward	13
<i>Report on Task 2.3.3</i>	13
General need for portal access to codes	13
Prototype Portal Models for Evaluation	14
Task 2.3.3 Path Forward	19
<i>Conclusions and Future Work</i>	19
<i>Appendix A: A Plan for an OpenSees Computational Simulation Component</i>	21
Components of the Computational Subsystem	21
Integration Requirements	23
Other Considerations	24

Overview

This section provides an overview of the Simulation Component of the NEESgrid project. The simulation component of NEESgrid has long been criticized by members of the earthquake engineering community as insufficiently ambitious, and the recent change of Principal Investigator on the NEESgrid award has created a great opportunity to forge a more responsive path towards a community-driven model of computational simulation. This summary report provides an overview of the results of past plans for the NEESgrid simulation component, and also provides suggestions for how existing and proposed NSF-funded work can be leveraged to build a computational portal for the NEES MRE that is fully responsive to the real needs of the earthquake engineering community.

The Role of Simulation in the NEES Project

There is considerable discussion in the earthquake engineering community as to the role of computational simulation within the George E. Brown, Jr. Network for Earthquake Engineering Simulation, a.k.a. NEES. Some take the term “simulation” to mean laboratory experimentation, in which case the experimental facilities constructed with NEES funding implicitly define the term as “laboratory experimentation”. Others take the term to mean “computational simulation”, and especially such computation that supports the design and execution of large-scale physical experiments.

Webster’s Universal Unabridged Dictionary defines “simulation” as:

“imitation or enactment, as of something anticipated or in testing”

This definition is clearly relevant to the larger mission of NEES, namely the laboratory testing of physical experiments, for which NSF will spend approximately \$70M on fifteen different equipment sites. But Webster’s also provides this alternative definition:

“the representation of the behavior or characteristics of one system through the use of another system, especially a computer program designed for the purpose”

This alternative definition suggests a more ambitious role for computational simulation within the more general term “simulation”, so that the terminal “S” in “NEES” might be taken to stand for computational simulation instead of the more general sense of computational and laboratory simulations.

But given the relative amounts of funding expended so far on experimental versus computational simulation (i.e., more than two orders of magnitude more funding is devoted within NEES to experimental equipment deployment than to computational simulation), it is clear that the near-term role of computational simulation within the NEES project will be constrained by available funding. This realization is an essential element of the proposed path forward presented at the end of this document.

Uses of Simulation in Earthquake Engineering

There are four primary roles for computational simulation in relation to laboratory experiments in earthquake engineering, namely:

- (1) a priori simulation in support of experimental design optimization
- (2) a posteriori simulation in support of experimental interpretation
- (3) concurrent simulation that permits hybrid numeric/laboratory testing
- (4) purely computational simulation that replaces experimental efforts

The first role is an essential element of any large-scale experimental enterprise, because current laboratory experiments in earthquake engineering are large and complex systems, which benefit greatly from a priori computational simulation efforts performed to optimize the associated experimental designs, e.g., determining proper locations of sensors, predicting accurate estimates of significant physical responses, estimating time and financial resources required to construct and deploy the experiment, etc. This first role is facilitated by a wide range of computational mechanics packages, some of proprietary nature (e.g., ABAQUS, ANSYS, ADINA, LS-DYNA), others arising from community public-domain efforts (e.g., OpenSees).

The second role is a traditional one for computational simulation, and it generally involves the use of special-purpose tools for data reduction, data mining, and solution interpretation, e.g., interactive visualization applications capable of rendering complex physical systems used in engineering fields. While the second role is one of long standing in all engineering communities, there are many important avenues of research in this arena (especially those involving the effective mining of experimental and computational data) that are largely open research questions in need of substantial future research and development efforts.

The third role represents an emerging opportunity to fuse computational results with experimental testing, and is already commonly used in many areas of earthquake engineering research, e.g., pseudodynamic testing, where the inertia of the structure is modeled using the computer so it can be re-applied to the structure quasi-statically. Within the distributed nature of the NEES project, this role represents one of the most exciting research venues for computational simulation in structural engineering.

The fourth role is gaining in importance, and will be important in the future of earthquake engineering, but its full utility is currently hampered in many cases by imprecise knowledge of the relevant physics (e.g., soil liquefaction problems) or by uncertainty in material or geometric information (e.g., tsunami models arising from deep-ocean earthquakes). Where it is possible to gain an accurate understanding of the physics of the problem, it is possible to model large and complex problems on the computer that cannot reasonably be simulated using current experimental techniques.

This role was one of the motivating principles behind the funding of the NEES project, and the range of problems where computational simulation can serve as an equal partner to laboratory experimentation continues to grow with time.

One example of the fourth class of computational simulation is shown in the figure below, where a finite-element transient analysis of a concrete dam in Colorado is shown¹. In this model, the foundation (including near-field earthquake motions), the dam, and the reservoir are modeled using appropriate finite-element approximations, and the entire system's dynamic response is computed using a massively-parallel distributed-memory supercomputer.

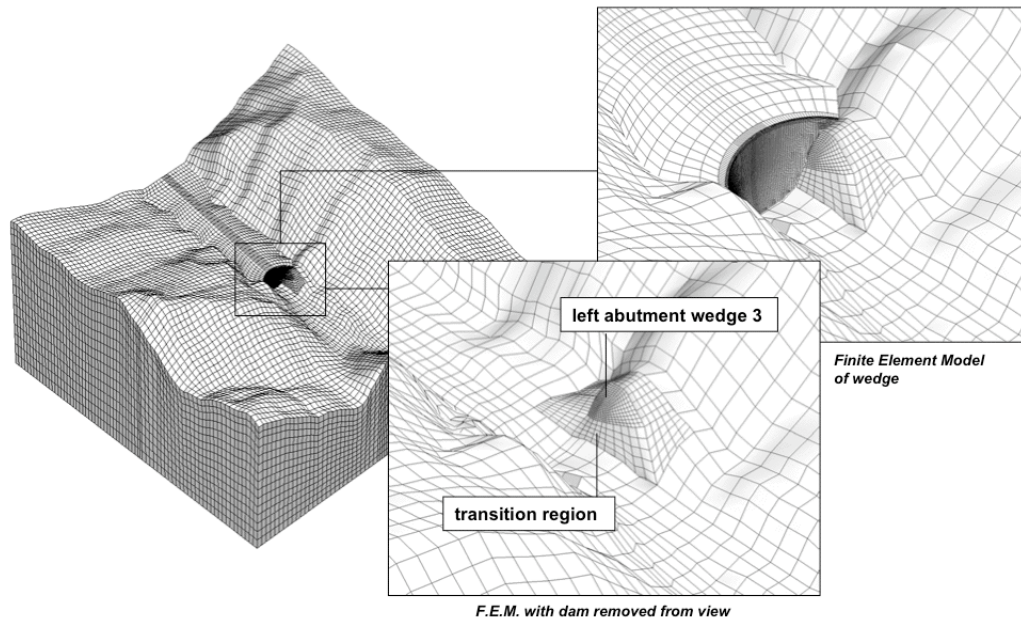


Figure 1: Finite-Element Mesh for Morrow Point Dam

As of today, such large-scale computational problems are difficult to set up, expensive to model, and notoriously intractable to validate. The latter characteristic is of course due to the fact that such experiments are inherently difficult to perform in the real world, so the characteristics that make this class of computational simulation problems meaningful also make this class of problems extremely difficult.

Task Descriptions

The three tasks that form the basis of the NEESgrid Simulation Component are these:

1. the prototypical collection and dissemination of software tools useful in earthquake engineering research and practice, including a prototype repository of links to available software supporting the use of computational simulation in the field of earthquake engineering. Because most software tools are maintained and archived locally by their development teams, this collection effort is arranged as a web portal

¹ This work was performed by Chad Noble of the New Technologies Engineering Division of Lawrence Livermore's Engineering Directorate, under the supervision of David McCallen of LLNL

of links to existing software resources, instead of as a centralized source code repository.

2. The prototypical collection and dissemination of content produced by those tools, including related technical documents, presentations, graphics and visualizations. This content will be disseminated based on large-scale topics (e.g., systems identification in the prototype version of this content site), as well as on smaller-scale how-to documents that illustrate specific methods of analysis.
3. prototype extension and targeted enhancement of community codes, so that advanced technology developed under the NEESgrid project (e.g., grid-mediated access to high-performance computing resources, CHEF-enabled collaborative tools) can be used to increase the value of existing and future applications used by earthquake engineering researchers and practitioners. This arena is essentially open-ended in scope, so much of this work needs to be done via coordination instead of via construction.

These strategies are used to derive the underlying simulation subsystem for NEESgrid, as shown in the figure below (from the NEESgrid Systems Architecture documentation).

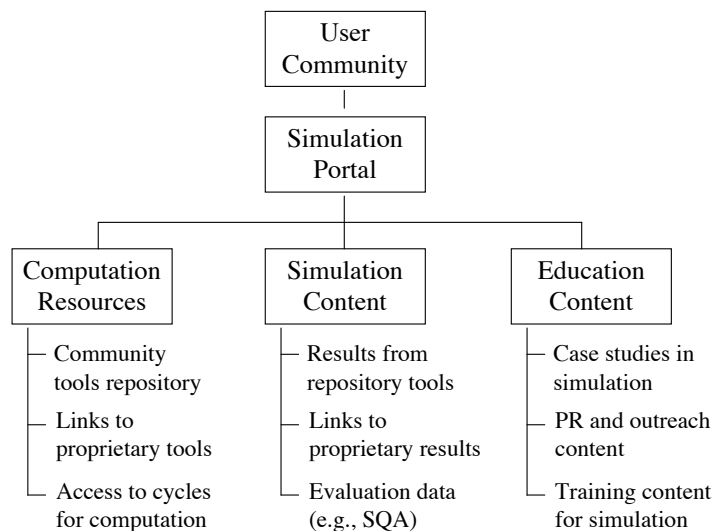


Figure 2: Simulation Subsystem Architecture

The portal provides for uniform access to simulation tools, simulation results, and education and public-outreach content, and will eventually be realized as a CHEF application that has the same look-and-feel as all other NEES collaboratory tools. This access will be supported by various capabilities that augment the value of the community portal, including searches based on data and metadata information, and computational resources where community software tools can be located, obtained, and effectively utilized.

In practice, researchers and practitioners of earthquake engineering require collaboration tools and content that are cross-cutting by nature, so that the task decomposition described above is an imperfect one, in that the computational simulation aids required by the community borrow from more than one area at a time. For this reason, a Simulation Working Group was assembled from leading researchers in engineering computation for defining the initial direction of the Computational Simulation effort, and the need for higher-level cross-cutting content (such as that described below, in Task 2.3.2) was clearly articulated from within this working group.

Report on Task 2.3.1

Task 2.3.1 provides for a prototype archive that permits browsing of relevant software resources available for use in earthquake engineering. The original vision of this task was that the archive would serve as a software repository for open-source community code developers in earthquake engineering, but input from the community quickly and clearly demonstrated that what was desired was not a centralized archive of downloadable software, but an integrated and extensible collection of links to various software resources, with appropriate overviews and links to gain more information. Since many of the applications used in engineering practice (e.g., ABAQUS) are proprietary and hence not available for downloading from a central NEES software site, this more distributed archive-of-links approach more readily matched the reality of how software resources are used in current earthquake engineering practice.

General overview of modeling tools and techniques

Earthquake engineering analysis tools are generally derived from a finite-element formulation, as such analysis methods are widely known and well-developed in the earthquake engineering community. Since so many earthquake engineering physical responses are nonlinear in nature (e.g., soil liquefaction, structural ductility, fluid-structure interaction), it is essential for researchers and practitioners in earthquake engineering to understand key concepts involved in the use of nonlinear transient finite-element models, including the role of software quality (e.g., appropriate verification and validation techniques), the qualitative nature of feasible physical response in nonlinear mechanics problems, and the proper choice of element, solution, and analysis types for a given class of problems. In addition, issues such as cost, scalability², and availability for a given computer platform are of great interest to earthquake engineers.

Three particular classifications are of especial interest to engineers considering using a particular software application for their work:

² scalability refers to the effective use of additional computer resources so that the scale (i.e., fidelity) of the problem can be increased by utilizing additional resources, e.g., more processors, memory, etc. In practice, the resource generally identified with the term *scalability* is the number of processors used in the analysis, so that *scalable* can be taken to mean “capable of parallel programming efficiencies”, e.g., if N processors are used to compute the results of an analysis, the time required to perform the analysis would decrease by a factor approximating 1/N, relative to the time spent performing the analysis on a single processor.

- Commercial-grade production applications (such as ABAQUS, LS-DYNA, or ANSYS), which implies a high level of software quality and thus attendant confidence that a correct analysis method will result in correct analysis results,
- Open-source or community applications, which are by definition free, but which may or may not include a formal software testing plan to insure that the resulting analyses are error-free (it should be noted that open-source software is often among the least bug-prone to be found anywhere, due to the fact that everyone can inspect the source code and propose fixes for any errors found there – for example, in the world of operating systems software, the open-source Linux operating system is among the most robust and reliable software applications ever developed, so it is possible for open-source software to be both free and of excellent quality), and
- Framework applications, where much of the underlying computer science technology has been cleanly separated from the engineering analysis software technology, so that users of the software application can concentrate on engineering principles instead of on underlying computer science implementation details. Frameworks are especially useful on more complex problems, e.g., multiphysics problems such as soil liquefaction or coupled problems such as wave-structure interaction effects, because of the higher-level orientation found in a software frameworks environment.

Because of the importance of software quality and methods development in the application of earthquake engineering tools, the prototype archive from Task 2.3.1 includes the capability to evaluate software quality and other relevant software tool metadata, e.g., cost, supported computer platforms, etc.

Sample Prototype Archive Content

Sample content from the prototype software tool archive is shown below. Figures 3 and 4 demonstrate sample pages from the tool archive, while Figure 5 shows a page from a backgrounder on effective software development using computational frameworks. Figure 3 presents information related to a popular commercial software package (LS-DYNA), while Figure 4 presents information related to a popular open-source framework (OpenSees, from the PEER project).

Each page of application description content is intended to summarize the general capabilities and salient features of a given software tool. The archive permits finding tools by alphabetical order of name, by associated set of features (e.g., a structural dynamics program), or via special news notes that highlight a specific software development effort or technique.

Ultimately, the software tool archive ought to have a rich metadata structure associated with individual tools, so that engineers can browse and search more easily through the accumulated collection of various tools. At present, standards for such metadata do not yet exist, though this limitation is becoming better-known, and hence this is a natural arena for research and development towards improved usability of the NEES system.

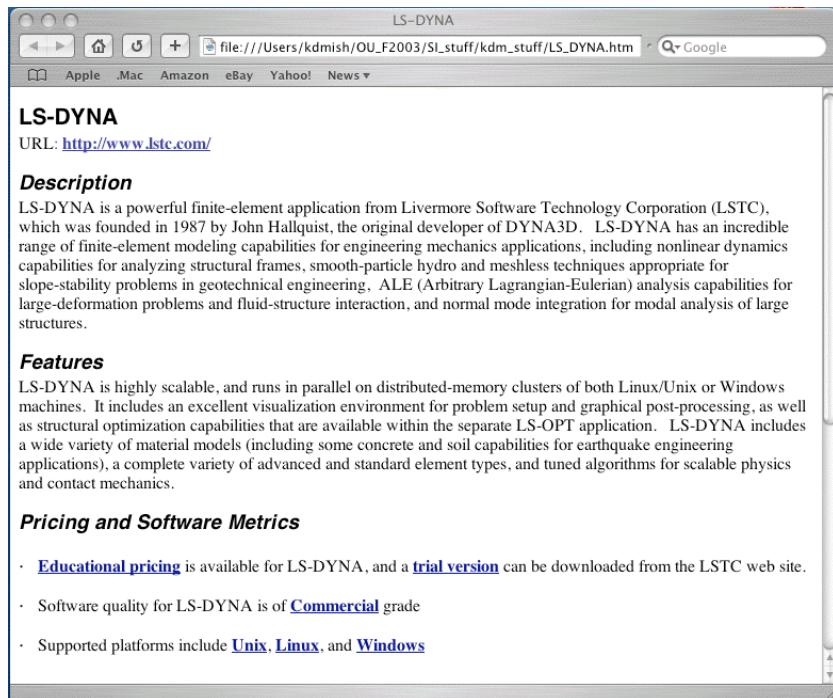


Figure 3: Example Code Archive Page (LS-DYNA)

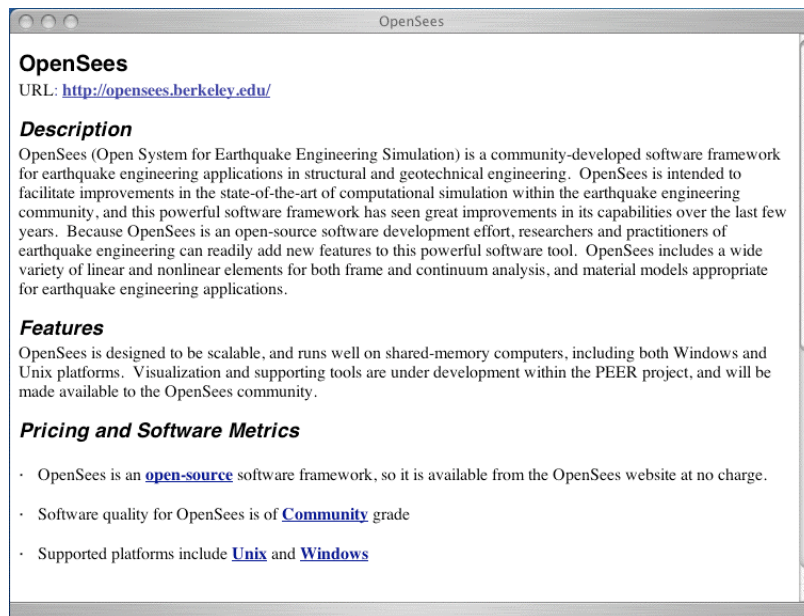


Figure 4: Example Code Archive Page (OpenSees)

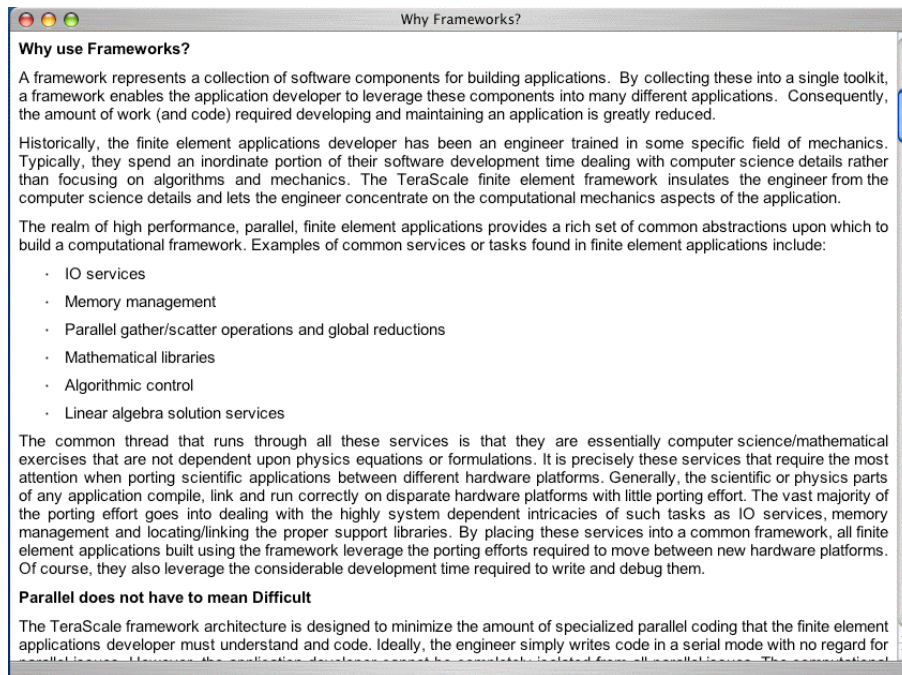


Figure 5: Example Code Archive Page (Frameworks Whitepaper)

Task 2.3.1 Path Forward

The software resource archive has been implemented only in a prototype format, based on web searches performed by students using earthquake-engineering-specific search criteria (these searches have been regularized, and then augmented by specific knowledge of commercial software tools such as LS-DYNA). The resulting archive thus represents potentially only the tip of the iceberg of available software resources, and the current content for tools that can be found in the archive may omit new features that have not been publicized over the web or within the community yet, e.g., extension of a serial version of an application to a newer scalable version.

Thus the most important element in the path forward for Task 2.3.1 is widespread dissemination of the existence and form of the code archive, so that interested software developers can add descriptions of their products to the community software resource base, as well as links to relevant results generated by their software tools.

Report on Task 2.3.2

Task 2.3.2 provides for a prototype archive of content developed using various available software tools, so that researchers and practitioners in earthquake engineering can find content that can be used for training, education, outreach, validation, or public relations efforts. This task is thus superficially similar to Task 2.3.1, but targeted towards content resources instead of software resources.

The Role of Digital Content

From a practical standpoint, digital content (i.e., data organized into information that is stored digitally as web-accessible content) may be found in a much broader variety of forms than would be appropriate for software, e.g., content can include reports, publications, backgrounders, FAQ's, images, animations, or other forms of representation. In addition, one of most useful forms of content is a list of appropriate applications for modeling a given class of problem, so that such content includes references to the archive of Task 2.3.1 as a special case.

Thus the templates used for Task 2.3.2's prototype implementations are less standardized than those found in Task 2.3.1, and the path forward's call for new content resources must also reflect the greater variety of digital content.

There are several natural (and non-exclusive) classifications of content use:

- Education, including both university and K-12 components
- Outreach, including dissemination of results to the general public
- Training, including continuing education and certificate programs for practitioners
- Validation, where software applications can be tested against experimental or computational content for validation or verification purposes, respectively
- Publication, including community-oriented technical reports and papers

In addition, unlike the situation in Task 2.3.1 (where most software developers would prefer to maintain their own local repositories of source and object files), many content developers would prefer storing and disseminating their content from a remote location, e.g., the centralized data repository of the NEES project.

Prototype Community Needs

Sample content from the prototype repository is presented in Figures 6 and 7 below. Figure 6 demonstrates a method requested by practitioners at the California Department of Transportation, namely the topic of how a commercial mechanical engineering software tool (in this case, ANSYS) could be used to model a complex laboratory experiment involving structural testing of highway bridges. Most bridge analyses are performed using simplified computational models (e.g., space frame idealizations), so that the goal of capturing local mechanical effects is often intractable. Thus, this sample content on the use of ANSYS was developed, including sample input files and representative validation measures to insure solution quality. Figure 7 is taken from a larger-scale cross-cutting study of the application of systems identification technology to problems of interest in earthquake engineering – the resulting technical whitepaper that catalogs resources (including human resources) appropriate to effective use of systems identification techniques represents a capstone content-development example.

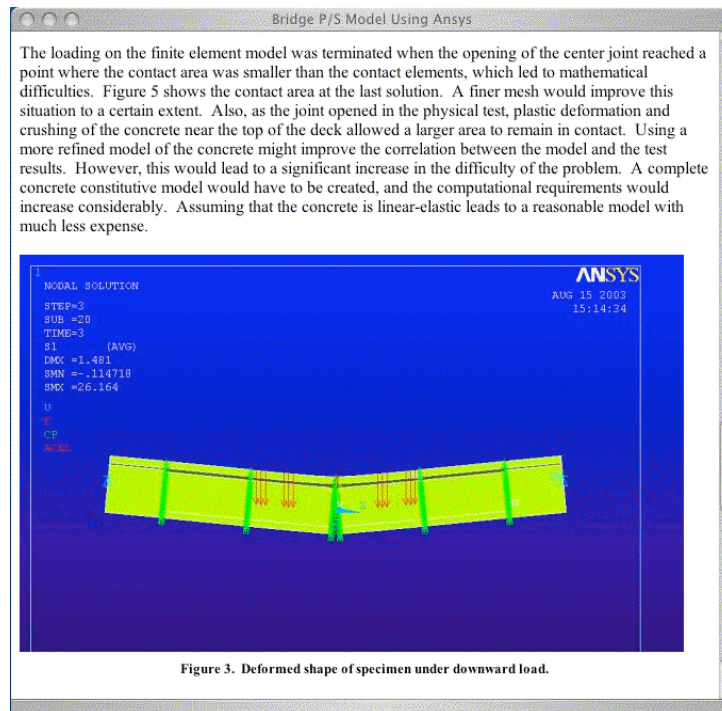


Figure 6: Example Content Page (Bridge FEA Model)

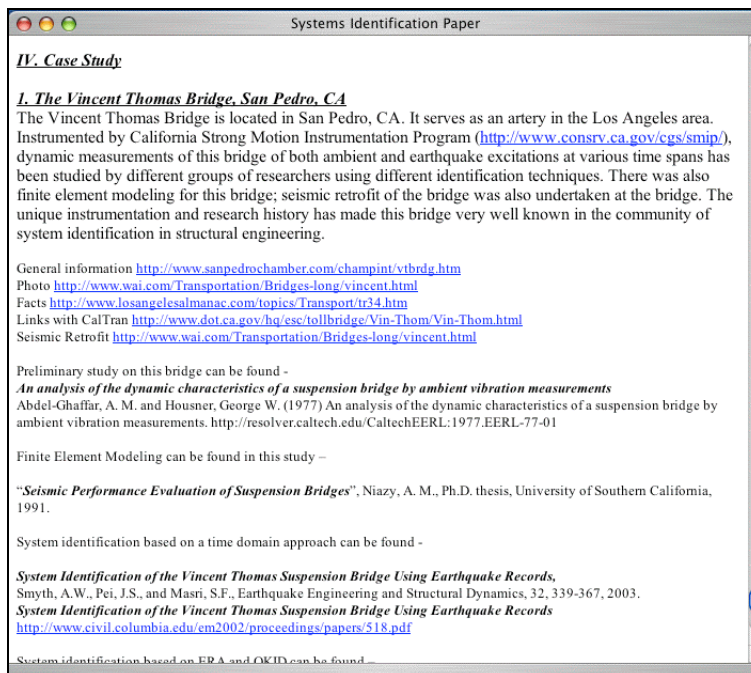


Figure 7: Example Content Page (Systems Identification Whitepaper)

Task 2.3.2 Path Forward

As in Task 2.3.1, the most important future task associated with the content archive is dissemination of the existence of this archive, so that content (or links to content) can be collected and disseminated in the interest of the earthquake engineering community. Because the form of such content is largely open-ended, creation of effective templates for submission of content resources will be a much more complex problem, and hence standardization efforts may not be entirely warranted, beyond an appropriate metadata description for digital content.

Because of the fundamental role of Task 2.3.3 within the NEESgrid simulation effort, it is essential to gather content demonstrating the use of OpenSees within earthquake engineering, so that development of content (for training, verification, and use within experiments) arising from OpenSees analysis should be a high priority for Task 2.3.2 within the last year of the NEESgrid project.

Report on Task 2.3.3

Task 2.3.3 provides for a prototype computational portal that is extensible in overall function and that can incorporate community software tools and NEES datasets. This task is thus necessarily open-ended in scope, and because of its reliance on new information technology, inherently risky in implementation. The underlying technological risk warranted examination of more than one candidate implementation, and thus coordination with efforts outside of the NEESgrid project was chosen to insure an improved likelihood of success, as well as a potential to leverage funding outside the NEESgrid project.

General need for portal access to codes

The primary goal of the simulation portal is that of permitting a variety of software tools to be run from the desktop of the earthquake engineer, using local or remote computational resources, and local or remote earthquake engineering data. Ideally, the location of the resources and the data should be transparent to the engineer using them, but in the near-term, this is not a cost-effective goal (though it certainly is an attractive long-term objective for the NEES computational portal).

In the prototype model for the portal, the distinction between remote and local execution of engineering applications is effected by the following classification:

- (1) applications to be run locally can be found using the web content that defines the prototype code archive, e.g., links to commercial and non-proprietary tools
- (2) applications that are run remotely are accessed through a remote login/job monitoring application that utilizes an Application Service Provider (ASP) model to supply computational resources from a distance.

A good example of the first class of applications would be a MATLAB script that is downloaded from a colleague's website to be run locally on the earthquake engineer's desktop computer. An example of the second class of applications would provide for access from that same engineers' desktop to a remote supercomputer or dedicated cluster,

with appropriately easy-to-use tools used to submit jobs to the remote computer, or to view locally the results of simulations that were performed remotely.

The schematic for the associated model of distributed collaborative computing is shown in the figure below, where the images emphasize the fact that the users of the simulation applications, the data that drives those simulations, and the computational resources that perform the numerical computing are not necessarily located in any particular geographical proximity to each other.

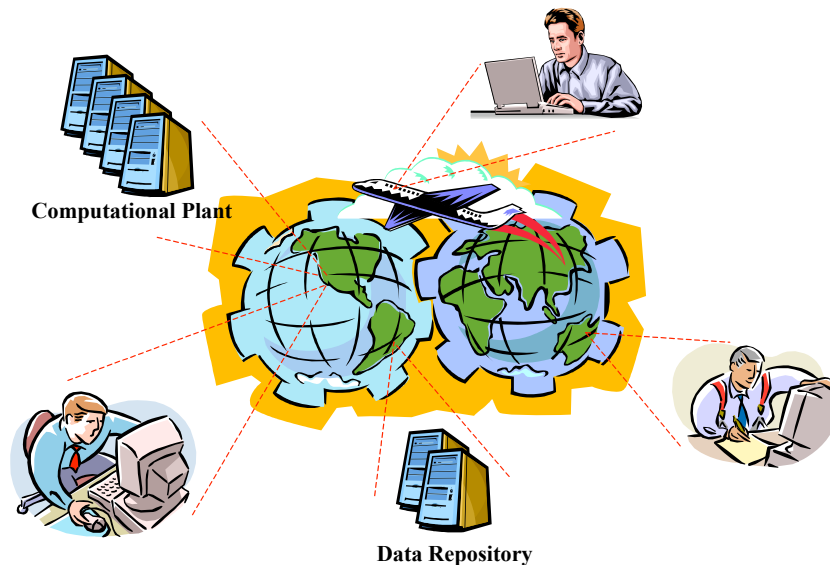


Figure 8: Remote Computing Model for NEES Computational Portal

Prototype Portal Models for Evaluation

Two computational portal models were evaluated under the OU subaward of the NEESgrid Simulation Component:

- (1) the SPUR (Seismic Performance of Urban Regions) model, under development by a team consisting of researchers at Mississippi State University, Carnegie-Mellon University, and the University of California, Berkeley, and
- (2) the Terascale application service provider (ASP) model, under development at Terascale, LLC in New Mexico.

Each of these projects has substantial capabilities for deployment as a candidate NEES computational portal, and each was considered carefully for deployment as the prototype NEES computational subsystem. Each of these prototype portal efforts is based on the combination of a remote job submission/monitoring environment that provides distance-computing capabilities, and a software framework that permits modeling the complex physics found in earthquake engineering problems.

A software framework represents a collection of common software components for building different computer codes. The basic premise behind the use of a framework is

the recognition of a common set of tasks that must be accomplished in writing any computer application code. These tasks can be factored out of the application codes and collected into a single set of components. The goal is to “write once, use often” these common components of the framework. This unique aspect of a framework allows multiple applications developed within a same framework to "talk" to each other in a seamless manner, which greatly facilitates handling the coupling that naturally occurs in many earthquake engineering settings (e.g., soil-structure interaction problems, wave-shore or wave-structure interactions from tsunami research, etc.).

The framework model also makes it easy to add additional physics to a computer code. Finally, use of a computational framework also makes it straightforward to develop codes to take advantage of modern computational advances such as parallel and distributed computing. These modern computational techniques are absolutely essential when analyzing large 3-D problems commonly found in earthquake engineering

The SPUR Portal Effort

The SPUR portal project was funded by the National Science Foundation separately from the NEESgrid award, and proof-of-concept of this approach to constructing a community simulation portal was demonstrated in September 2003 at the Corvallis, Oregon NEES Awardees meeting. The SPUR project utilizes the OpenSees computational framework from the Pacific Earthquake Engineering Research (PEER) Center for its structural modeling, and the combination of OpenSees and SPUR’s portal effort will constitute future simulation efforts within the NEESgrid Simulation Component (see Appendix A for a suggested path forward for this phase of NEESgrid).

In addition to those structural capabilities demonstrated within the SPUR project, OpenSees has substantial capacity to perform geotechnical and soil-structure-interaction analyses, so it makes an excellent candidate for a community code framework for research and practice in earthquake engineering.

The SPUR project’s systems integration technology is in a prototype form, so it will take time and resources to cast this technology into a robust formulation that integrates with the existing data and remote-computing models used in NEESgrid. However, the SPUR team has already made great progress in implementing their prototype portal model, and the personnel working on both the low-level software infrastructure and the higher-level earthquake engineering modeling tools are of uniformly excellent caliber, so there is a very high likelihood of success that this technology can be successfully integrated and deployed in support of the NEES project.

The Terascale Portal Effort

The Terascale portal was funded primarily by Lawrence Livermore and Sandia National Laboratories, in large part to provide for advanced finite-element mechanics capabilities for use in national laboratory and academic collaborations. The Terascale portal effort uses commercial off-the-shelf (COTS) technology to handle remote computing capabilities, and is thus ready for use in the near-term. In fact, the Terascale Applications Service Provider (ASP) model is so simple to use that it is currently being utilized at the University of Oklahoma to introduce undergraduate mechanics-of-materials students to

finite-element techniques capable of helping illustrate such advanced mechanics principles as stress concentrations, and the limitations of St. Venant's principle.

One attractive aspect of the Terascale model that led to its extension and use in the NEESgrid computational simulation effort was that its ASP model was designed to incorporate third-party software applications. Prototype efforts have demonstrated that the OpenSees framework can feasibly be deployed within the Terascale ASP, so that an alternative near-term implementation of a portal for use with OpenSees is readily available should it become necessary to deploy one.

Sample earthquake-engineering analyses using both the Terascale framework and the Terascale ASP software are shown in the following figures, which illustrate the application of various Terascale computational tools, including:

- (1) Configurator, an interactive application that permits the engineer to design and populate complex data and metadata formats relevant for a particular field of research or practice. One of the key goals of this application is to shield the engineer from the idiosyncrasies of low-level data and metadata representations such as HDF or XML.
- (2) Grok, which permits graphical user interaction with the resulting data and metadata formats, including creation of high-level metadata needed to define the physical model (e.g., location of sensors, conceptual data for mesh generation, etc.), and some simple visualization of results.
- (3) Mesher, which creates 2D and 3D meshes from the defining metadata of the problem, e.g., materials, high-level geometry, etc (see Figure 9 below).
- (4) The Terascale Applications Framework, which performs the computational physics using scalable technology, so that the same analysis software and data can run on a single-processor desktop computer or on a remote supercomputer, and
- (5) Focus, which performs advanced client-server visualization so that an engineer can work effectively in settings remote from where the computational data has been stored, using a thin-client applications model. Focus implements a wide variety of visualization capabilities required for engineering problems based on finite-element analyses, including color fringe plots and computer-generated animations of transient analyses (see Figures 10, 11, and 12 below).

Figure 9: Mesh Metadata for Geotechnical Model

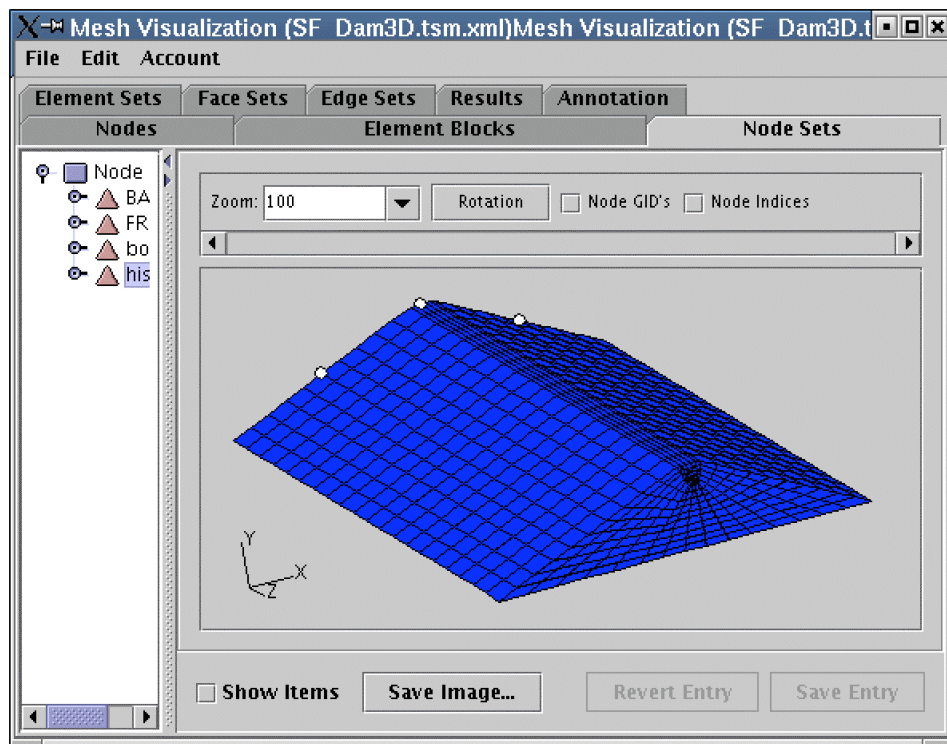


Figure 10: 3D Mesh Derived from Geotechnical Model Metadata

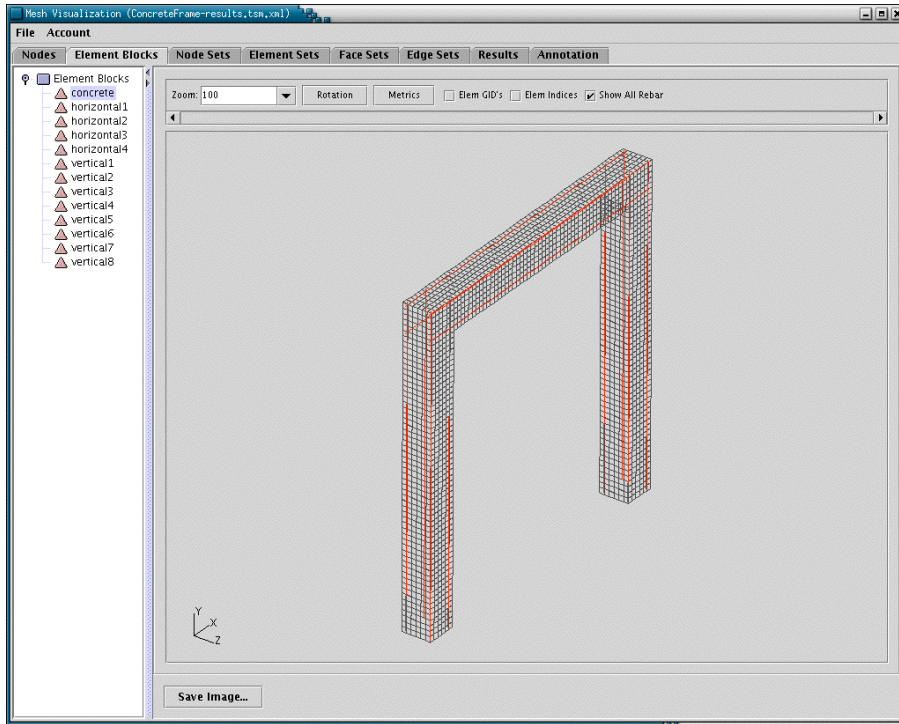


Figure 11: 3D Reinforced Concrete Mesh from Structural Model Metadata

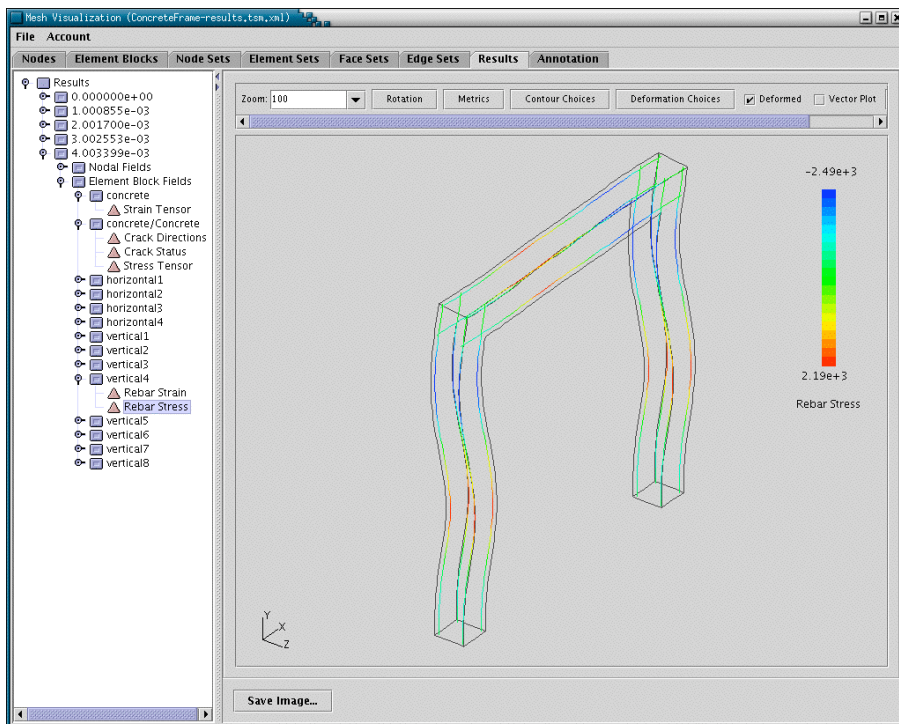


Figure 12: 3D Reinforced Concrete Results from Structural Model

Under the first year of NEESgrid's simulation effort, the Terascale framework's mechanics orientation was augmented towards earthquake-engineering capabilities, including:

- Addition of public-domain high-performance scalable iterative solver technology to permit finer-resolution analyses of long-duration problems commonly found in earthquake engineering.
- Development of discrete steel reinforcement modeling tools to facilitate high-fidelity simulation of the dynamics of reinforced concrete structures.

The former extension permitted a broad range of analysis and research capabilities in support of high-resolution nonlinear problems such as those found in earthquake engineering settings. The latter extension facilitates a wide variety of structural models involving the use of concrete, prestressed concrete, and masonry systems.

The authors intend to deploy the Terascale framework for free early-adopter community use in Q1-Q2/2004, using a 32-processor Itanium Linux Cluster that was purchased under an MRI grant from NSF. This deployment effort will provide for access to computational resources (including appropriate education and training efforts) so that earthquake engineers can begin using framework technology in research and practice. It will also provide a low-risk environment for engineers wishing to deploy their scalable code on very large supercomputers that utilize Itanium technology, e.g., the NSF-funded TeraGrid effort at UIUC/NCSA, Caltech, ANL, and UCSD/SDSC.

Task 2.3.3 Path Forward

Task 2.3.3 is the most ambitious and technically challenging of the current simulation tasks, and its future development constitutes most of the remaining new work on the NEESgrid simulation component. There is a need for available visualization tools for use with OpenSees, and the authors are evaluating various open-source visualization tools (such as the VisIt scalable visualization application from LLNL, which is an open-source visualization project that includes considerable capabilities for rendering and viewing complex finite-element datasets). Resources that insure that visualization tools are integrated with the other technology arising from the SPUR project would be well-spent.

The extension of the mechanics-oriented Terascale framework for earthquake engineering applications will continue, and considerable progress is currently being made on this front, including the recent addition of soil-structure interaction capabilities that permit use of one-dimensional and two-dimensional elements (e.g., beams, columns, plates) within continuum finite-element approximations.

Conclusions and Future Work

The first tasks of NEESgrid simulation effort are complete in prototypical form, and thus the more mundane aspects of this exciting research and development venue are finished. The most important step for Tasks 2.3.1 and 2.3.2 is the promotion of templates for software and content resources, followed by the collection of available community resources implemented within those template representations.

Future efforts within NEESgrid will be based on the PEER OpenSees framework, with integration technology developed within the SPUR project. This solution should provide excellent open-source capabilities for use within the earthquake engineering community, and an outline of this proposed effort as developed by the authors is found in Appendix A. This content was circulated to the SI team in August 2003, and provides a feasible path to proceed from the initial un-ambitious simulation scope towards a technical solution more responsive to the needs of the earthquake engineering community.

There are still gaps in the current simulation component, including the all-important arena of interactive visualization, and in cataloguing many of the applications domains relevant to tsunami modeling. Appropriate resources should be deployed to insure that these gaps are filled in order that all aspects of engineering computing (including the all-important issue of usability and user-interface design) are considered in the completion of the NEESgrid project.

The authors wish to thank Joy Pauschke of the National Science Foundation for her continuing efforts in funding and integrating the SPUR effort so that it might better serve the NEES project, Bill Spencer of UIUC's Civil Engineering department for taking on the difficult but all-important role of serving as NEESgrid Principal Investigator over the last 15 months of its construction phase, and the members of the SPUR team (including Tomas Haupt, Greg Fenves, Jacobo Bielak, and Roger King) for their excellent technical work in demonstrating appropriate computational technology, and for showing the community how computational simulation can substantially advance the state-of-the-art in earthquake engineering research and practice.

Appendix A: A Plan for an OpenSees Computational Simulation Component

This appendix describes the outline of an extensible NEES computational simulation portal that can be constructed using existing NEESgrid and community resources. This portal will be responsive to those community requirements identified by the SI and CD projects, will integrate seamlessly with SI products where appropriate, will be sufficiently easy to deploy so that the success of this effort will be readily replicable within the community of earthquake engineering researchers and practitioners, and will permit future expansion of existing computational capabilities at relatively low cost. The tasks required to deploy this computational portal map well to the existing tasks on the computational simulation component of NEESgrid, so this system can be designed and deployed with little or no additional resources.

Components of the Computational Subsystem

There are four components of the proposed computational subsystem, and each is required for effective deployment of a system responsive to community needs:

- (1) software resources to provide convenient access to community models and tools,
- (2) data resources to provide input data, model parameters, and results for verification
- (3) training resources to demonstrate how software and data resources can be used,
and
- (4) computational resources (e.g., Linux clusters) to perform analyses

Each of these components is briefly described below, including subsequent discussion as to how these components integrate with the NEESgrid effort at UIUC/NCSA.

Software Resources

Access to community software resources will be provided via a portal interface, so that earthquake engineers can find, download, and use these applications from their desktops, regardless of whether the resulting analysis is a simple Matlab script that runs locally on their desktop computer, or a high-fidelity finite-element application that requires a remote supercomputer. Access to these software resources will be simplified by the use of appropriate metadata to characterize the various computational tools by function, supported platform, software quality, and interoperability with NEES standards (e.g., NEESML).

Software resources are the computational tools that perform earthquake engineering analyses. Most of these tools are based on finite-element analysis (FEA) techniques, and while many are community-developed and freely available for use (e.g., OpenSees from the PEER center at Berkeley), other community software tools are proprietary and thus require appropriate licensing fees for use (e.g., ABAQUS, from HKS, Inc.). The proposed computational portal will utilize open-source or other readily-available software to the extent possible, to simplify intellectual-property licensing agreements, and to minimize costs

Data Resources

There are three forms of data inherent in computational simulation:

- input data and parameters, e.g., geometry of the model, material parameters, etc.
- generated output, which is often stored in a unique tool-dependent binary format, and
- extracted data, which is harvested from the generated output for subsequent use, e.g., for post-processing or for comparison with experimental results at discrete points.

The most important data resources are those that define the model, whether that model is from an experiment or a purely virtual computational simulation. For example, geometric information, material parameters, earthquake records, initial and boundary conditions, and other related data are used to form the input required for a computational analysis. These data resources must be well-integrated within the NEESgrid data repository, either as experimental data that is used to drive a computational simulation, or as extracted data that is used in conjunction with an experimental plan (e.g., to analyze a proposed experiment in order to optimize sensor locations, etc.).

The generated results of a computational simulation may not necessarily be desired within the NEESgrid data repository, for various reasons (e.g., the data is too bulky, or it's so easy to regenerate that it is thus wasteful to store), but the other two classes of data and metadata should be captured reliably, because these data resources define the computational simulation and the subsequent storage of simulation results for later retrieval and use.

Training Resources

Computational simulation applications such as OpenSees are powerful tools for analysis, and with that power comes the possibility of misuse due to analyst inexperience or other potential engineering mistakes. While an ill-posed experimental design may readily demonstrate its shortcomings (e.g., by premature experimental failure) so that the resulting data gathered is appropriately suspect, an ill-posed computational simulation can be much more difficult to detect before it is utilized, and hence incorrect computational data arising from miscues in analysis are a potentially serious problem for the earthquake engineering community.

Appropriate training materials will be included with the software and data resources of this computational subsystem, both on general principles of earthquake engineering analysis (e.g., basic introductions to FEA methods) and on specific topics such as the particular software used (e.g., OpenSees has a different learning curve than does ABAQUS or ANSYS). These training resources will be relevant to earthquake engineering practice, and will include sample input files to help analysts gain experience with new tools or new analysis capabilities, and sample output files required to insure that the resulting analyses are running properly to completion.

Computational Resources

Software and data provide the capabilities to perform analyses, but there must be appropriate hardware resources available to complete the picture of a successful

computational analysis. Rapid improvements in performance have made desktop computers capable of solving a large fraction of existing engineering analyses, so the proposed computational subsystem must readily utilize the power available on a typical engineering desktop workstation. In addition to desktop resources, the portal interface will permit seamless integration of local and remote clusters that provide a migration path towards higher-resolution and higher-fidelity analyses that are becoming more common as more advanced computational techniques (e.g., implicit finite-element analysis of transient nonlinear response) become widely available within the earthquake engineering community. Another important advantage of a web-based portal interface for computational simulation is that the full diversity of software tools found in engineering research and practice can be accessed from the desktop workstation of the engineer. Thus open-source tools can be used along with proprietary tools to permit verification of difficult analyses, or to facilitate code-based methods (e.g., ATC-55) that rely on computational simulation techniques.

Integration Requirements

It is essential that the computational portal integrate seamlessly with the products being developed within the NEES MRE, and especially with those software and data capabilities being developed under the aegis of the NEESgrid Systems Integration award. The content below provides an overview of how such integration can readily be accomplished.

Integration with Existing SI Products

As mentioned above, the key requirement for integration with the SI project is the capability to harvest data from the NEESgrid data repository for use in performing computational simulations, or to ingest computational data results into the repository for subsequent use by other members of the community. Another important requirement arises from the need to perform coupled numerical/physical experiments, where a computational simulation analysis is treated as another experimental component in a multi-site experimental setting.

These integration requirements can be handled by decomposing the computational simulation data into appropriate metadata representations and more bulky mesh-dependent data. The former can be stored or retrieved from the NEESgrid data repository, while the latter can be stored locally, or within the repository, or deleted if future simulation data regeneration is straightforward enough. The metadata for computational simulation must include appropriate capabilities for hybrid experiments, so that computational simulation metadata may end being developed up as a superset of some aspects of experimental metadata.

Integration with SI Experiment-Based Deployment Methods

One obvious integration strategy for the proposed computational portal involves performing supporting computational analyses of the experiments that will be developed as part of the SI experiment-based deployment process, for example, developing computational analyses of the MOST experiment using OpenSees or a commercial tool

such as ANSYS. This strategy insures that the successful deployment of the NEESgrid operational subsystem is accompanied by concomitant deployment of the computational portal, so that all components of NEESgrid are fully-functional by Oct 1, 2004. This integration strategy could readily be utilized in all the proposed Experiment-Based Deployment examples under consideration for the NEESgrid project, and such an ambitious integration effort would provide validation and verification opportunities for a wide variety of community codes, in much the same way that the NSF VELACS project revolutionized the state of the art of geotechnical engineering simulation tools for earthquake engineering practice.

Integration with Existing SI Computational Simulation Tasks

The proposed computational subsystem effort maps well to existing SI computational simulation tasks, as shown in the table below. This provides for cost-effective integration activities that require little or no retooling of existing SI tasks.

Proposed Computational Portal Component	Associated NEESgrid Tasks	Required Additional Resources	Available Community Supporting Efforts
Software Resources	Task 2.3.1 Task 2.3.3 Task 1.1.5	None: already in SI scope	NSF Earthquake Engineering Centers, NSF SPUR Project
Data Resources	Task 2.4 ??	Uncertain until scope better-defined	Community data models (e.g. Oregon State, with extensions), COTS models (e.g., STEP)
Training Resources	Task 2.3.2	None: can be done within SI scope	Earthquake-engineering-relevant documentation for supported tools (e.g., OpenSees)
Computational Resources	None	None	OU/OSCER resources, NSF HPC Centers (NCSA, SDSC, PSC)

Other Considerations

If this vision is an appropriate one, it needs to be fleshed out with specific milestones and deliverables based on feasible budgets, and these deliverables can readily be developed within the setting of a campaign goal, e.g., “model the MOST experiment to high accuracy by Jan 1, 2004”, or “perform and validate a full soil-foundation-structure-interface simulation using OpenSees by March 31, 2004”.