

(Draft Whitepaper Version: 1.0
Last modified December 17, 2003)

The MOST Experiment, July 30, 2003

Feedback on this document should be directed to bfs@uiuc.edu

Acknowledgment: This work was supported by the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Program of the National Science Foundation under Award Numbers CMS-0117853 (NEESgrid), CMS-0217325 (NEES MUST-SIM at UIUC), and CMS-0086592 (NEES FHT at CU-BOULDER).

Table of Contents

1	Introduction	6
1.1	A brief overview of the MOST experiment	6
1.1.1	Glossary of Acronyms Used In This Report.....	6
1.2	Rationale	7
1.2.1	Necessity for Multi-Site Integrated Physical-Analytical Simulations	7
1.2.2	Selection of Structure to be Simulated	11
2	NEESgrid components	12
2.1	Core Grid Services	13
2.2	Data and Metadata Management	14
2.2.1	Data service components	14
2.2.2	NEESML.....	16
2.2.3	MOST Metadata	17
2.3	NEESgrid Telepresence Mode (TPM)	18
2.4	Data Acquisition and Streaming Data Services	19
2.5	Computation and Simulation Services	19
2.6	NEESgrid User Interface	21
3	Problem Description	21
3.1	The Concept	21
3.2	Experiment Setup	23
3.2.1	University of Illinois at Urbana-Champaign.....	23
3.2.2	University of Colorado at Boulder	29
3.2.3	National Center for Supercomputer Applications.....	31
3.3	Substructure PSD Testing Software	32
3.3.1	Design Philosophy.....	32
3.3.2	Simulation Coordinator	32
3.3.3	Master Computational Module.....	33
3.3.4	Slave Computational Module.....	33
3.3.5	Experimental Module	33
4	Execution of Distributed Simulation	33
4.1	Verification and Testing	33

4.2	Accessing Results	34
4.2.1	Using NEESgrid CHEF Tools	34
5	Community participation	43
5.1	Public relations before, during, after experiment	43
5.1.1	Postcards	43
5.1.2	ES-VTC presentations.....	43
5.1.3	Press	43
5.2	Website and multimedia	44
5.2.1	Website Usage During the MOST Event	44
5.3	CHEF usage	48
6	Feedback.....	50
6.1	Whole community	50
6.2	MOST team members	51
7	Future directions	54
8	Contributors to the MOST Experiment.....	55

Table of Figures

Figure 1: Experimental masonry structure, Structures Laboratory at Georgia Tech.....	8
Figure 2: Schematic of 1960s non-seismically-designed building.....	9
Figure 3: Finite idealization of building in Figure 2.....	9
Figure 4: Structural damage from the August, 1999 Turkey earthquake.....	10
Figure 5: The role of the multi-site simulation in seismic performance evaluation.	11
Figure 6: MOST Two-bay Single Story Steel Frame Structure.	12
Figure 7: The NEESgrid Big Brother monitoring service (http://neespop.ncsa.uiuc.edu/bb/).....	14
Figure 8: NEES repository graphical interface	15
Figure 9: Data flow paths during the MOST experiment.	16
Figure 10: UIUC MOST metadata conceptual model.....	18
Figure 11: DAQ and NSDS.	19
Figure 12: SAC frame tested in the MOST experiment.....	22
Figure 13: Substructure for MOST experiment	23
Figure 14: Column tested at UIUC	23
Figure 15: UIUC experiment layout.....	24
Figure 16: UIUC experimental set-up.....	24
Figure 17: Photoelastic Stress Measurement System	25
Figure 18: Instrumentation of the UIUC column.....	26
Figure 19: Krypton Measurement System.....	26
Figure 20: Measurement of control point displacement and rotation.....	27
Figure 21: Servo control diagram for the control point movement	28
Figure 22: Schematic of the CU experimental setup.....	29
Figure 23: Overview of the CU experimental setup.....	30
Figure 24: Flow chart of CU control system	31
Figure 25: Modules of PSD software	32
Figure 27: Selecting an event to view with the Data Viewer tool.....	35
Figure 28: Add a new view of the experiment to your arrangement.	36

Figure 29: Adding a view and selecting variables.	36
Figure 30: Arrangement of multiple views in the Data Viewer.....	37
Figure 31: Reviewing past chat interactions about the MOST experiment.....	38
Figure 32: Posting a chat message.....	38
Figure 33: Real-time video from Colorado in the CHEF interface.	39
Figure 34: A screenshot from the University of Colorado (UCol) E-notebook displaying experimental data from the MOST experiment.	40
Figure 35: NTCP Sample Tool.....	41
Figure 36: Users Present.....	41
Figure 37: The "floating tool" feature allows users, for example, to view data and streaming video while chatting with other participants.	42
Figure 38: The MOST website, http://www.neesgrid.org/most	44
Figure 39 - Daily usage on the NEESgrid website for July 2003.....	45
Figure 40 - Daily usage on the MOST section of the website for July 2003.....	45
Figure 41 - Usage of MOST section of website by country for July 2003.....	46
Figure 42 - Daily usage on the MOST section of the website for August 2003 ...	47
Figure 43: MOST participation over time	48
Figure 44: Chat messages over time	50

1 Introduction

1.1 A brief overview of the MOST experiment

The Multi-Site Online Simulation Test (MOST) took place on July 30, 2003. The MOST experiment was a large-scale experiment conducted in multiple geographical locations which combined physical experiments with numerical simulation in an interchangeable manner. The partnership between the NEESgrid SI Team, the MUST-SIM team at UIUC, and the FHT team at the University of Colorado at Boulder showcased new (and previously developed) NEESgrid capabilities and demonstrated the first integration of a prototype implementation of the full NEESgrid system with application software developed by earthquake engineers (UIUC, Colorado and USC) to support domain-specific earthquake engineering experiments.

This document describes in detail the NEESgrid components, the experimental setup, the process of preparing and conducting the experiment, media and public relations activities, and feedback from the community and MOST team members about the experiment. Overall, the experiment has been considered a success, although improvements will be made in future experiment-based deployments of NEESgrid 2.0, primarily in the areas of planning, fault-tolerance, and the CHEF interface.

1.1.1 Key to Acronyms Used In This Report

This section provides a key to acronyms that appear throughout the report and in many cases indicates where specific components and concepts are explained.

Acronym	Full Name	Refer to section
ANL	Argonne National Laboratory	
API	Application Programming Interface	2.2.1
CEPT	Cable-Extension Position Transducer	3.2.1
CHEF	CompreHensive collaborativE Framework	4.2.1
DAQ	Data Acquisition	2.4
EERI	Earthquake Engineering Research Institute	5.1.1
ES	Equipment Site	
FHT	Fast Hybrid Test	1.1
GSI	Grid Security Infrastructure	2.1
GUI	Graphical User Interface	2.3
LED	Light emitting diodes	3.2.1
LVDT	Linear Variable Differential Transformers	3.2.1
MAE Center	Mid-America Earthquake Center	1.2.1
MDS	Monitoring and Discovery Service	2.1

MOST	Multi-Site Online Simulation Test	1.1
MUST-SIM	Multi-Axial Full-Scale Sub-Structures Testing and Simulation	1.1
NEES	Network for Earthquake Engineering Simulation	
NEESML	NEES Mark-up Language	2.2.2
NFMS	NEESgrid File Management Service	2.2
NMDS	NEESgrid Metadata Service	2.2
NSDS	NEES Streaming Data System	2.4
NTCP	NEES Telecontrol Protocol	2.5
PID	Proportional, Integral, Derivative	3.2.1
PSD	Pseudo-Dynamic	1.2.1
PZT	Pan-Zoom-Tilt	2.3
SCRAMnet	Shared Common Ram Network	3.2.2
SCXI	Signal Conditioning eXtensions for Instrumentation	3.2.1
SI	System Integrator	
TPM	Telepresence Mode	2.3
UIUC	University of Illinois at Urbana-Champaign	
USC/ISI	University of Southern California Information Science Institute	

1.2 Rationale

1.2.1 Necessity for Multi-Site Integrated Physical-Analytical Simulations

Earthquakes affect vulnerable communities in a broad-ranging manner. It is therefore imperative that all available tools should be deployed to mitigate earthquake effects. The tools of earthquake performance assessment are (i) testing, (ii) analysis, and (iii) collecting data from regions hit by destructive earthquakes. Each of the three tools has advantages and drawbacks. Their relative merits and application potential are briefly assessed below.



Figure 1: Experimental masonry structure, Structures Laboratory at Georgia Tech

Experimental testing is an essential and powerful investigation tool in earthquake engineering applications. However, it is not without shortcomings. The three approaches of *monotonic*, *cyclic*, and *dynamic* loading offer some complementary and other conflicting data. It is easier to test structures monotonically, but the relevance to earthquake applications is rather remote since no degradation of stiffness or strength is taken into account. This leads naturally to cyclic (constant or variable amplitude) loading as the next step up the complexity scale. Once a decision to test cyclically has been taken, a number of critical decisions are required, each of which may influence the obtained results in a fundamental manner and may lead to totally different observations hence conclusion. The applicability of cyclic loading results to earthquake engineering is therefore questionable.

The question of where (quantitatively) earthquake response results are, in comparison with monotonic and cyclic investigations, should at least in part be answered by dynamic (shaking table or on-line computer-controlled) testing. Shaking table testing is close to reality, but scale effects, full dynamic similitude, and difficulties with soil-structure interaction modeling pose serious hurdles. The development of on-line, computer-controlled (pseudo-dynamic or PSD) testing has by-and-large liberated dynamic testing from the constraints of scale effects, but strain-rate insensitivity is a pre-requisite for the acceptance of the obtained results. Figure 1 shows a MAE Center full-scale masonry structure prior to testing at the Structures Laboratory at the Georgia Institute of Technology. Whereas the structure is quite realistic, it is founded on a concrete slab on top of the laboratory strong-floor. Therefore its periods of vibration often bear little resemblance to its *in situ* counterpart.

Analytical investigations are fast-developing. However, similar to the situation with laboratory testing, decisions are needed with regard to the type of model, method and

input. The model may vary from a single-degree-of-freedom idealization of the whole structural system to a detailed 2D or 3D representation of the structure, foundation and surrounding soil. The method may be static or dynamic, with various sub-classes of either of the two options. Great advancements have taken place in recent years in the aforementioned ingredients of seismic analysis. Figure 2 depicts a 1960s non-seismically designed RC building alongside its finite element idealization (

Figure 3) for the purposes of inelastic dynamic response history analysis. Even the center lines of the frame and slab are difficult to place in their natural positions due to inherent shortcomings in flexural finite elements for frames.

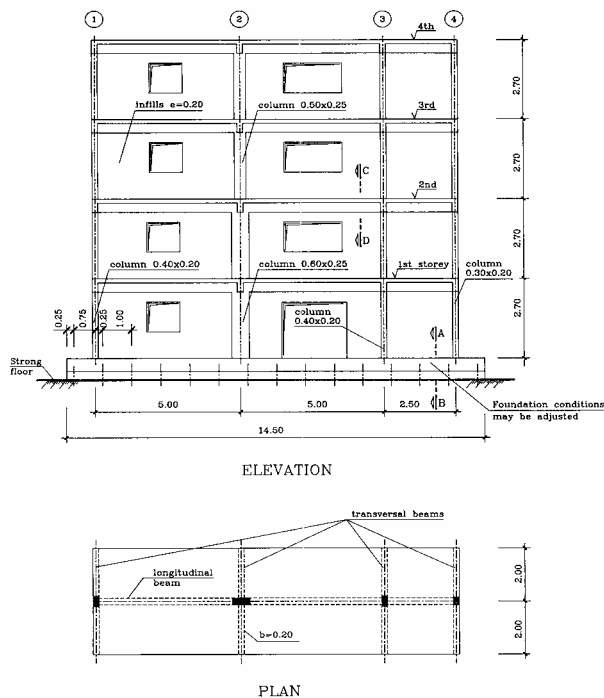


Figure 2: Schematic of 1960s non-seismically-designed building

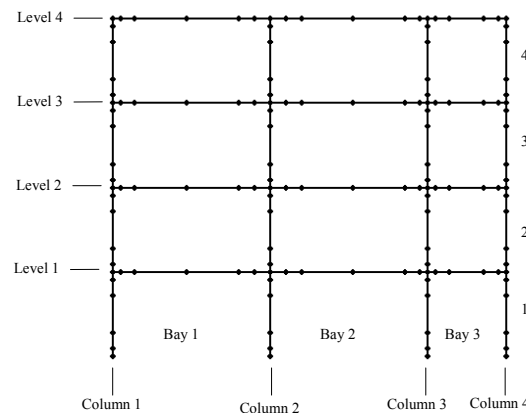


Figure 3: Finite idealization of building in Figure 2

Indeed, developments in analytical capabilities have outpaced the advances in testing, due to the faster turnover in analysis and the relatively low costs involved, underpinned

by the leaps in computer technology. Notwithstanding, there are modeling problems that continue to be unbreached, such as shear deformation and failure of RC structures, detailed three-phase (matrix, voids and moisture) soil response under variable amplitude dynamic loading, or even bolt-hole ovalization in steel connections.



Figure 4: Structural damage from the August, 1999 Turkey earthquake.

Field observations are ultimately the most realistic since all aspects of source, path, site, foundation, structural, and non-structural components are taken into account. However, this result is for a specific set of the aforementioned parameters, all of which influence the damage inflicted on the exposed building stock. Even if the seismological quantities are those sought for a particular application, the soil substrata, structural material, resistance system, age, configuration and foundation type are unlikely to be those under investigation. Consequently, the data collected is generically regional in nature and cannot be readily applied in a predictive fashion. Finally, the frequency of earthquake occurrence is such that its applicability and usefulness is serendipitous. In Figure 4, a modern RC structure devastated by the Kocaeli (Turkey) earthquake of August 1999 is shown. The design practice in the Gulf of Izmit area is such that the results are not applicable to other regions, thus rendering field-collected data of very limited use. In general, field observations can only be used to verify testing and analysis models and to provide rather heterogeneous data sets that may support analytical studies of populations of structure.

Assessment of the relative merits and drawbacks of the three seismic performance evaluation methods leads naturally to the concept of multi-site on-line computer-controlled integrated testing-analysis of complex systems (Figure 5). Such an approach, which is the underlying vision of NEES, would optimally combine the various realistic and attractive features of each assessment approach, thus availing of test results far superior to those obtained from existing approaches. With reaction walls, geotechnical centrifuges, shaking tables, structural and geotechnical field facilities, instrumented sites and a wave tank, NEES furnishes the earthquake engineering community with a comprehensive toolkit for the application of multi-site simulations, employing the NEESgrid cyberinfrastructure.

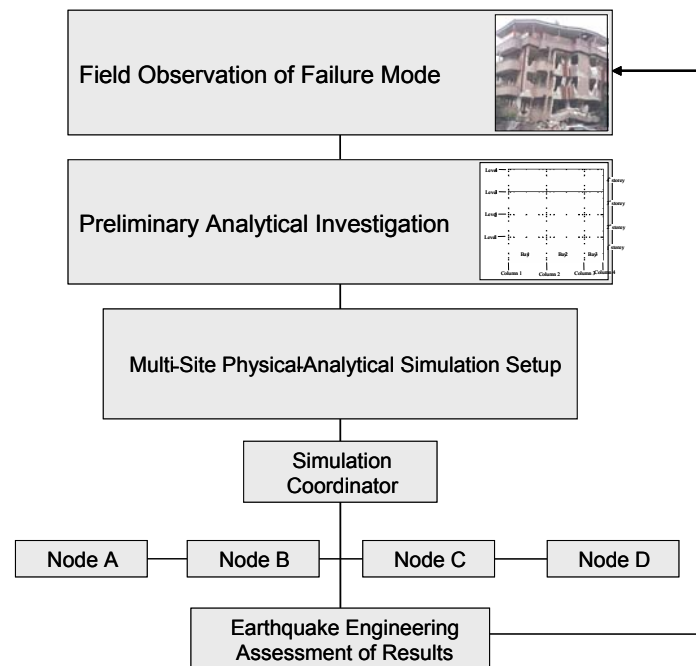


Figure 5: The role of the multi-site simulation in seismic performance evaluation.

In the context established above, the MOST features, as the first proof test of NEESgrid developments on a real experiment, the components which commonly appear in earthquake engineering investigations, namely finite element analysis and structural testing of components.

1.2.2 Selection of Structure to be Simulated

Considering that MOST is the first proof test of the NEESgrid software capabilities utilizing three sites, two experimental and one analytical, the conceptual design of the simulation was driven by the objectives in the following respect:

- The simulation should feature commonly-used components of seismic performance assessment;
- It should be sufficiently complex to provide a challenging development and deployment environment;
- It should have an accurately quantifiable response that can be used for validation; and
- It should be repeatable without change in the measured response.

The selected structure (

Figure 6) is a two-bay single-story steel frame that is responding in the elastic range under the design earthquake ground motion. The frame exhibits asymmetric response about the vertical axis since one beam-column connection (Colorado side) is pinned, while the

other (Illinois side) is a moment connection, hence it is sufficiently complex. The middle column is pinned at the base in order to increase the respective moments to which the exterior columns are subjected, for a given load.

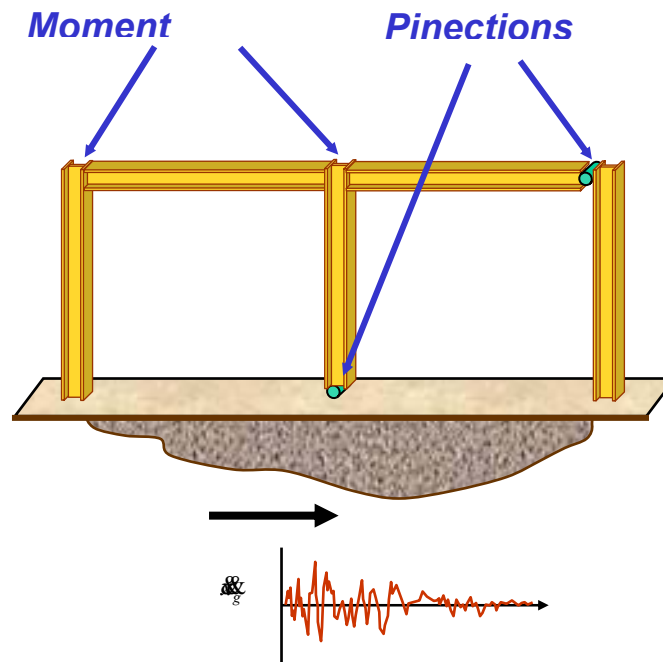


Figure 6: MOST Two-bay Single Story Steel Frame Structure.

Having two bays provides an opportunity to test and analyze flexural response of three columns, since there are three sites involved. A single-story structure would have limited or no geometrically-nonlinear effects that may complicated the validation of the obtained results. Moreover, having a moment connection at the Illinois side provides an opportunity for the MUST-SIM team to develop multi-actuator control, which is necessary for the development of the UIUC NEES site. Finally, use of steel renders the test completely repeatable at the anticipated response amplitude of ± 15 mm at the roof level, because inelastic degradation, low- and high-cycle fatigue are not operative.

2 NEESgrid components

By integrating the strengths of the physical experimental setups at UIUC and CU, described in Section 3.2, with the computational features of the NEESgrid, the MOST experiment demonstrated the full range of NEESgrid's capabilities, which fall into the following broad categories:

- **Core grid services**, which ensured security and managed grid resources during all site-to-site transactions;
- **Data management services**, which enabled the transfer of data and metadata between repositories;
- **Telepresence services**, which were responsible for exchanging control and status information between sites, observing the experiment and capturing data from a remote site, and controlling the experiment from a remote site;
- **Data Acquisition and Streaming Data Services**, which enabled the transfer of sensor data from the local data acquisition system to the remote users and to repositories;
- **Computation and simulation services**, which enabled access to computer resources and simulation software.

What follow are brief descriptions of all components, and services, including an assessment of their performance during the experiment. Detailed descriptions of these systems and their components may be found in NEESgrid System Architecture, Version 1.1 (http://www.neesgrid.org/documents/NEESgrid_SystemArch_v1.1.pdf).

2.1 Core Grid Services

The core grid services component of the MOST experiment included the following features:

- Grid Security Infrastructure (GSI) for enabling secure authentication and communication over an open network. This included the MyProxy online credential repository, which allows a user to store his or her grid credentials and retrieve a proxy credential whenever logging on to the grid.
- Monitoring and Discovery Service (MDS), which enables the discovery of new resources and equipment.
- Big Brother monitoring service (Figure 7), which provides the <http://grid.ncsa.uiuc.edu/myproxy/> automatic monitoring of health and status of NEESgrid machines. (<http://neespop.ncsa.uiuc.edu/bb/>)
- Network Weather Service (NWS), which provides network load information.
- Improved GPT-based packaging services with user-convenient installation schemes.

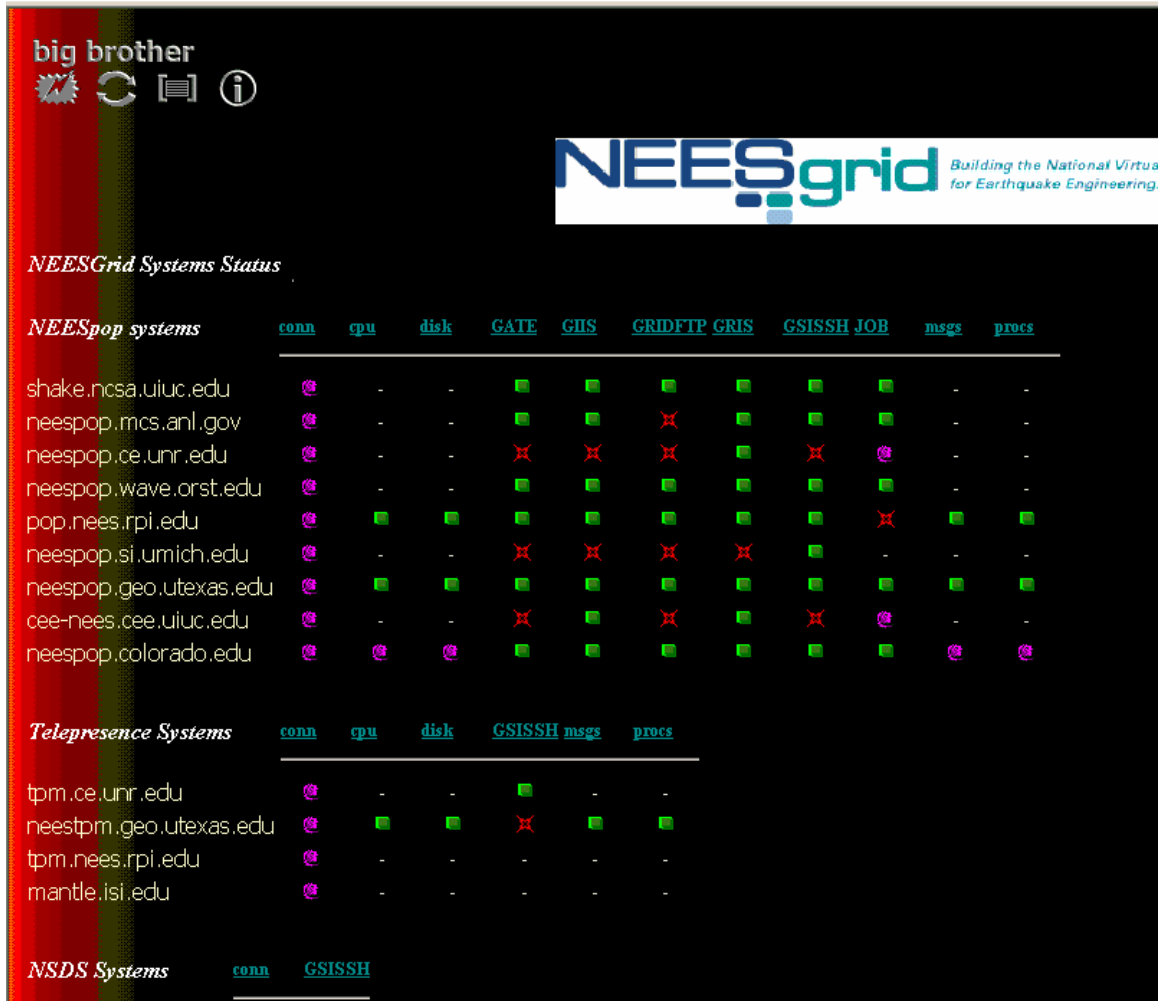


Figure 7: The NEESgrid Big Brother monitoring service (<http://neespop.ncsa.uiuc.edu/bb/>)

2.2 Data and Metadata Management

2.2.1 Data service components

The NEESgrid metadata service (NMDS) implements repository functions associated with metadata, including metadata storage, retrieval, search, security, and provision of an API. This service supports CHEF-based metadata ingestion, browsing, and search end user capabilities, as well as archiving and application access to metadata. It also provides the file management system with a way to store and retrieve metadata about files and file transfers.

The NEESgrid file management service (NFMS) implements repository functions associated with data, including sensor data, images, video, and documentation. These

functions include security, file transfer, naming and directory services, linking files to metadata, and provision of an API. This supports CHEF-based data ingestion, browsing, and data viewer interfaces.

The CHEF browser provides a user interface to the MOST repository. It allows the user to upload, download, browse, and search data and metadata. The users interact with metadata using the graphical interface in CHEF (Figure 8).

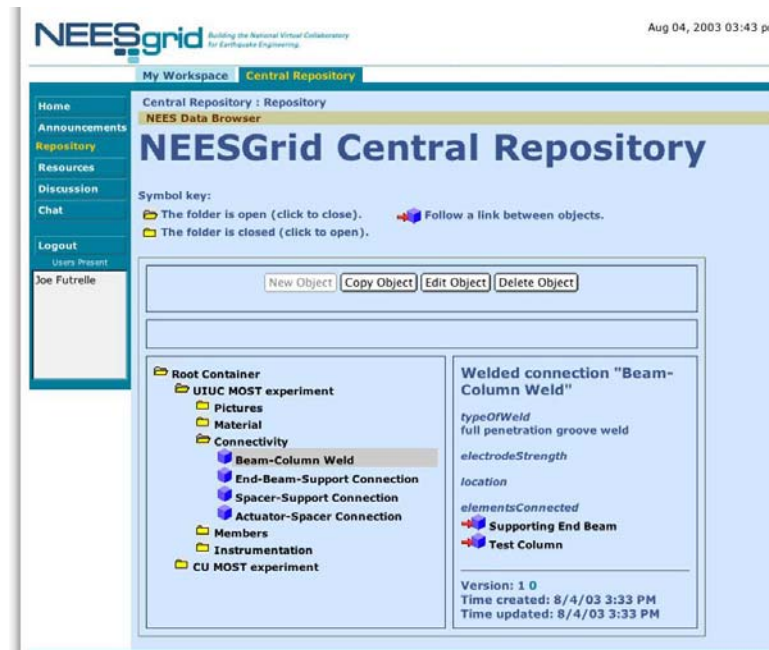


Figure 8: NEES repository graphical interface

The data ingester is a tool that was used in MOST to move data files into site repositories from local machines used for data acquisition or machines from which numerical simulation data originates. It provides an example of an application that uses the NFMS API.

As shown in Figure 9, simulation and sensor data was gathered to the UIUC/Newmark repository using a distributed set of data ingestion components, one for each data source. In the case of the LabVIEW Data Acquisition Systems (DAQ) at UIUC and CU-Boulder, the data was written by the DAQ code to staging directories on the local disks of the site NEESpops. A data ingestion component on each NEESpop periodically scanned the staging directory for new files. In the case of the MATLAB simulation model, the data was written by MATLAB to a staging directory on the NEESMOST machine, and a data ingestion component scanned that directory.

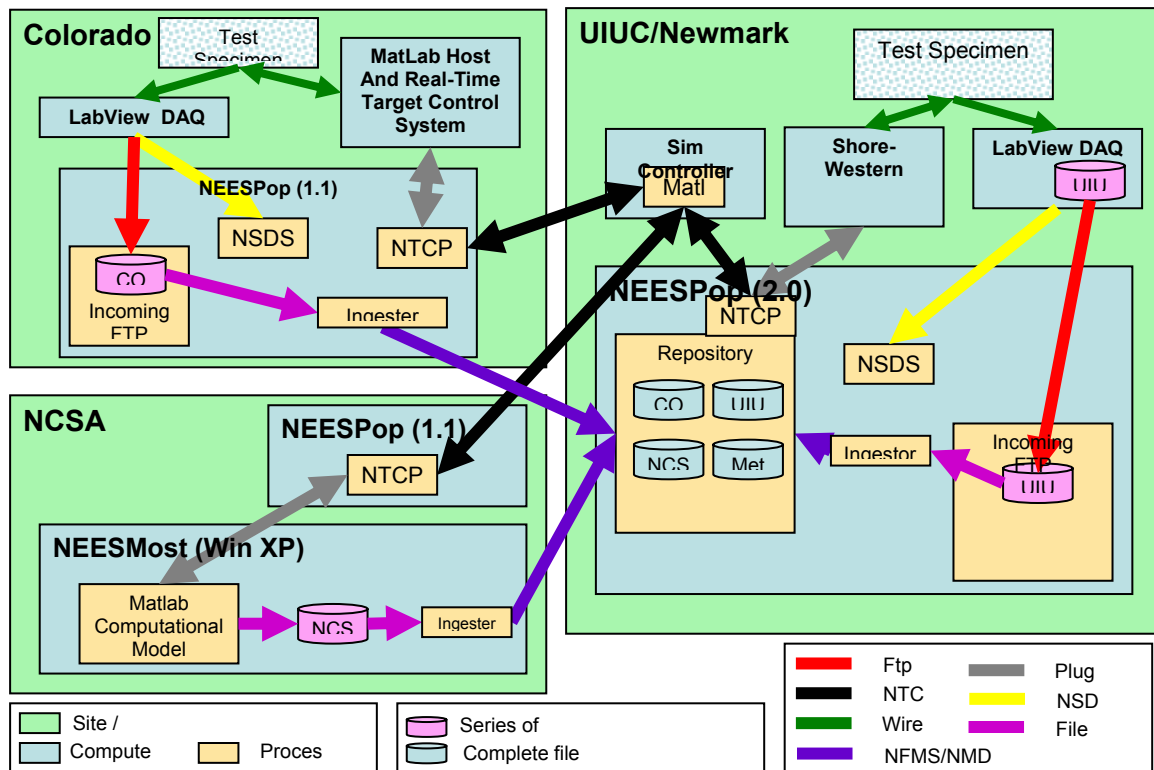


Figure 9: Data flow paths during the MOST experiment.

Once the data ingestion component at a given site detected a file or set of new files in the staging directory, it concatenated all the newly-arrived data into one file fragment. Then, it used NFMS to negotiate a GridFTP transfer between its host machine and the Newmark NEESPop. Once the GridFTP transfer had completed successfully, NFMS reassembled the complete data in the Newmark repository and generated the appropriately updated metadata to represent the data files, so that they could be located and displayed using the CHEF interface to the repository, or downloaded to users' machines. Because GridFTP was used as a transport mechanism, all of the simulation and sensor data that was transported over a LAN during MOST was transferred securely using encrypted channels authenticated with Grid credentials.

2.2.2 NEESML

The NEES Metadata service provides applications with the ability to create, manage, link, and discover metadata objects in the NEES repository. The NEESML language is used to populate the NEES repository with the objects and definitions of object types. The NEESML is an XML format for defining NEES metadata schemas and uploading metadata objects into the NEES metadata repository. NEESML provides syntax for defining object types, object attributes, and relationship between object types, including inheritance. It also provides a means for creating instances of object types, including the ability to link objects together in a variety of ways. Metadata uses objects to define other

objects; this can be done by defining logical relations between the different objects. (For more information on NEESML see the NEESML Reference User Guide, http://www.neesgrid.org/documents/TR_2003_15.pdf.)

Producing the metadata for MOST was a complex task, especially in the absence of a NEESgrid data dictionary. A significant amount of work was dedicated to developing the metadata schema for this test, i.e. defining the types of objects that can characterize the experimental set-up, the sensors, loading conditions, etc. and the relations between them. Similar work was done for describing the parameters of the numerical simulation (nodes, elements, boundary conditions, etc). After these objects were created, assigning the proper values to them was a relatively trivial task. The benefit is that these objects, once defined and implemented in NEESML, can be reused for other tests and will enter the NEES data dictionary.

2.2.3 MOST Metadata

Metadata that is correct and complete is crucial in the context of experiments that involve remote users that do not have direct access to the physical test structure. It is also of outmost importance for archived tests results, since it provides the means to understand where the data is coming from, how it was collected and under what experimental conditions it was produced. Only data that is accompanied by proper metadata can be of real use to researchers that operate in a collaborative environment. Producing it requires discipline, meticulousness and accuracy.

Defining the MOST metadata was accomplished in a two-stage process. In the first stage a conceptual model was created using a simple graphical schema. In this stage, the required objects for defining each model were identified and the logical relations between the objects were established. Figure 10 shows the conceptual model for the UIUC experimental model. The second stage consisted of implementing the predefined objects in NEESML, and writing the required code for the metadata schemas.

For clarity, the MOST metadata was defined using three separate models. In the first model we defined the MOST frame and the associated computational model. The second model described the UIUC experimental model and the set-up at UIUC. >"The third model defined the CU (part of the experiment). A detailed presentation of the process for creating the metadata for the UIUC model will be available in the report "Experimental Setup for a Multi-site Pseudo-Dynamic Physical-Analytical Simulation of a Steel Frame" (H. Seliem and N. Nakata, under development).

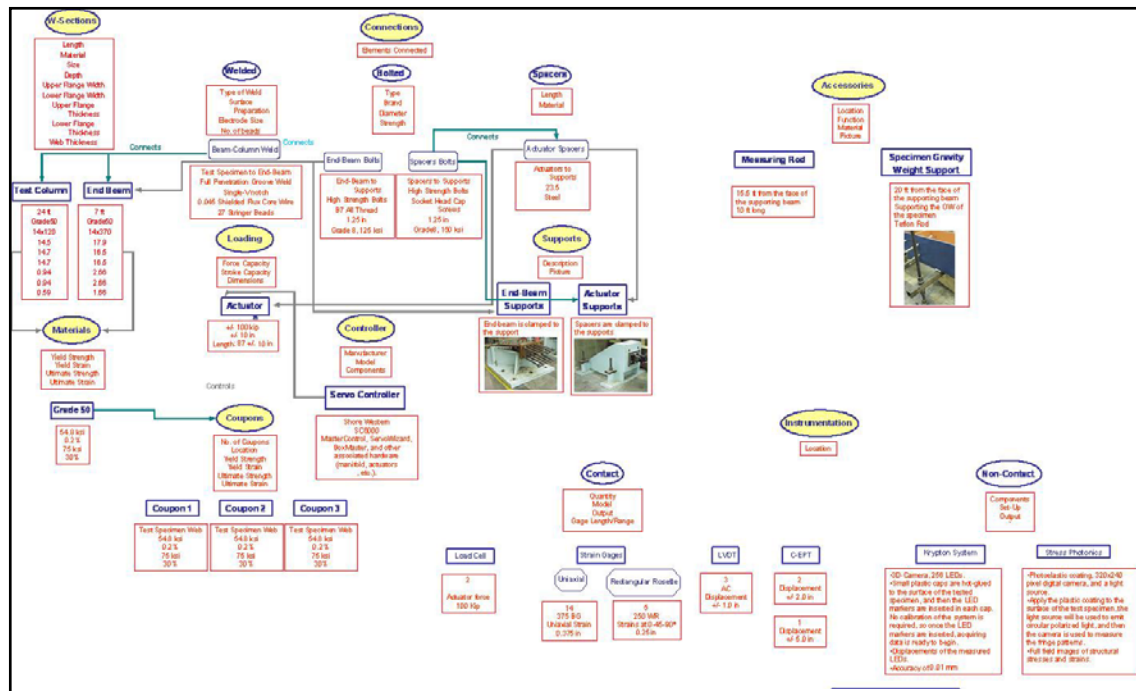


Figure 10: UIUC MOST metadata conceptual model

2.3 NEESgrid Telepresence Mode (TPM)

The telepresence mode provides an interface and display that is intuitive to users using commodity WWW browsers such as Netscape, version 7.x and Microsoft Internet Explorer, version 5.x. Remote telerobotic video cameras with PZT (pan-zoom-tilt) capabilities enable users to remotely view lab space and the physical experiment. Fixed video cameras positioned by local collaborators also allow remote, site-specific observations. High-resolution, static images, uploaded by the local users, can also be viewed. TPM also enables synchronous and asynchronous monitoring of the preparation and construction of tests and test specimens.

Another feature of the NEESgrid telepresence mode is the electronic notebook (E-Notebook) for documenting and sharing experimental data. The E-Notebook is a WWW enabled electronic system with a browser-based graphical user interface (GUI) which allows users to store, review, search and share text, images, and to a limited extent documents. It also allows users to upload high-resolution, static images for remote viewing.

For future experiments, some adjustment will be necessary in the CHEF interface performance and display of all TPM interfaces as well as the E-Notebook. During the MOST experiment, both the TPM interfaces and E-Notebook operated without failure within a CHEF Window. Currently, however, the CHEF Interface only permits a small window for information display, whereas TP modes and the E-Notebook will require full

screen and full scrolling capabilities in the future. The ANL and University of Michigan teams are currently working to resolve this issue.

2.4 Data Acquisition and Streaming Data Services

The data acquisition (DAQ) subsystem provides tools and components for sites to integrate their laboratories into the NEESgrid. Included are a reference implementation in LabVIEW, and an API for use in their DAQ code. Also available are implementations in C for two other inexpensive DAQ devices, and a pure software implementation for use as example code.

Streaming data components allow remote applications to receive streaming data from experiments running within NEESgrid. The NEESgrid Streaming Data Service (NSDS) runs on the NEESpop server at each Equipment Site. It provides a standard network service interface to the (near real-time) streaming data generated by the site-specific data acquisition system.

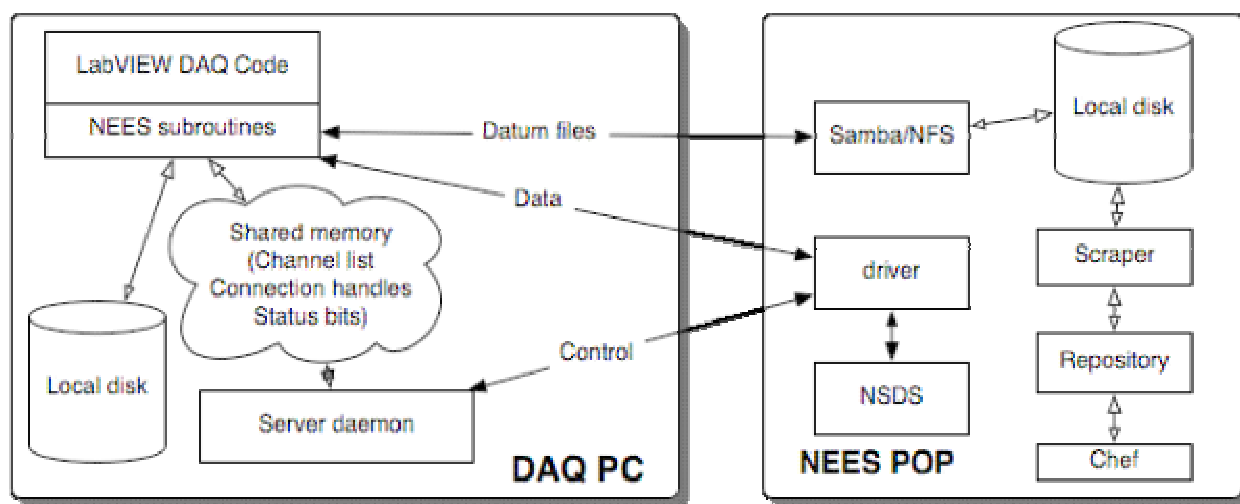


Figure 11: DAQ and NSDS.

As shown in Figure 11, the NEESgrid routines added to the site-specific local DAQ codes handle streaming, connections, data ingester interactions, list maintenance and so forth. Once the data is streamed to the NSDS, it handles the issues (e.g., authorization, authentication, encryption, multiplexing, and subscription management) involved in securely streaming data to a large set of clients.

2.5 Computation and Simulation Services

The NEESgrid Teleoperations Control Protocol (NTCP) and service provides for remote access to control systems (such as servo-hydraulic systems) and simulated control

systems (such as computational simulations that model the actions of servo-hydraulic systems on experiment specimens). The design goals for NTCP were:

- To define a single protocol that could be used to communicate with many different configurations of control hardware and software, thus facilitating the development of distributed experiments involving sites with heterogeneous control systems.
- To define a single protocol that could be used to control both physical experiments and computational simulations, enabling experimenters to switch easily between physical experiments and computational simulations.
- To design a protocol that supports fault-tolerance, so that a minor problem (such as a network interruption) during the course of a distributed hybrid experiment need not cause the experiment to terminate.
- To allow for the negotiation of request parameters prior to execution.

The NTCP is transaction-based: a client (such as the simulation coordinator in the MOST experiment) creates a transaction by sending a *proposal* (a set of requested actions) to the NTCP server. In the most common case, the NTCP server accepts the proposal and creates a new transaction. After the client has had proposals accepted by all NTCP servers involved in a time-step, the client sends requests to these NTCP servers to *execute* the transaction (to actually perform – or simulate – the actions listed in the proposal). Finally, the client receives the transaction results, and can continue on to determine the desired actions for the next time-step. The NTCP protocol supports at-most-once semantics, so that if a client makes a request and does not receive a reply, the client can re-send the request without any danger of the same actions being executed twice.

The implementation of NTCP includes a core NTCP server comprised of two main parts: (i) the generic NTCP services, such as managing the transaction state, and (ii) the control plugin interface, which is used to communicate with backend systems (such as hardware control systems or simulations) to perform actions. The *control plugin interface* is a Java interface definition; a *control plugin* is a Java class that implements the methods specified in that interface. Each equipment site installs the core NTCP server and configures it to use the plugin appropriate for the control system employed at that site.

For the MOST experiment, we provided several components related to NTCP:

- The NTCP server and Java client API.
- The Shore-Western plugin, which was used to control the Shore Western servo-hydraulic system at UIUC,
- The Mplugin, which was used to control MATLAB simulations,
- A MATLAB NTCP client toolbox (written using the NTCP Java Client API)
- A MATLAB NTCP backend toolbox

2.6 NEESgrid User Interface

The NEESgrid unified user interface that allowed remote participants to view the MOST experiment, interact with other users and display the MOST data, is based on the CHEF environment (www.chefproject.org). CHEF, which was developed and supported by the University of Michigan, provided an online collaborative environment as part of the NEESpop distribution.

CHEF components available for users during the experiment included a data viewer, which allowed users to view real-time and stored data and to customize the configuration of the view display; a real-time text chat which allowed participants to discuss the experiment and archived all messages to allow later review; a real-time video feed from both of the physical testing sites; user access to E-Notebooks at both UIUC and Colorado, in which MOST team members recorded lab notes; an NTCP Sample Tool, which allowed participants to monitor the communication between the physical testing sites, the computer simulation, and the simulation coordinator; and screens for posting of discussions (as opposed to real-time chat) and announcements. A detailed account of the CHEF user experience is included in 4.2.1.

3 Problem Description

3.1 The Concept

In spite of the potential of the pseudo-dynamic testing, some problems may arise regarding testing of very large structures that may exceed the capabilities of a single experimental facility. In addition, the lateral resistance of many structures is governed mainly by certain critical parts that suffer the most severe inelastic deformations during an earthquake. In such cases, it may be uneconomical and unnecessary to test the entire structure. Pseudo-dynamic substructure testing offers an attractive alternative in which analytical subassemblages can be combined with a physical test assemblage to simulate the dynamic response of the entire structure. Therefore, part(s) of the structure will be physically tested, while the remaining part will be analytically modeled and handled only within the computer. The pseudo-dynamic substructure approach overcomes many problems that arise from testing very large structures.

Multi-Site Pseudo-Dynamic Substructure testing (MS-PSDS) is a relatively new technique that utilizes the substructuring technique to simulate the dynamic response of the structure using geographically distributed experiments. The entire structure to be tested pseudo-dynamically is divided into various substructures. Each substructure is to be physically tested or numerically simulated at the same time at a different location. The master computer controls the overall experiment and communicates with the test sites and simulation computers via the internet. This experimental technique allows for testing a wide range of large structures, structures that might otherwise be beyond the experimental capabilities of certain laboratory. The Multi-Site Pseudo-Dynamic testing

technique was introduced for the first time very recently in Japan.¹ (need a reference here)

The frame tested in the MOST experiment has beam-column connections similar to those in the SAC Consortium Benchmark Structure shown in Figure 12. The MOST structure is a two-bay frame with one pinned beam-column connection at the right and moment connections for the other columns. To test this frame using the MS-PSDS technique, as illustrated in Figure 13, the left and right columns were tested at UIUC and Colorado, respectively, and the rest of the frame was numerically simulated on a computer at NCSA.

The main design considerations for the frame included the available test configurations and the hydraulic actuator capacities at UIUC and CU. The frame was statically analyzed using SAP-2000 to choose the most appropriate dimensions and component sizes that would fulfill the design considerations. It was decided that the support of the middle column would be a hinged support in order to reduce the required hydraulic actuator force to yield the tested columns. The chosen dimensions and cross-sections were as follows: columns height (H) equal to 4572 mm (180 in); beams span (L) equal to 7315.2 mm (288 in); column cross-sections were W14x120; and beams cross-sections were W10x60. All members are steel Grade 50.

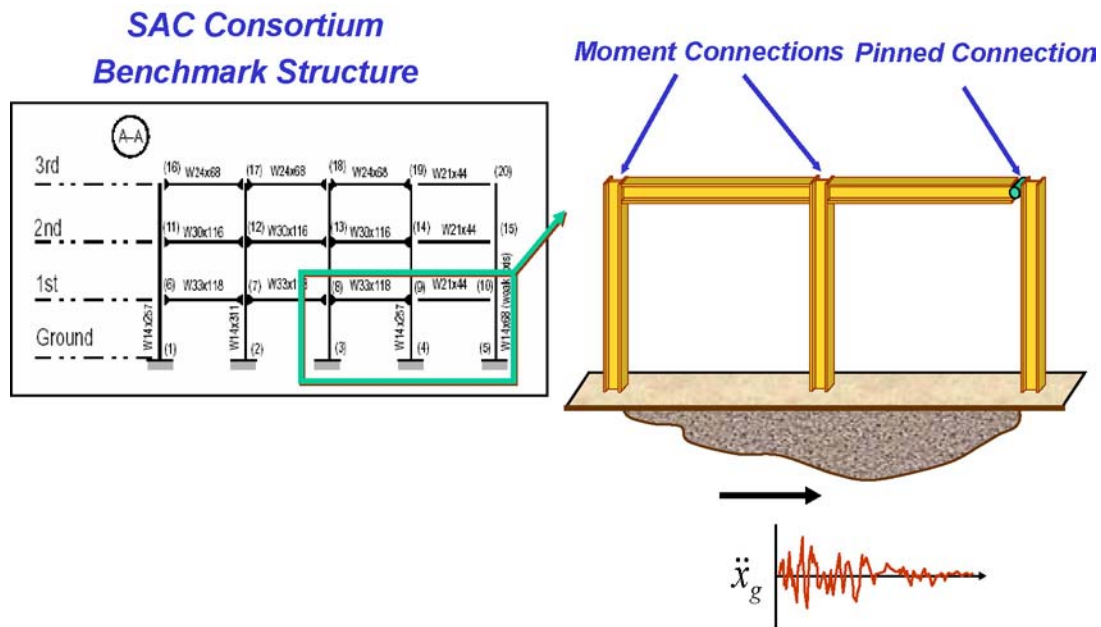


Figure 12: SAC frame tested in the MOST experiment

¹ E. Watanabe, K. Sugiura, K. Nagata, T. Yamaguchi and K. Niwa, "Multi-phase Interaction Testing System by means of the Internet," Proc. of 1st International Conference on Advances in Structural Engineering and Mechanics, Seoul, Korea, 1999.

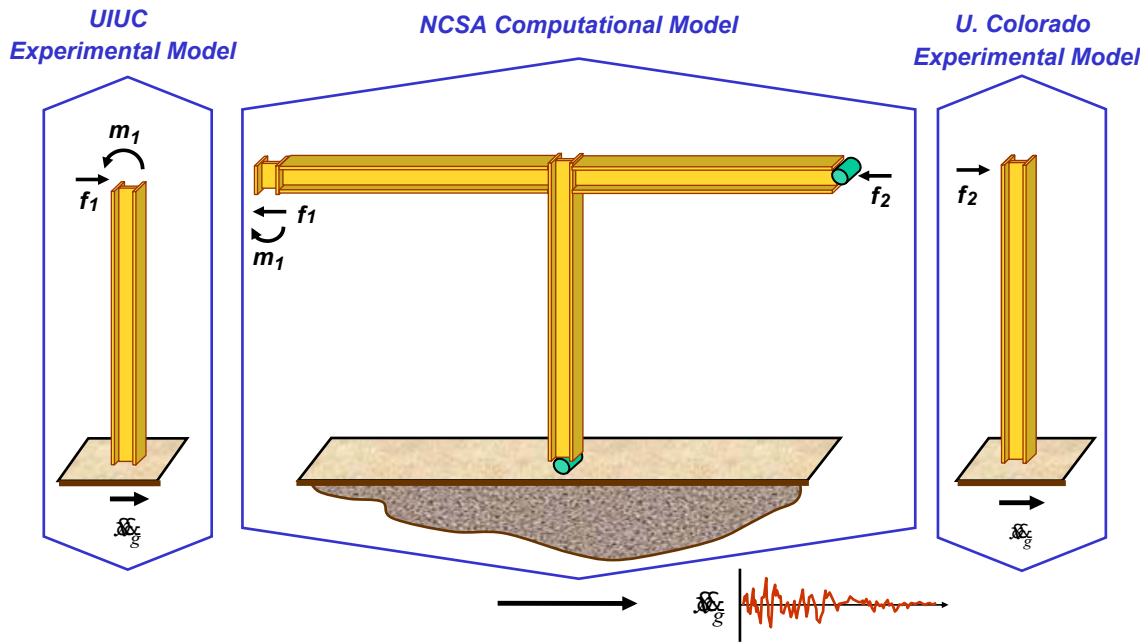


Figure 13: Substructure for MOST experiment

3.2 Experiment Setup

3.2.1 University of Illinois at Urbana-Champaign

The left column of the frame was physically tested at Newmark Civil Engineering Laboratory at UIUC. Because of the fixed connection between the column and beam, the column acts in double-bending. Therefore, moments at the top and the base of the tested column must be developed at the same time. To achieve the required loading, two hydraulic actuators were used and the tested specimen was longer than the actual column, as shown in Figure 14.

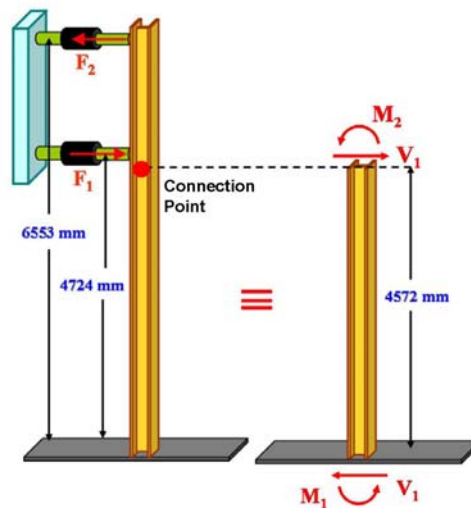


Figure 14: Column tested at UIUC

It is worth noting that it is the column response (displacement, rotation, lateral force and moment) at the control point that needs to be used for the overall MOST frame simulation. Because the column control point is at a different location from the attachment points for the actuators, additional challenge is imposed on the servo-hydraulic control system that will be discussed in detail later in this section.

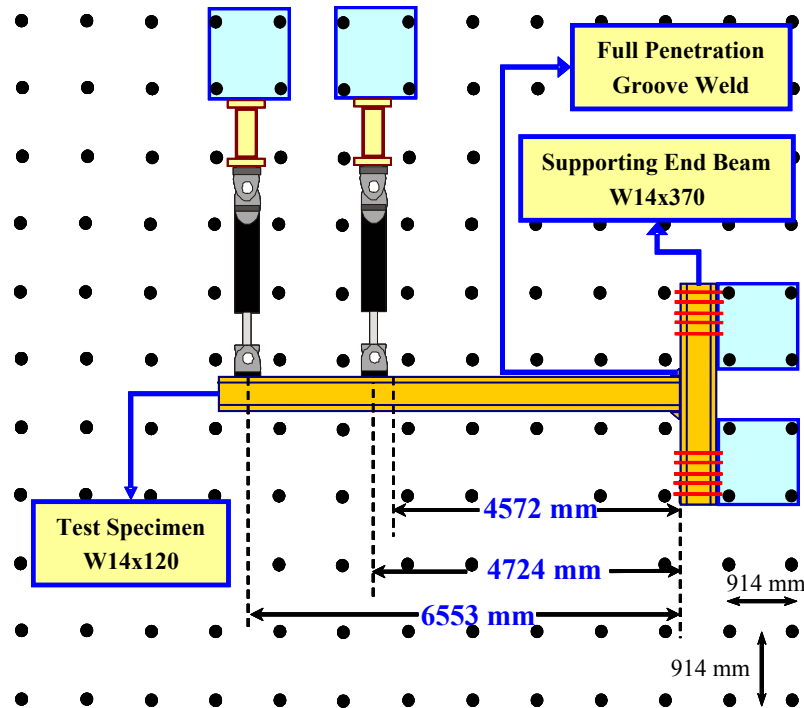


Figure 15: UIUC experiment layout

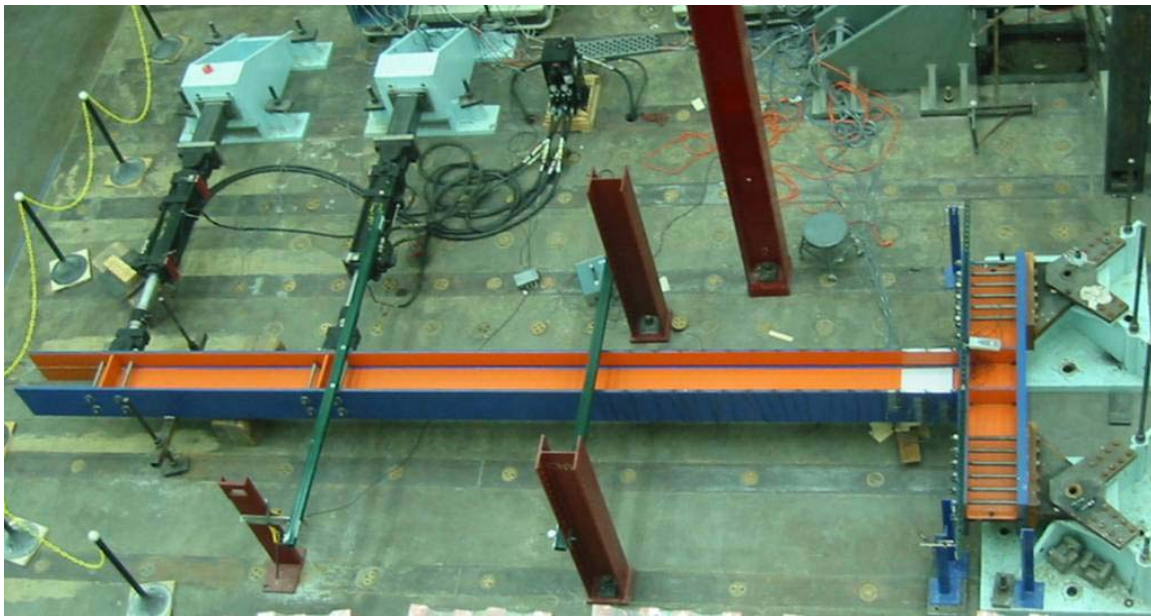


Figure 16: UIUC experimental set-up

Figure 15 and Figure 16 show the experimental setup and layout at UIUC. The column was tested horizontally to eliminate the difficulties, costs and dangers associated with hoisting actuators more than 6 m in the air. Two steel supports were used to keep the two 100-kips, ± 5 -inch-stroke MTS hydraulic actuators in place, and two steel spacers were used to connect the actuators to the supports. The column was directly welded to the end-beam using a full-penetration groove weld to assure the column fixity at the base, and the end-beam was then bolted to the two steel supports. The rotation of the supporting end beam was measured during the experiment to account for any rotation of the end beam that might occur due to the end-beam supports flexibility. The lateral movement was also monitored to verify the boundary conditions of the tested column. Three Sensotec miniature AC Linear Variable Differential Transformers (LVDT)s, two for rotation and one for lateral displacement, were used to measure the rotation and the lateral movement of the end beam.

Two types of advanced non-contact sensing systems, the *Photoelastic Stress Measurement System* and the *Krypton Measurement System*, were used during the MOST experiment in parallel with traditional sensors, such as LVDTs, cable extension position sensors or string pots, and strain gauges.

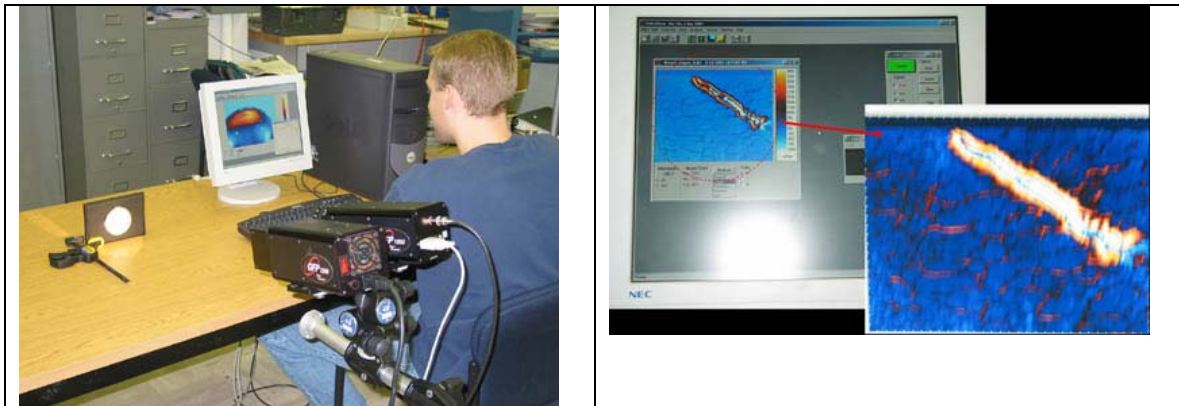


Figure 17: Photoelastic Stress Measurement System

Stress Photonics Gray Field Polariscopes (GFP-1000), as shown in Figure 17, is a modern Photoelastic Stress Measurement System that quickly produces full-field images of structural stresses and strains based on the principles of photoelasticity. After several photoelastic sheets were attached to the surface of the tested specimen (Figure 18), a light source was used to emit circular polarized light, and a 320x240 pixel digital camera was used to measure the fringe patterns to create a full-field strain map.

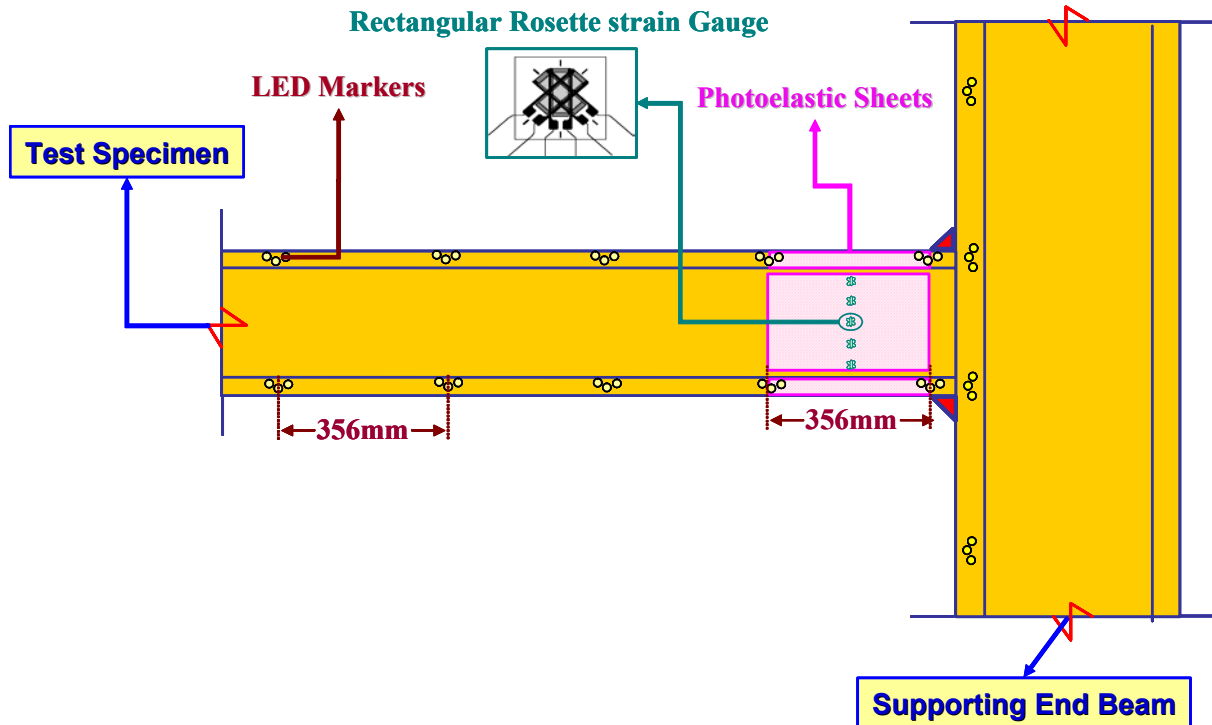


Figure 18: Instrumentation of the UIUC column



Figure 19: Krypton Measurement System

The Krypton Measurement System, shown in Figure 19, measures the position of light emitting diodes (LED) inserted in small plastic caps hot-glued to the surface of the tested column (Figure 18). Because the LED markers were installed on the supporting end-beam, the Krypton System was also able to monitor its lateral movement and rotation during the experiment in addition to LVDTs.

Measurements from strain gauges were also compared with those obtained from the Stress Photonics and the Krypton Systems. Fourteen uniaxial strain gauges of 10 mm

gauge length were attached to the tested column flanges to measure the longitudinal strains in the flanges. The longitudinal strains obtained from the strain gauges were directly compared with the longitudinal strains obtained from the Krypton System. Rectangular rosette strain gauges of 6 mm gauge length were attached to the tested column's web to measure the in-plane shear strains. The measured in-plane shear strains obtained from the rectangular rosette strain gauges were directly compared to those obtained from the Stress Photonics. Figure 18 shows the location of the rectangular rosette strain gauges.

Cable-Extension Position Transducers (CEPT) were used to measure the lateral displacement and the rotation at the tip of the tested column at the control point. The measurements of the CEPT were used in the outer hydraulic control loop.

As previously mentioned, the column control point at the UIUC site is not located at one of the actuator attachment points. Three CEPTs were attached to the column at the connection point to measure the column lateral displacement and to a steel arm mounted on the column at the connection point, as shown in Figure 20.

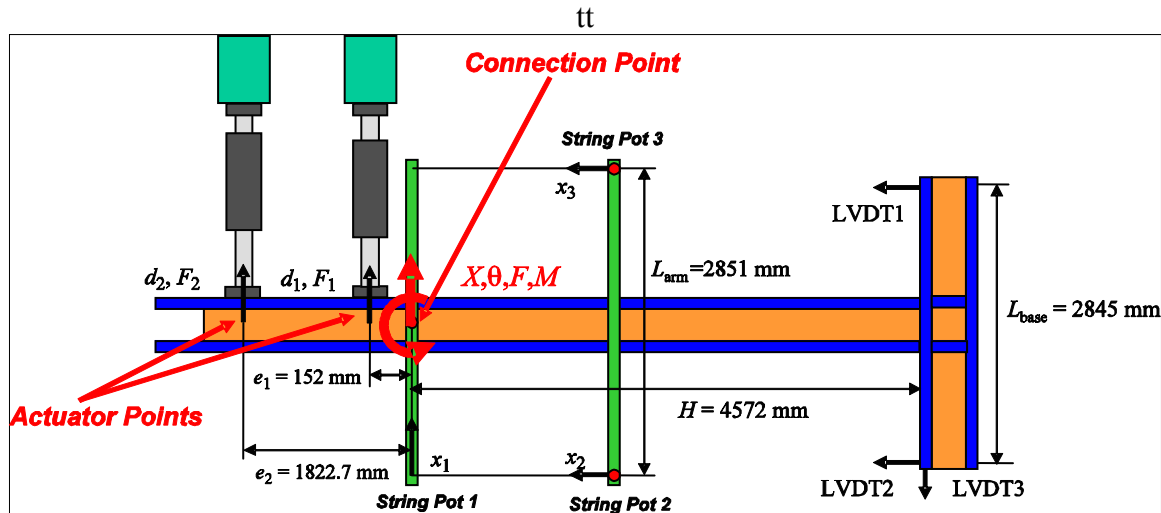


Figure 20: Measurement of control point displacement and rotation

The rotation of the column at the control point was obtained as the ratio between the difference of these two CEPTs and the length of the steel arm.

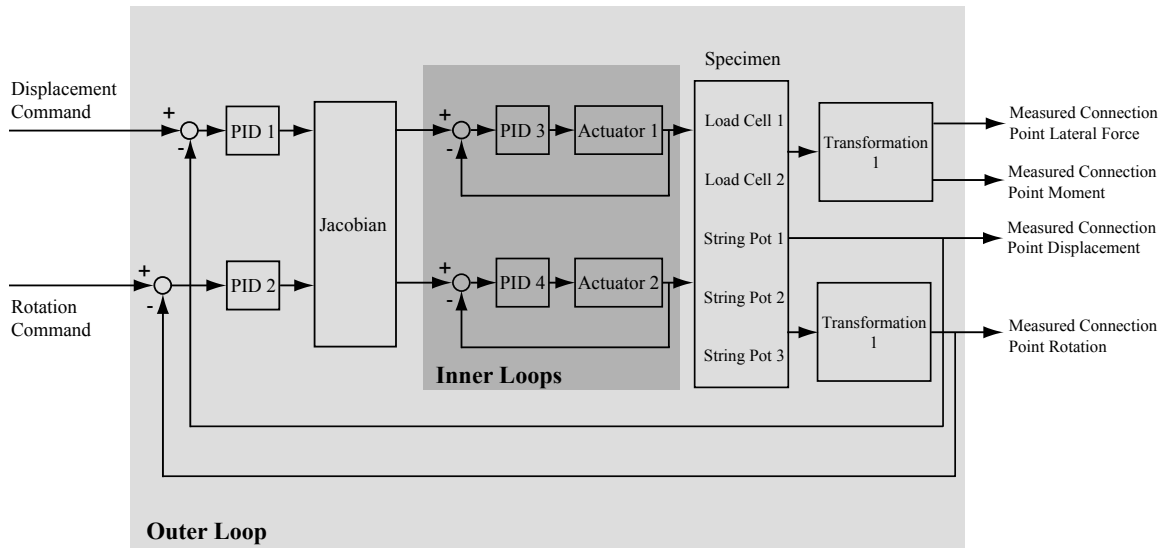


Figure 21: Servo control diagram for the control point movement

The two actuators at UIUC were controlled by a servo-controller produced by Shore Western. As illustrated in Figure 21, the controller has two inner loops to dictate the movement of each individual actuator in displacement control mode. In the outer loop, the error between the commanded and measured lateral displacement and rotation of the column control point is first calculated. After the error signal passes PID controllers, a Jacobian matrix is used to convert the PID controller outputs from the physical coordinate to actuator coordinate. The outputs from the Jacobian are then input to the servo controller inner loops to adjust the positions of two hydraulic actuators such that desired movement at the column control point can be achieved. The outer loop was implemented by a “Box Master” program developed by Shore Western.

A Windows 2000-based PC connected the UIUC site with remote sites. The SI team developed a Shore Western plugin that provided the interface between NTCP and the Shore Western controller to specify the actuators movements to achieve the target displacement and rotation. The DAQ system at UIUC was event-driven: i.e., the acquired data was not continuous, only the data at the instant when the target displacement and rotation were achieved was recorded and stored in the central repository. Thus, at the same time movement was initiated and measurements sent, at the same time, the servo controller also signaled the DAQ PC to trigger the DAQ's acquiring the data at that instant.

The sensor signal from strain gauges, LVDTs, and string pots came into the DAQ through their corresponding National Instrument SCXI modules. LabVIEW sent the acquired sensor data out using its internet toolkit. Two paths were used for the MOST experiment, one through NSDS and the other through Samba, which mounted the remote file space as a network drive on the DAQ PC where the LabVIEW can directly write data files. Once the file was written, it was moved to the central repository by the data

ingestion program and remote users could then view the experimental data to remote users via the CHEF interface.

Due to current development stage limitations, image files from advanced non-contact sensing systems were stored on the local web server at UIUC. Remote users were able to access these files during the experiment through a link in the CHEF interface.

3.2.2 University of Colorado at Boulder

The right column of the frame, shown in Figure 22, was physically tested at the Structures and Materials Testing Laboratory at CU. The column is a cantilever column because of the beam-column pin connection in the overall SAC frame tested in the MOST experiment. As illustrated in Figure 22 and Figure 23, the column was tested in a horizontal position and was rigidly connected to a vertical supporting steel structure suppressing all translational and rotational degrees of freedom. The support structure was attached to the concrete testing floor. Tri-plate stiffeners with the dimension of $270 \times 127 \times 13$ mm were added at the base of the tested column to assure the column fixity, and the tested column height was measured from the location of the tri-plate stiffeners, as shown in Figure 22. The column has an effective height of 4572 mm and was driven by a 110 kips, ± 5 inch stroke MTS actuator during the experiment.

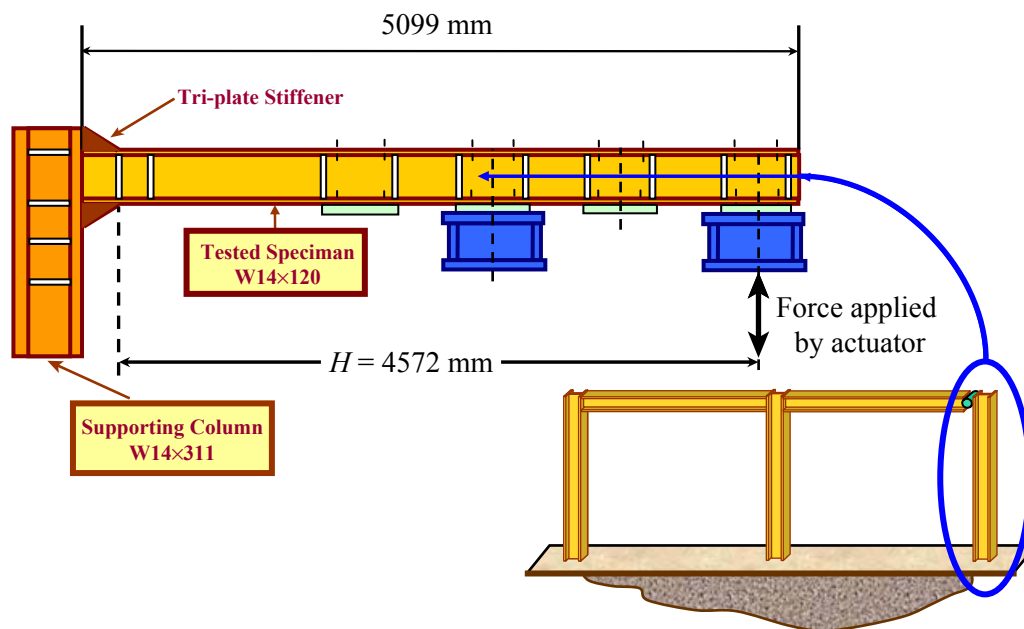


Figure 22: Schematic of the CU experimental setup

Four different types of instrumentation were utilized in the MOST experiment at CU. The displacement of the cantilever beam was measured by the actuator's internal LVDT and an externally mounted digital encoder type displacement sensor manufactured by Heidenhain. The digital encoder was positioned adjacent and parallel to the actuator. The

load applied to the test structure was measured using the hydraulic actuator's built-in load cell. This load cell is a Wheatstone bridge type and its manufactured, calibrated and maintained by the MTS. Four Micro Measurement EA-06-250AE-350 uniaxial strain gauges were also used to measure the strain of the top flange of the cantilever beam. These strain gauges have a gauge length of 10 mm and are open-faced with a 1 mil tough, flexible polyimide film backing. The strain gauges were located 305 mm away from the face of the supporting column.



Figure 23: Overview of the CU experimental setup

To connect the CU equipment site with remote testing sites for the MOST experiment, a *MOST Control Host PC* running a MATLAB script was employed to relay commands and measurement between the CU experiment and the Simulation Coordinator located at the NCSA. This communication is carried out using the NTCP protocol developed by the SI team. The MATLAB script running on the *MOST Control Host PC* also provides the final check on the integrity and validity of the control command for each step. The control command then relay to the *MOST Control Target PC*. The target computer runs a real-time Simulink application that transforms the target displacement into a series of smooth ramp commands and writes these commands to the high-speed fiber optic Shared Common Ram Network (SCRAMNet). The *MTS Real-time Digital Control System* reads the ramp commands from the appropriate location in the shared memory and directs the movement of the hydraulic actuator. The data acquisition program written in LabVIEW measures test structure response and sends measured displacement and force back to the Simulation Coordinator. The flow chart of the CU control system is illustrated in Figure 24.

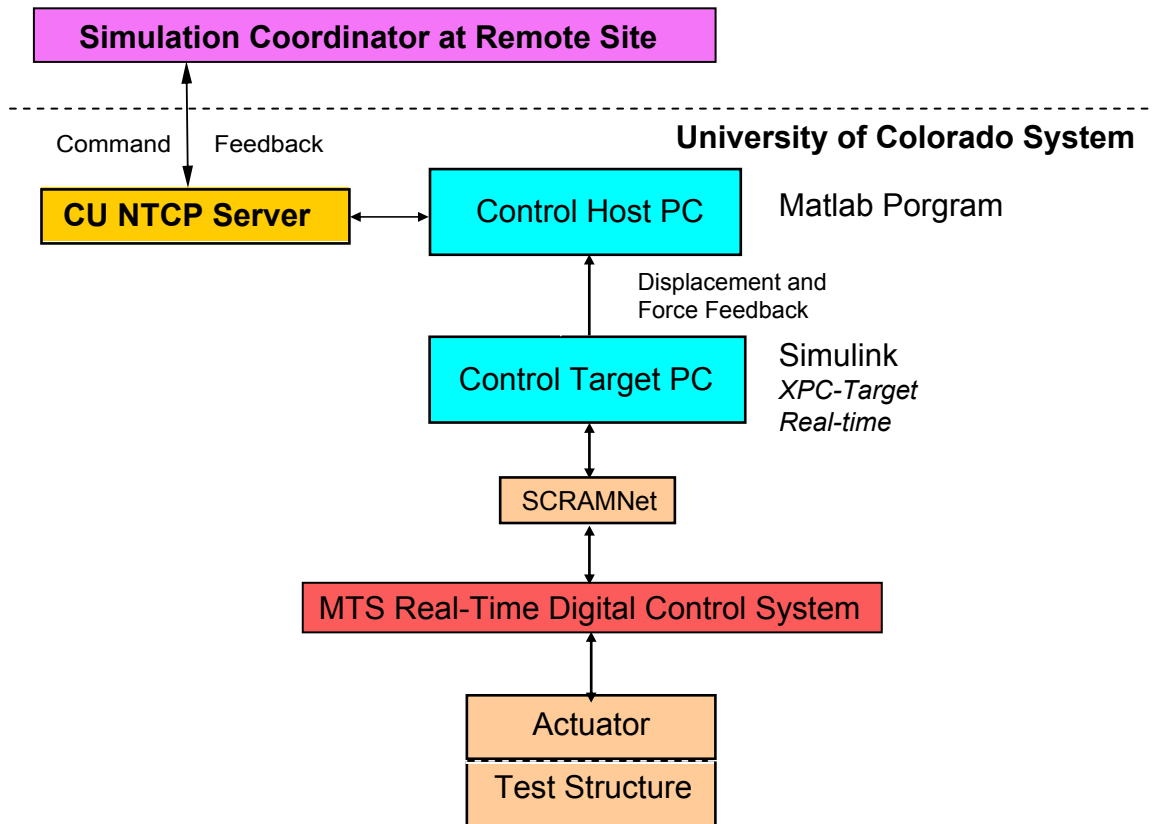


Figure 24: Flow chart of CU control system

During the experiment, the structural response was also streamed to remote users and at the same store in the main data repository for archiving. The data for the CU portion of the MOST experiment was arranged in columns and each row corresponded to a set of data at a given instant in time. A data file in this format was generated each time when a new set of data became available, *i.e.* at the conclusion of each displacement ramp.

3.2.3 National Center for Supercomputer Applications

At NCSA, a Pentium 2.4 GHz Windows based machine with 512 MB memory was utilized to simulate the rest of the frame using a program written in MATLAB. This machine acted as a master computational module and communicated with the Simulation Coordinator through the NTCP. The simulation scheme used was an α -operator splitting (α -OS) method.^{2 3} The α -OS method is a non-iterative implicit time integration scheme. As such, it is a good candidate for pseudo-dynamic testing since it provides the unconditional numerical stability while preserving the implementation simplicity of explicit schemes.

² Didier Combescure and Pierre Pegon, "Alpha-Operator Splitting time integration technique for pseudodynamic testing Error propagation analysis," *Soil Dynamics and Earthquake Engineering* 16 (1997)427-443

³ Masayoshi Nakashima, et al. "Integration Techniques for Substructure Pseudo Dynamic Test," *Proceedings of the Fourth U. S. National Conference on Earthquake Engineering* (1990)2:515-524.

3.3 Substructure PSD Testing Software

3.3.1 Design Philosophy

Because of the multi-site nature of substructure PSD testing, a modular software framework was developed, including four distinct modules:

- Simulation Coordinator
- Master Computational Module
- Slave Computational Module
- Experimental Module

As shown in Figure 25, the moderator of this software is the Simulation Coordinator. This architecture allows each module to run in geographically remote locations and only requiring each module to interface with Simulation Coordinator through NTCP. This approach provides great flexibility for execution of substructure PSD experiments. Moreover, in this design the Slave Computation Module and Experimental Module are interchangeable. Therefore, experiments can be debugged using the computational simulation modules, thus minimizing the risk of unexpected damage to the physical structure/hardware.

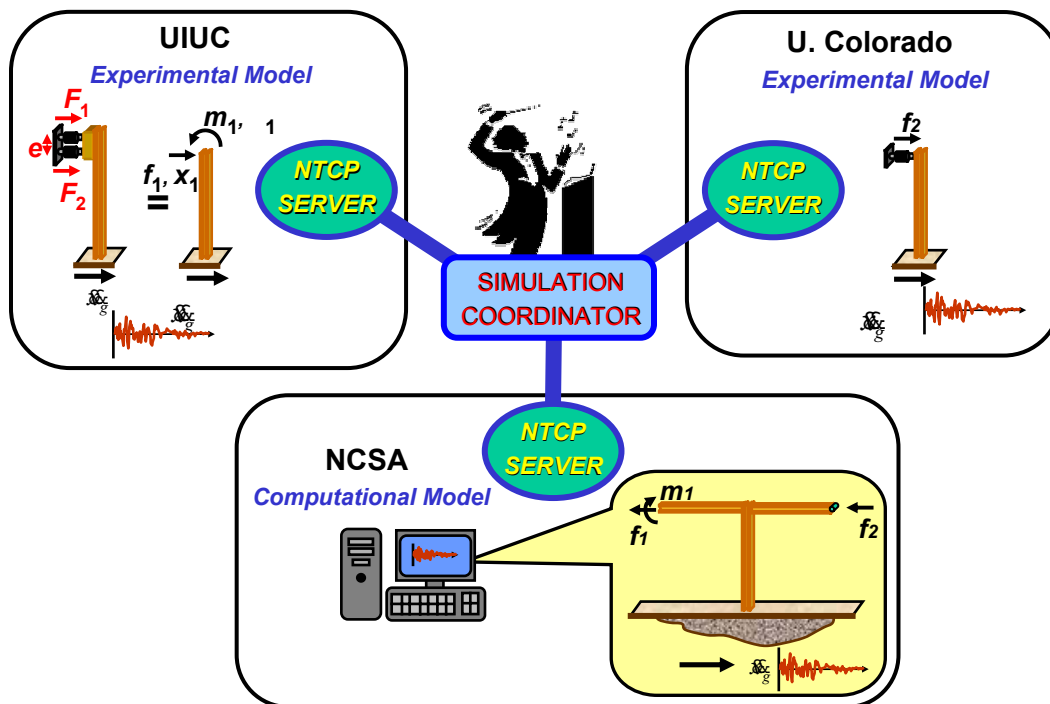


Figure 25: Modules of PSD software

3.3.2 Simulation Coordinator

The Simulation Coordinator is the command center during the experiment; all modules communicate directly with the Simulation Coordinator through NTCP. The Simulation Coordinator sends the requests for specific action to and receives replies from

individual modules. In addition, the Simulation Coordinator handles exceptions during the experiment, such as losing a network connection, invalid response, etc.

3.3.3 Master Computational Module

The master computational module performs the numerical integration, combining results from the experimental or the slave computational node. At each time step, the master computation module numerically calculates the frame response and sends the target displacement and rotation of the control points to the Simulation Coordinator through NTCP. It waits until receiving the measured force and moment response before the numerical integration proceeds. The integration scheme used in the simulation is the α -operator splitting (α -OS) method referenced in Section 3.2.3.

3.3.4 Slave Computational Module

Although the slave computation module was not used during the formal MOST experiment, it was indeed used extensively during the development stage for debug purposes. Slave computation modules were developed to numerically simulate the UIUC and CU columns, respectively. The input to the slave computational module is the target displacement and rotation (UIUC only), and the output is the corresponding measured displacement, rotation (UIUC only), force and moment (UIUC only) after the target displacement has been applied to the structure. Because the input and output for the slave computational module is the same as that of experiment module, it can be seamlessly interchanged with the experiment module. Note that this feature provided flexibility to account for the different schedules of the hardware and software development teams.

3.3.5 Experimental Module

The experiment module includes hardware control system and communication software for the remote sites. For this experiment, only the target displacement and rotation (UIUC only) and measured displacement, rotation (UIUC only), force and moment (UIUC only) are interfaced with remote sites through NTCP communication software. Note that the input and output of the experimental module are the same as that of the slave computational module; it can be interchanged with computational module without any extra effort.

4 Execution of Distributed Simulation

4.1 Verification and Testing

Prior to the experiment itself, the MOST setup underwent a series of tests that focused both on individual components and on the entire system. For the NTCP server and java client API, Java module tests, or JUnit tests, were created and run throughout the development process.

Various components were tested in several different combinations to evaluate their integration, culminating in a dry run of the entire experiment. Telepresence components were all operating prior to MOST, and their operation was verified on the day of the demonstration. Data service components, including NMDS and NFMS, underwent testing during the development process and specifically in advance of MOST. NMDS passed a set of JUnit-like tests as part of the build process; NFMS passed a set of actual JUnit tests. As a result of a series of testing phases in which data was generated and ingested first from a development machine and then from the sites, the data ingester's fault tolerance was greatly improved. For the streaming code, testing was performed with the new DNDTester (<http://www.mcs.anl.gov/neesgrid/dndtester/>) application to verify functionality and examine external triggering at Argonne, UIUC, and CU, where the code was modified to communicate with CU's Scramnet/MTS system. Finally, a set of tests is currently under development for the CHEF repository browser.

Beginning July 24, the MOST team conducted a series of test runs in preparation for the actual experiment on July 30. The first test run involved only the simulation coordinator of the MOST experiment and its interaction via NTCP servers with simulations at UIUC and Colorado. The second rehearsal run tested the Shore Western servo-hydraulic system, control software, and plugin at UIUC, while MOST simulations ran on servers at NCSA and Colorado. This test run uncovered a small problem in the simulation coordinator (the relaxation test was failing), which resulted in a small change in that program before the final dry run. The final dry run, which ran successfully, involved almost all the components that would be demonstrated in the July 30 run, linking the simulation component at NCSA with the two experimental columns at UIUC and CU.

4.2 Accessing Results

4.2.1 Using NEESgrid CHEF Tools

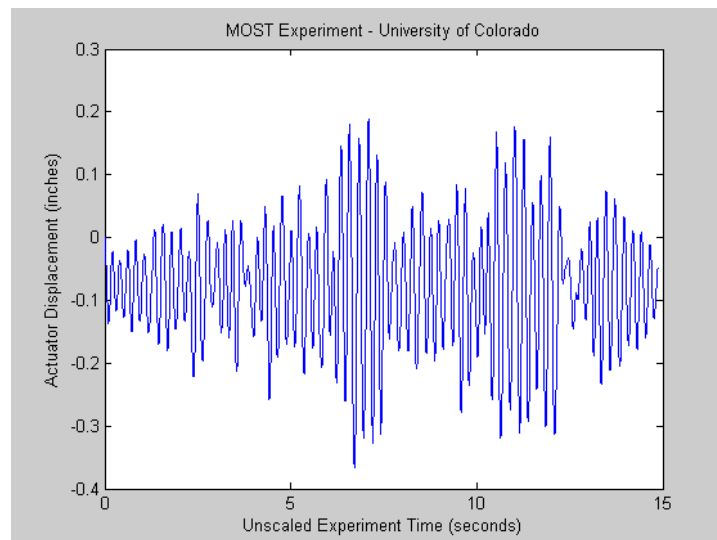


Figure 26: Measured displacement response at CU, as viewed locally.

In the laboratory, data can easily be visualized using local data viewers (). However, for remote users, access to the MOST experiment is provided through the CHEF interface. To observe the test and collaborate with others, users access tools via logging in to MOST. Each available tool has a button on the left side of the window.

The tools most relevant to the MOST experiment are:

- Data Viewer
- Chat
- Video
- E-Notebook
- NTCP
- Users Present

4.2.1.1 CHEF Data Viewer

Users can view real-time and stored data using the Data Viewer tool. In order to use the Data Viewer, users first select the MOST experiment “Event” from the Data Viewer tool (Figure 27).

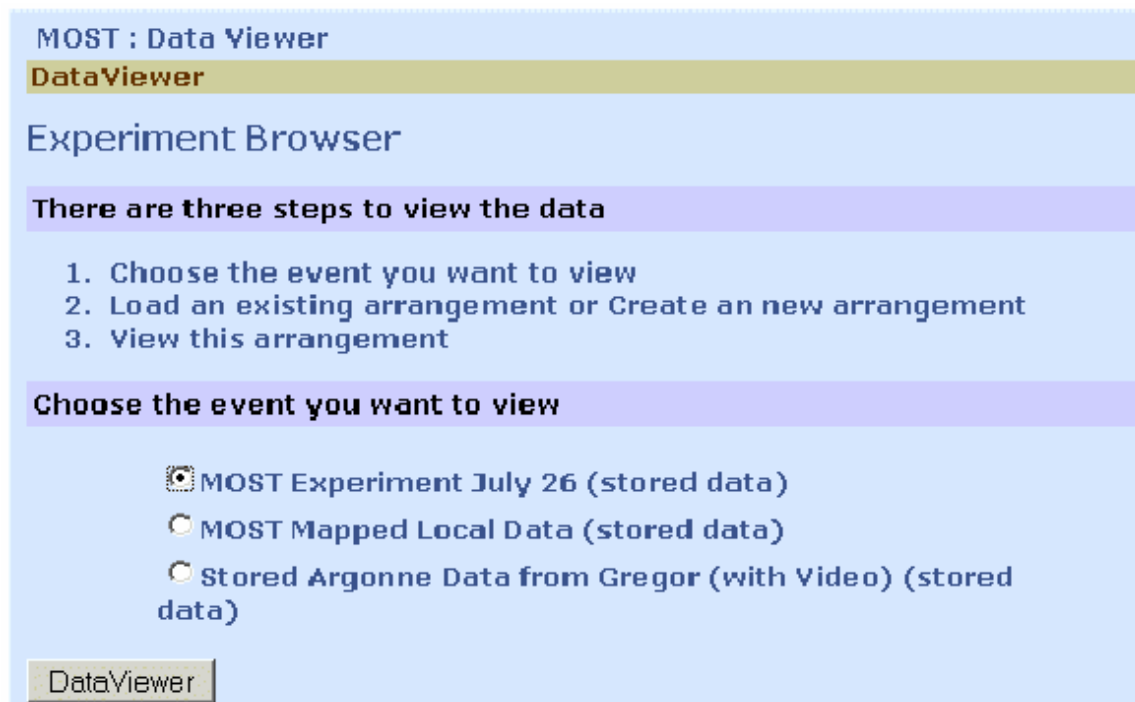


Figure 27: Selecting an event to view with the Data Viewer tool.

Users then click the “Add View” button to construct a series of "views" into an "arrangement" (Figure 28). A "view" refers to a single graph or view of a model,

whereas an "arrangement" is a collection of one or more views that are displayed together.

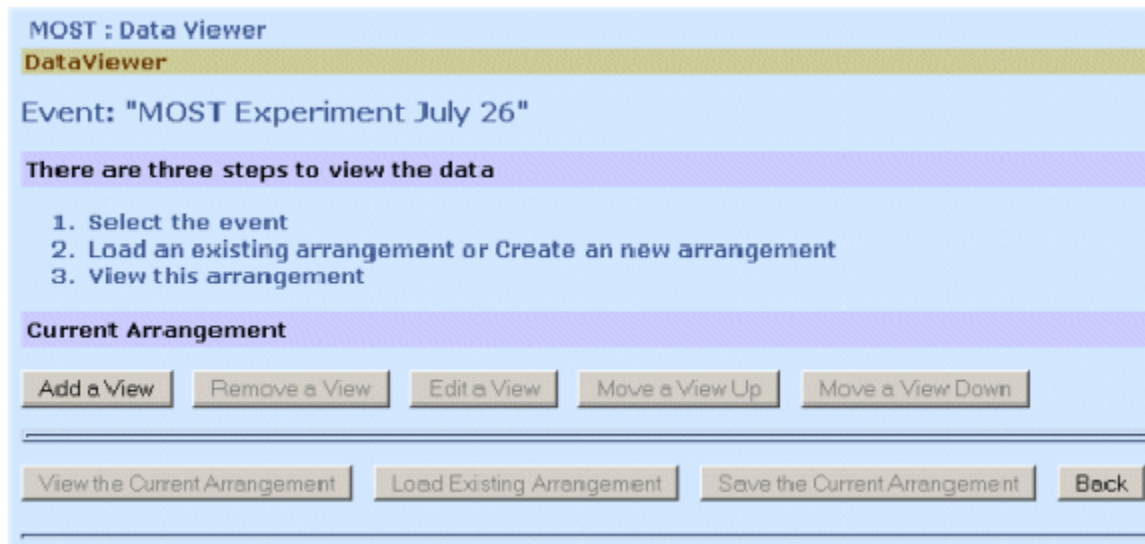


Figure 28: Add a new view of the experiment to your arrangement.

When adding a view, users selected either an XY-plot or a 2D dynamic structural model (Figure 29). Selecting an XY-plot required determining the variables to be represented on each axis, the type of diagram, and its color.

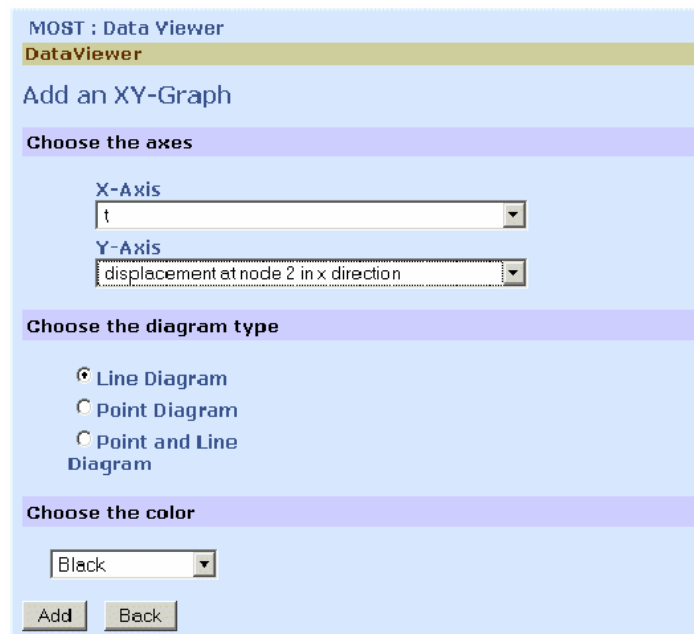


Figure 29: Adding a view and selecting variables.

Arrangements of one or more views may then be saved or viewed; the Data Viewer automatically organizes a given arrangement to allow users to see each of the views. At the top of the Data Viewer, a set of VCR buttons allows users to play, pause, rewind, and fast-forward the data viewer, while at the bottom of the data viewer a clickable timeline enables users to see the state of the Data Viewer at any given time point (Figure 30).

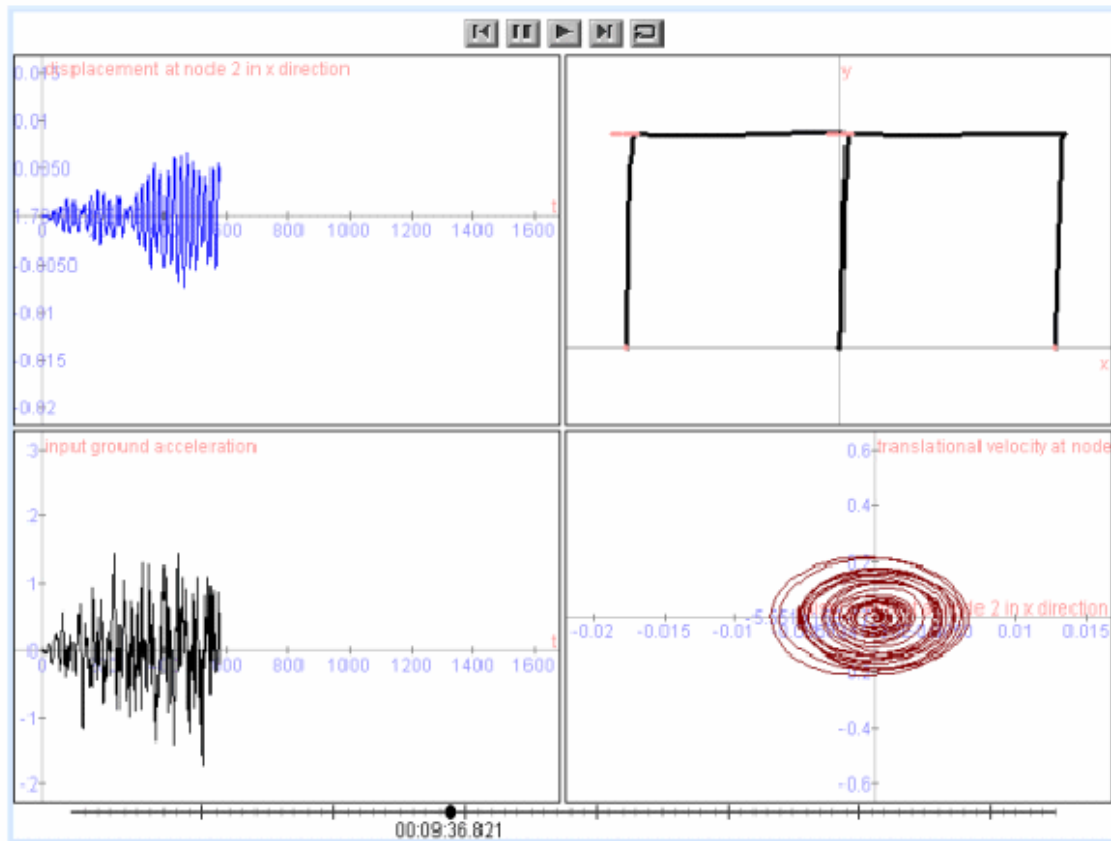


Figure 30: Arrangement of multiple views in the Data Viewer

4.2.1.2 CHEF Chat

A real-time text chat allows participants to discuss the MOST experiment. Unlike some other chat tools, messages are *persistent*, or continue to exist after their senders have logged off), so that participants who arrive late or after the experiment ended can scroll back and read messages that were posted before they logged in.



Figure 31: Reviewing past chat interactions about the MOST experiment.

On the right side of the window in Figure 31 is a list of users currently using the chat tool (this is different from the list on the lower left corner of the CHEF screen, as in Figure 32). On the left side of the chat tool is the conversation. Each user's contributions are preceded by his or her name and the time and date of the message. Users post messages by typing into the text box at the bottom of the chat tool and clicking the "Send" button (Figure 32).

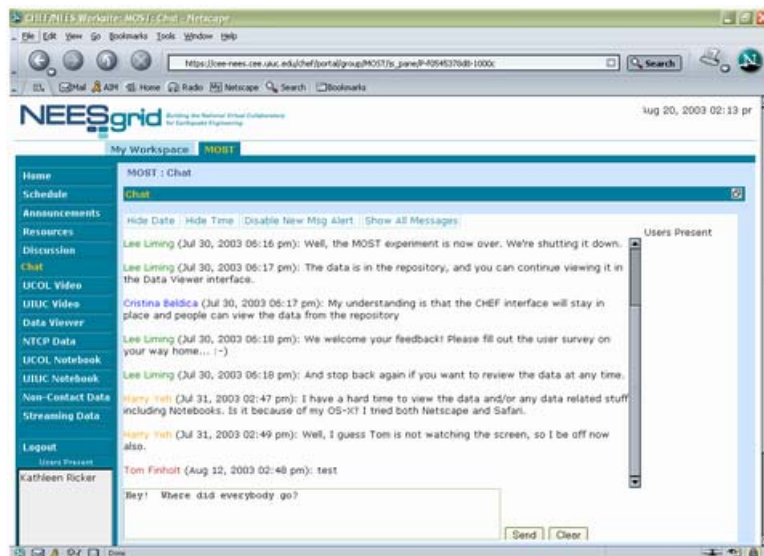


Figure 32: Posting a chat message.

4.2.1.3 CHEF Video Interface

During the MOST experiment, real-time video from both of physical testing sites was available, with at least one accessible camera at each site. To access the camera at either Colorado or UIUC, users could click on the appropriate Video button (Figure 33).



Figure 33: Real-time video from Colorado in the CHEF interface.

The video streams relied on push technology supported by recent versions of Netscape; Internet Explorer users may have had some difficulties.

4.2.1.4 CHEF E-Notebook Interface

Researchers on the MOST team kept their lab notes using the NEESgrid E-Notebook (Figure 34). Each site's E-Notebook may be accessed through the relevant Notebook button and contains many pages modified as interesting observations were made.

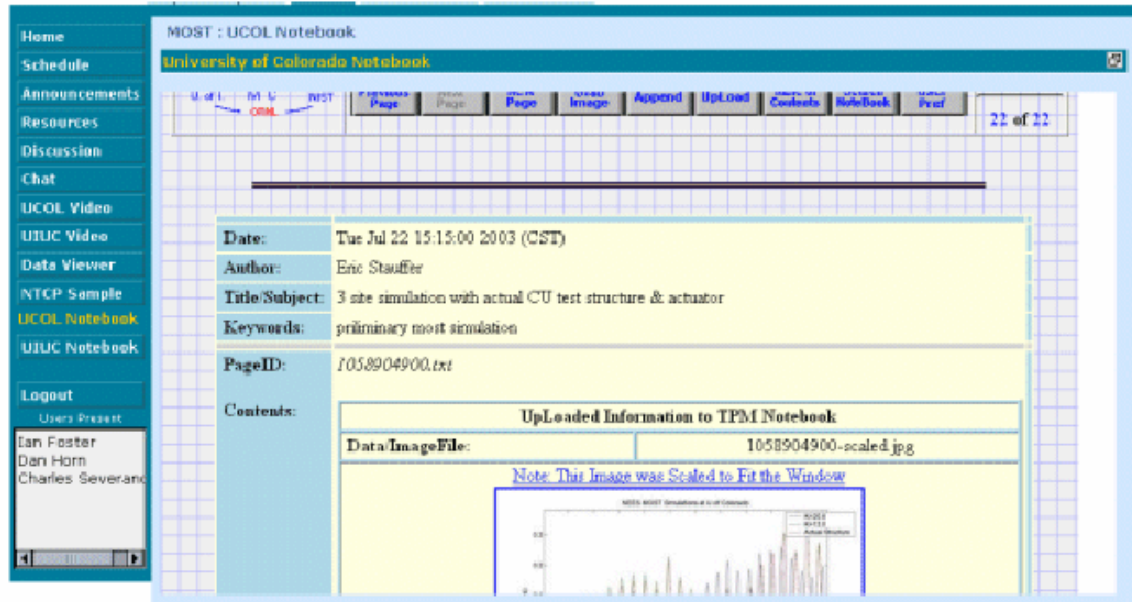


Figure 34: A screenshot from the University of Colorado (UCol) E-notebook displaying experimental data from the MOST experiment.

4.2.1.5 CHEF NTCP Sample Tool

The NTCP sample tool (Figure 35) shows remote user how information flows between different “sites”, either physical testing or numerical simulations. Although during the MOST experiment the NTCP sample tool did not display live data, users could click on any of the arrows in the diagram to see example messages passed back and forth, to get a sense for the communication between the different components while the experiment was running. In future experiments the NTCP Sample tool will allow participants to monitor live communication between these components.

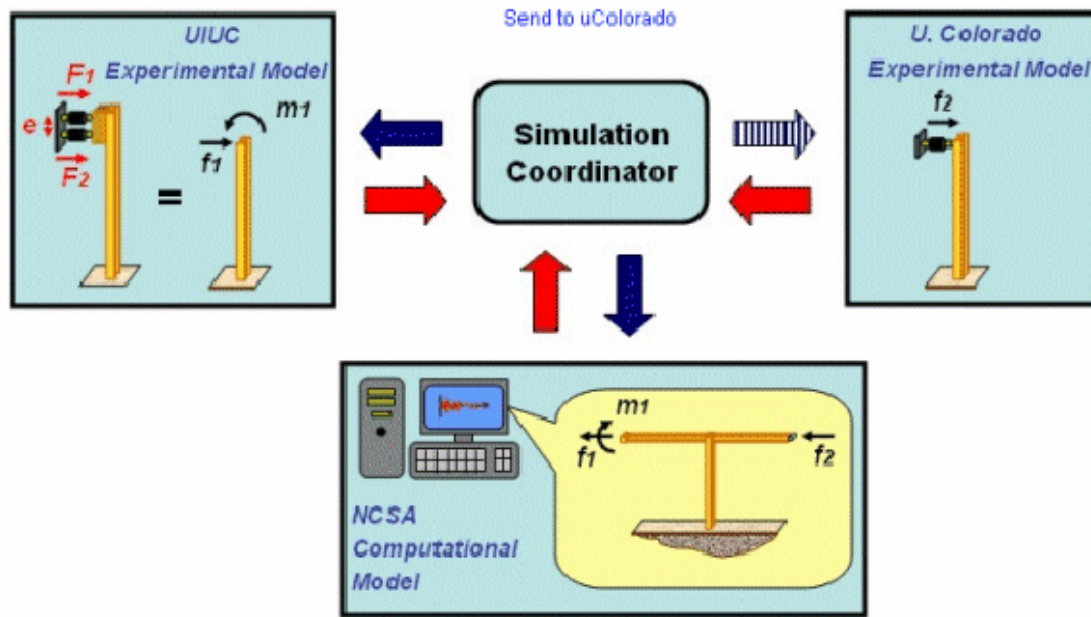


Figure 35: NTCP Sample Tool


4.2.1.6 Users Present

On the lower left corner of the CHEF window is a list of all the users currently online. This feature allows participants to know which collaborators are currently participating.



Figure 36: Users Present

4.2.1.7 Efficiently Using Multiple CHEF Tools

Users wishing to use more than one tool at a time may, for instance, view a video feed while chatting with other participants and monitoring sensor data by activating the float button () in the top right corner of most CHEF tools. When it is pressed, the active tool "floats" into its own resizable window. The main browser window may then be used to access other tools, many of which may also be floated (Figure 37).

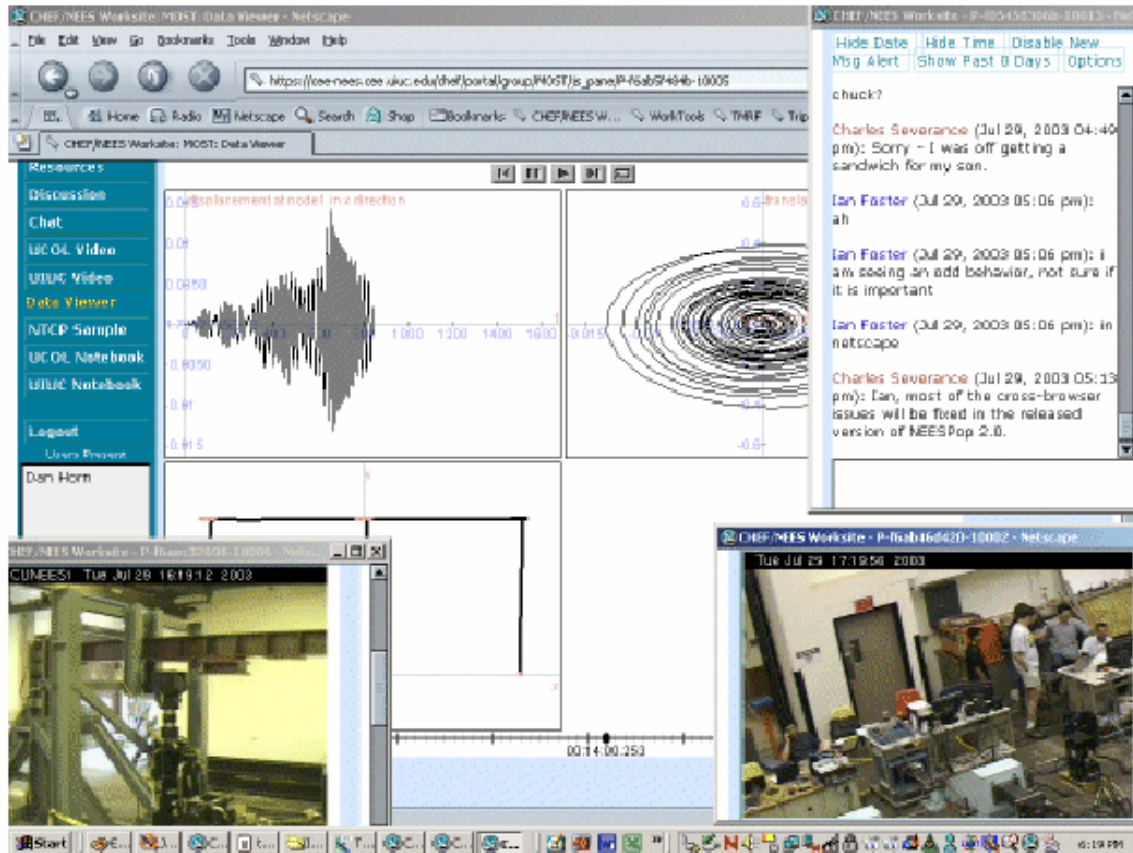



Figure 37: The "floating tool" feature allows users, for example, to view data and streaming video while chatting with other participants.

Multiple tools may be floated at one time, moving and resizing them to create efficient views. Users may “dock” a tool by clicking the button in the top right corner of the tool ().

Information about CHEF usage during the experiment can be found in Section 5.3.

5 Community participation

5.1 Public relations before, during, after experiment

During, before, and after the experiment, the Public Affairs Department at NCSA provided valuable support, advice, and assistance with publicity for the MOST experiment in several ways: creating a postcard to inform the larger earthquake engineering community through the mail, creating a web site as an entry point for the experiment, and handling publicity and the mainstream media during the experiment itself.

5.1.1 Postcards

The NCSA Public Affairs Department produced a postcard that was mailed out to about 2200 members of the earthquake engineering community to alert them to the upcoming experiment. Recipients included members of Earthquake Engineering Research Institute (EERI) and NEES Consortium. The intention behind this postcard was to pique the interest of potential viewers of the experiment, as well as to communicate sufficient information about how the experiment would demonstrate the features of NEESgrid for conducting scientific research in ways that engineers themselves might find potentially useful for their own work.

5.1.2 ES-VTC presentations

In mid-June, the MOST team initiated a series of presentations about the upcoming test as part of the weekly NEES ES Technical Forum. The objective was to inform the NEES community about the technical details of the MOST test and the NEESgrid components, and to increase awareness. After an introductory overview by Bill Spencer and Carl Kesselman on June 12 and a presentation by Benson Shing on Pseudodynamic Algorithms, each following presentation focused on one specific set of NEESgrid services and capabilities offered as part of the MOST experiment, such as telepresence management, the CHEF user interface, or the data and metadata repositories.

During the ES-VTC meetings, the presentations could be viewed via Placeware presentation software; afterwards, the presentations were archived on the MOST website. A complete schedule of all MOST presentations may be found at in Appendix A. The PowerPoint files are archived on the MOST website (<http://www.neesgrid.org/most/index5.html>.)

5.1.3 Press

MOST received considerable online, broadcast, and print media coverage both during and after the experiment. In addition to NCSA press releases issued before and after the experiment, MOST was featured in an article in the Champaign News-Gazette and in a newsfeed from the Associated Press. Interviews with MOST staff and members of the SI team were also aired on WILL-FM, the UIUC-run public radio station, and three local television stations: WICD-TV (CBS affiliate), WAND-TV (ABC affiliate), and WCIA-TV (NBC affiliate). The Public Affairs Division's Public Information Specialist coordinated all media interviews and tracked media coverage.

5.2 Website and multimedia

The MOST website (<http://www.neesgrid.org/most>) integrated text and multimedia to provided visitors and participants with an entryway and introduction to the experiment. In the months leading up to the experiment, the website functioned as a clearinghouse for information and documents pertaining to the experiment, such as details on the experimental setups at UIUC and CU-Boulder and the NEESgrid features being developed by the SI team. The site (Figure 38) was updated every couple of days with relevant news items, including the PowerPoint presentations from the cycle of ES Technical Forums dedicated to the MOST experiment.

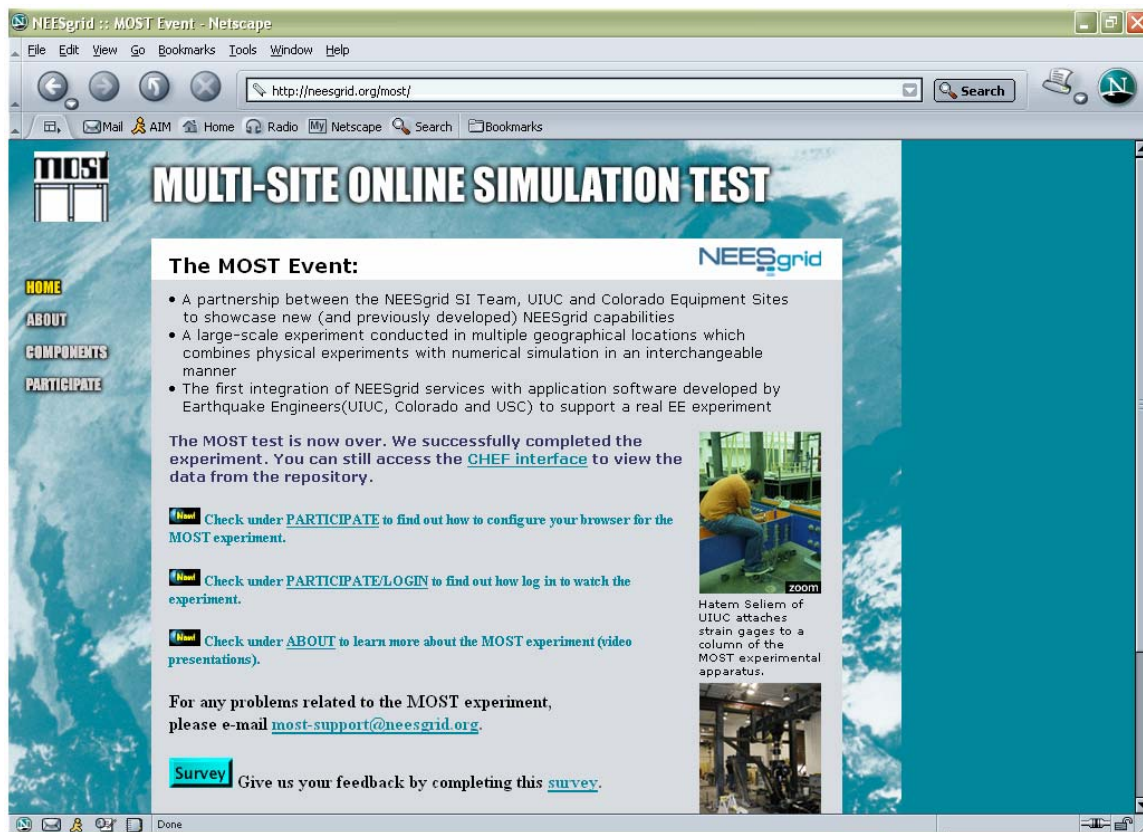


Figure 38: The MOST website, <http://www.neesgrid.org/most>

5.2.1 Website Usage During the MOST Event

5.2.1.1 Usage Statistics for NEESgrid.org

During the month of July, there was an average of 227 daily visitors (6,248 daily hits) on the NEESgrid website. On July 30, 2003, the day of the MOST event, there were 505 visitors (45,061 hits) on the NEESgrid website.

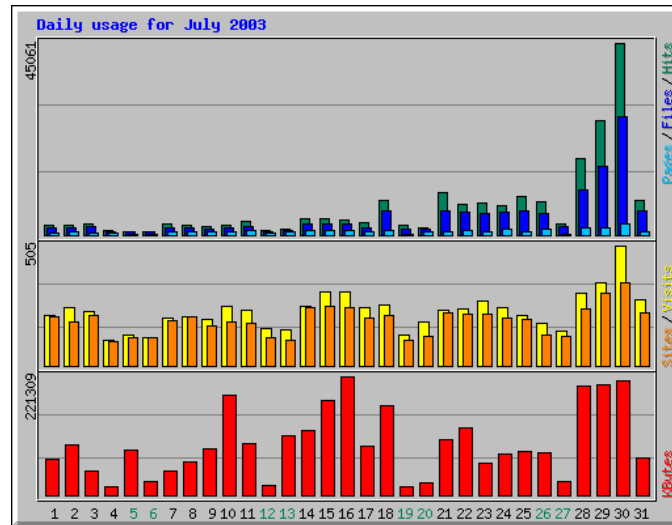


Figure 39 - Daily usage on the NEESgrid website for July 2003

5.2.1.2 Usage Statistics for MOST Section of NEESgrid.org

During the month of July, there was an average of 49 daily visitors (4,212 daily hits) on the MOST section of the NEESgrid website. On July 30, 2003, the day of the MOST event, there were 331 visitors (41,208 hits) on the MOST section of the NEESgrid website.

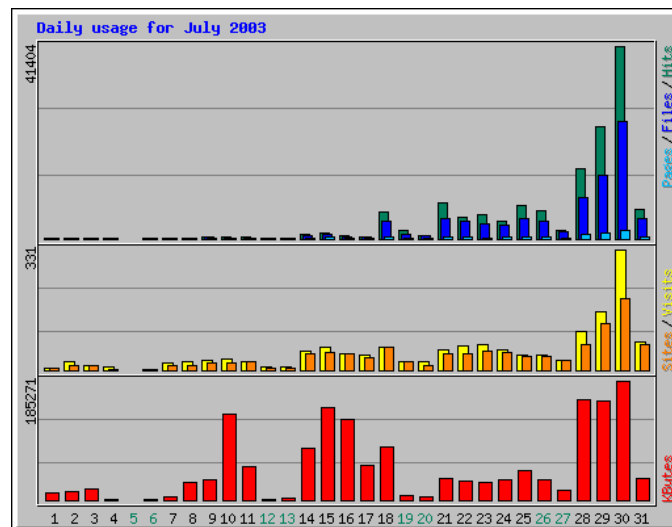


Figure 40 - Daily usage on the MOST section of the website for July 2003

NOTE: The daily averages for the MOST site can be misleading. It is obvious from looking at the graphs that the normal daily average of hits was approximately 292 before visitors started visiting the site in preparation for the experiment. The added activity started up on July 18th. At that point the daily average of hits jumped to about 5,469 until the day of the experiment.

The daily number of web hits on the MOST section of the NEESgrid website accounted for approximately 10% of the total web hits on the NEESgrid website during the month of July. On July 30 the daily number of web hits on the MOST section of the NEESgrid website accounted for approximately 91% of the total web hits on the NEESgrid website.

5.2.1.3 Where Did They Come From?

Visitors to the MOST section of the NEESgrid website represented 26 unique internet domain extensions and countries. While the majority of visitors were represented by US domains, a handful of visitors from 19 foreign countries visited the MOST website. Here is a summary of the top places visitors came from:

64%	US Educational (.edu)
8%	Network (.net)
7%	US Government (.gov)
6%	US Commercial (.com)
1%	Greece
1%	Non-Profit (.org)
1%	US Military (.mil)
1%	Brazil
3%	Other
8%	Unknown

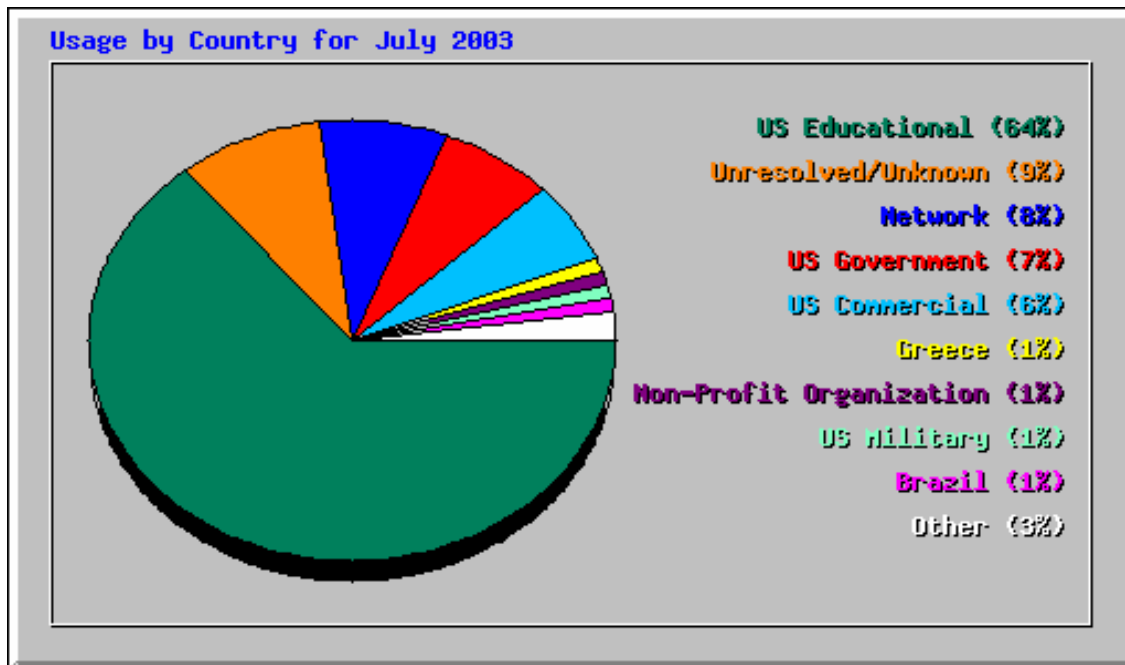


Figure 41 - Usage of MOST section of website by country for July 2003

5.2.1.4 After the MOST Event

While the majority of the website activity occurred during the days leading up to the MOST event and during the event, a steady stream of activity continues to trail off through the month of August 2003. During the month of August, there was an average of 28 daily visitors (1,443 daily hits) on the MOST section of the NEESgrid website. Figure 42 depicts daily usage during August 2003 can get a better idea of the way the activity trailed off by looking at the following daily usage graph from August 2003:

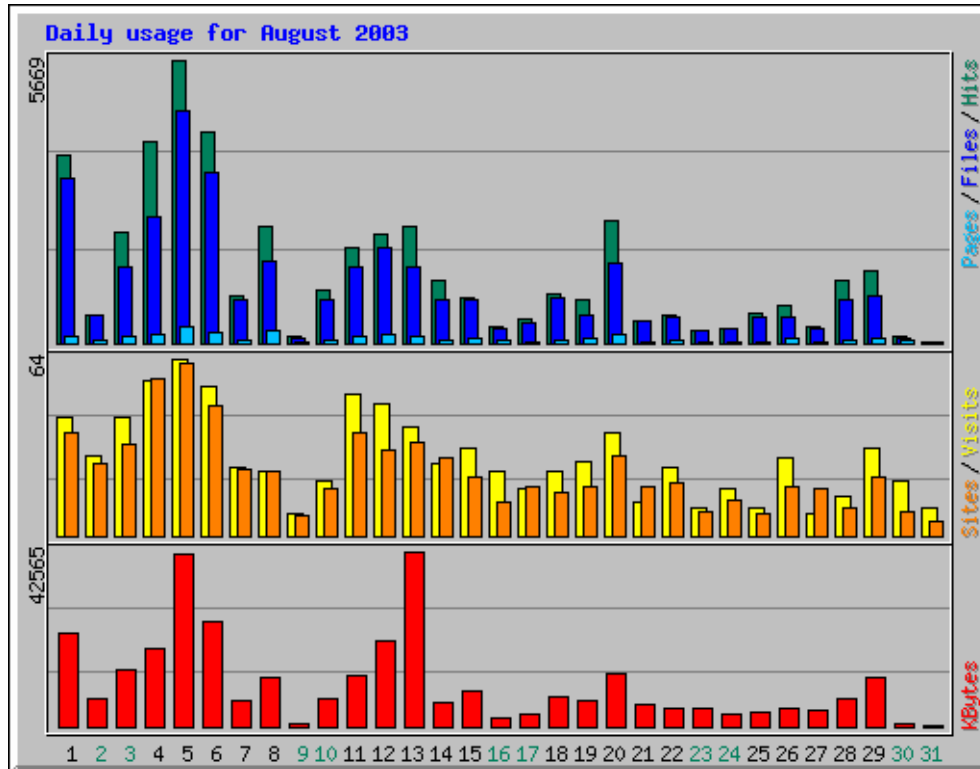


Figure 42 - Daily usage on the MOST section of the website for August 2003

5.3 CHEF usage

Tom Finholt and Dan Horn of the University of Michigan performed a series of analyses of CHEF usage based on usage logs. These analyses were focused on understanding patterns of use of the CHEF collaborative tools, such as who the remote participants were, when they participated, and how they collaborated through the CHEF chat tool.

Approximately 130 people logged on to observe the MOST experiment on the test day. Figure 43 shows the pattern of active participants over time. It shows a peak of participation on the main CHEF server at UIUC (65 users) in the first couple of hours of the test, followed by a gradual decline over the course of the experiment. The figure also shows the pattern of users observing the live data stream from the Colorado CHEF server. These data were generated by calculating the number of unique computers connected to the CHEF server during each 15-minute period during the experiment. This measurement actually slightly underestimates the number of observers, as participants at some sites shared computers.

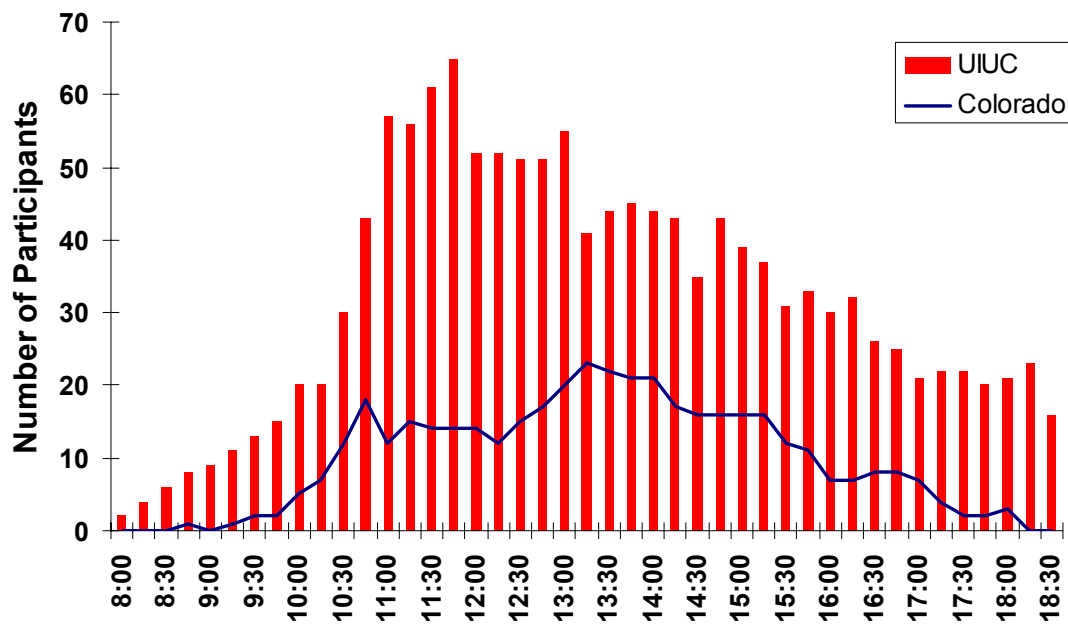


Figure 43: MOST participation over time

Table 1 shows the number and types of users logged in from each NEES equipment site. Again, some of these numbers are slight underestimates due to computer sharing. It should also be mentioned that the MOST experiment took place during a busy time, and low participation rates for some sites should not be seen as reflecting a lack of interest.

Site	Role		Other	Total
	PI	IT staff		
Illinois	1	3	5	9
Oregon State	1	4		5
Minnesota	1	2	1	4
Reno	1	2	1	4
Berkeley	1	--	2	3
Buffalo	--	1	1	2
Davis	2	--	--	2
BYU/UCSB/USC1		1	--	2
Cornell	--	1	1	2
Colorado	1	1	--	2
Lehigh	--	1	--	1
UCSD	--	--	1	1
RPI	--	1	--	1
UCLA	--	1	--	1
Texas	--	--	--	0
Total	9	18	12	39

Table 1: Users from NEES equipment sites

Chat was especially crucial to user interaction during the experiment. It allowed SI team members to interact with each other and, more importantly, from the user's perspective, to describe what was going on at any given time and field questions and listen to user concerns and responses. It also provided continuity throughout the several-hour experiment. Figure 44 shows the number of chat messages that were sent in any given quarter hour. The use of chat follows a pattern similar to that of total users (Figure 43). The key difference is a peak at the end of the experiment. This final peak reflects an increase in "housekeeping" messages that summarize the day's experiences and focus on future steps.

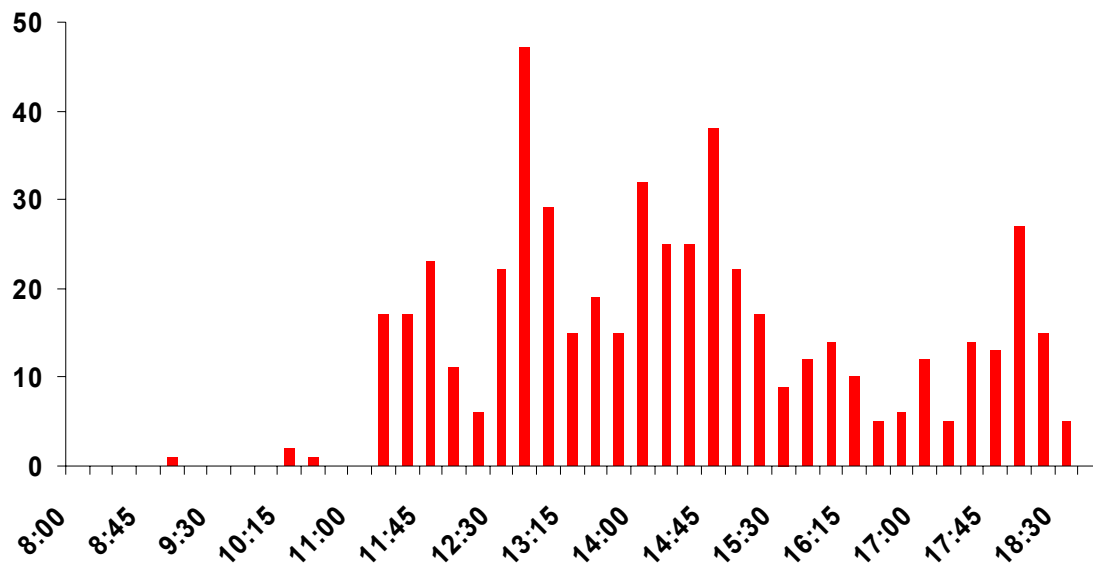


Figure 44: Chat messages over time

Tom Finholt and Dan Horn are continuing their analyses and plan on releasing a report that provides a more complete picture of the use of various CHEF tools by observers over the course of the experiment.

6 Feedback

6.1 *Whole community*

In addition to their analyses of CHEF usage logs, the University of Michigan team collected user comments from a variety of sources. Individuals representing the Equipment Sites and the System Integrator were surveyed before and after the MOST experiment. Additionally, individuals who observed the test were asked to complete a separate survey. These surveys included questions asking respondents to indicate their levels of expectation and satisfaction with the project and its components, as well as to provide feedback about the strengths and weaknesses of the system. In addition to the surveys, a number of participants provided feedback via the chat tool during the experiment, and via email after the experiment. These comments were aggregated and classified. The user comments from the MOST user community as a whole were largely positive. However, there were a number of concerns about bugs in the interface (which have been passed on to the CHEF team), and some challenges in reducing disorientation among some observers who were less familiar with the MOST project.

The MOST team also came away from the experiment with the following recommendations:

- Users need to be made aware of all the different discussion media available, and the differences between them. Some users were unaware that there was a chat feature or a discussion forum.
- Developers need to be aware that users may be unfamiliar with some common electronic communication media, such as interactive chat.
- Explicit orientation for users is essential. In particular, interactive tips and audio/visual input would help users get a much better sense of what's going on than flat text explanations.
- Documentation of experimental setup and NEESgrid features should be available and accessible for perusal.
- User documentation must be available in multiple modes and places, and not "buried" in the website.
- In the absence of other communication media, chat must be easy to use and reliable.

It's important to have a "voice of Mission Control" to orient users to what's going on and give the events some continuity, because people visit and leave throughout the several-hour duration of the experiment.

6.2 *MOST team members*

Some general lessons that the team took from the experiment included the importance of realistic simulations for testing basic algorithms and equipment, both hardware and software; the importance of good communication, and, above all, the importance of good planning.

Various simulations modes were tested extensively using the MATLAB programs prepared for the MOST experiment. These simulations progressively approached the actual testing conditions and worked extremely well in conjunction with the control and testing simulations that have been developed and carried-out at CU. For future experiments, it would be useful to implement more comprehensive tests for the data streaming and display aspects.

With respect to communication issues, the video and teleconferences provided an excellent opportunity for the geographically distributed teams to work together. Also the email groups provided a convenient means of communication. A more structured "mission control" mechanism as proposed by the SI team for future NEES experiments will go a long way to improve the coordination between teams.

Finally, good planning requires a balance between firm deadlines and adherence to schedule on the one hand and flexibility and creativity on the other. However, it cannot be emphasized enough that good planning is essential to ensuring that safety and good

judgment prevail in carrying out simulations using sophisticated and powerful high-performance hydraulic actuators. And it is important to prepare for every contingency: if there is a possibility that something could go wrong, it very likely will.

Additionally, the SI team identified the following important lessons:

1. **One of the major accomplishments of the MOST experiment was to demonstrate several new capabilities in the area of telecontrol.** The NTCP protocol was devised and implemented and tested in the real-life MOST experiment.

The use of NTCP in this real experiment led to several important changes and revisions in the NTCP protocol. These included adding requests for initiating and terminating an experiment. The addition of OpenSession and CloseSession was the first step in this process. In the future, a request will be created that sets the state of components, either physical or simulated, to one of several pre-determined states (e. g., initialized, shutdown, etc).

In addition, the exercise of writing an NTCP plugin for a hydraulic control system led to the formation of some design principles for this kind of control plugins.

The MOST experiment demonstrates that the Control protocol works as planned and is useful to a "real experiment" in the Earthquake Engineering field. As a result, new experiments will be devised to test the control protocol to work with faster (near real-time) dynamic requirements.

2. **Fault-tolerance is important.** Some of the fault-tolerance features implemented came into play during the MOST run. For example, careful design of sub-components helped address many of the points of failure, loss of a request, or network hiccup. Implementation of the client library's connection retries helped let us carry out the experiment almost to the end of the planned 1500 steps. The prior run of a 1500 step (distributed) simulation ran flawlessly.

Another important lesson learned was that it is important that all telecontrol sub-components (e.g., client applications, plugins, backend modules etc.) should follow the same principles for fault-tolerance as NTCP; this practice is essential to the success of the experiment. Future experiments will include specific fault-tolerant enhancements.

The need for fault tolerance was also made clear in the performance of the data ingester. During the experiment, the ingester was unable both to unblock its clients and ensure no data was lost under the condition that NFMS repeatedly refused to upload a particular file (because of the concurrency bug). There were two solutions to this problem; the one used for MOST was for data ingester clients to write sequences of files, rather than one file at a time. The solution for clients that write one file at a time is to provide a temporary staging area in which

data can be stored when NFMS fails; that should have been provided for MOST but was not. Under all other conditions, the data ingester worked perfectly; it tolerated all other component failures without stopping or losing data. Those failures include NFMS and NMDS outages. The lesson from all of this is of a general nature: that fault tolerance for live experiments requires 1) providing redundant mechanisms for caching data during failures and 2) eliminating fail fast conditions from clients.

Faults were encountered during the MOST experiment that will be handled by fault-tolerance features that have been planned but not yet implemented.

3. **Video of work areas should focus more specifically on the experiment.** The presence of telepresence cameras gave a sense of the activity of the work area and the people involved; however, in future experiments telepresence will provide a more detailed view of the experiment underway. In the weeks leading up to the day of the experiment, the camera at Newmark Lab showed activity in a work area (that is, video of people, not the experiment specimen). On the day of the experiment, video from this camera gave a compelling picture of the level of activity in the lab. However, video feed of sleep-deprived people toiling away in the nights before the experiment was not particularly useful.
4. **Design principles need to be communicated better.** Client applications, plugins, and backends should follow the same principles for fault-tolerance that NTCIP does. Otherwise, those people designing and implementing these components need to be informed of the possible consequences of not building fault-tolerance into the design.

For example, although NFMS otherwise performed as designed, it failed several times during the experiment because of a bug in its handling of concurrent access by multiple clients. This caused a number of other components to hang or fail, including the data ingester, which responded to the failure mode by entering an infinite loop attempting to retry the failed operation. The NFMS bug was fixed, and the lesson is to coordinate error-handling strategies among components in order to meet all the design goals. In this case, the aggregate behavior of the components was for the entire simulation to have to be restarted, whereas it should have just paused while the failing components were restarted.

5. **All sites must meet NEES specifications to ensure satisfactory performance.** All TPM components performed to ANL specifications; no components or tools failed. For example, NTP must be implemented on all video servers during the installation. During the MOST experiment, because not all sites were yet familiar with NTP, it was not yet implemented on all video servers during the MOST experiment; resulting in out-of-sync clocks until the servers were updated.

Likewise, it is essential for participating equipment sites to have a dedicated network; in some cases, the lack of one caused some sites to lose streaming

videofeeds from UIUC. However, the TPM tools did not fail when the network became overloaded.

Other more specific observations included the importance of checking boundary conditions, calibrating sensors carefully, and ensuring that software supports Internet Explorer fully. Regarding specific components:

- NMDS performed nearly flawlessly during MOST. Prior to MOST, NFMS was performing NMDS operations that ran slowly after many updates. This turned out to be a problem with NFMS, which was fixed before the experiment. Thus, it is important to document the kinds of use scenarios that are likely to be slow when they involve the creation of many object versions.
- Informal performance analysis of NFMS shows that its performance is almost entirely bounded by the performance of NMDS and GridFTP. It is apparent that NFMS transaction takes at least >1s in ideal conditions.
- The CHEF repository browser performed adequately for MOST. The most significant problem was a bug in which it was not possible to delete certain objects. That bug has been fixed for 2.0. What this suggests is that a documented testing script is needed

Finally, the MOST experiment underlined two important lessons: that developing standardized tools and resources allows the Earthquake Engineer to focus on the engineering problem at hand, and that having common tools leverages resources and experiences. While a number of potential improvements have been identified, it was clear that the basic system worked for the MOST experiment.

7 Future directions

The MOST experiment was an example of *Experiment-based Development* that demonstrated all the major functionality and services that have been provided by the SI team in the recent release of NEESgrid 2.0. With the inclusion of the latest services NEESgrid 2.0 has evolved to a state where it is useful to many of the ES without any additional development required by the SI team.

The Early Adopter (EA) program has shown that identifying and performing experiments representative of an ES area of interest is the most efficient way to get the site engaged in NEESgrid. In order to engage additional ES on a more aggressive schedule the SI team has proposed the concept of *Experiment-based Deployment* of the NEESgrid software. This activity consists of a close collaboration between the ES and an SI deployment team to perform an ES-defined experiment using the current capabilities of the NEESgrid software distribution. This experiment will provide a central focus and provide a synergistic partnership for deploying the NEESgrid at the ES. More information about Experiment-Based Deployment may be found in the document Experiment-Based

Deployment Strategy and Requirements

(http://www.neesgrid.org/documents/NEESgrid_epd_strat_req_v1.0.doc) on the NEESgrid website.

8 Contributors to the MOST Experiment

The success of the MOST experiment resulted from sustained, intense cooperation and hard work over a period of several months by a large number of contributors from several institutions. All those involved are here listed alphabetically by institution.

University of Southern California

Information Science Institute:

Mike D'Arcy
Nabil Deeb
Sridhar Gullapalli

Carl Kesselman*
Gaurang Mehta
Laura Pearlman

Department of Civil and Environmental Engineering:

Erik Johnson

Argonne National Laboratory

Paul Hubbard
Peter Lane
Lee Liming

Ravi Madduri
Pawel Plaszczak
Nestor Zaluzec

University of Michigan

Sung Joo Bae
Thomas Finholt*
Glenn Golden
Dan Horn

Nancy Moussa
Lars Schumann
Charles Severance

University of Colorado at Boulder (CU-Boulder)

Department of Civil and Environmental Engineering

Tom Bowen
Rae-young Jung
Benson Shing**

Erik Stauffer
Robb Wallen
Zhong Wei

University of Illinois at Urbana-Champaign (UIUC)

Department of Civil and Environmental Engineering

Dan Abrams
Liang Chen
Amr Elnashai **
Young Suk Kim
Dan Kuchma *

Narutoshi Nakata
Hatem Seliem
Bill Spencer **
Carrie Wagener
Guangqiang Yang

National Center for Supercomputing Applications (NCSA)

Kazi Anwar
Trish Barker
Cristina Beldica
Miranda Callahan
Joe Futrelle
Jeff Gaynor
David Gehrig

Sang-Chul Lee
Doru Marcusiu
Chase Phillips
Joel Plutchak
Kathleen Ricker
Vanitha Varadarajan

* Indicates Co-Principal Investigator

** Indicates Principal Investigator

Appendix A: List of ES-VTC Presentations

6-12-03: Carl Kesselman, University of Southern California/Information Sciences Institute, and B. F. Spencer, University of Illinois at Urbana-Champaign:

"The July MOST Experiment: Multi-site Online Simulation Testbed Experiment"

<http://www.neesgrid.org/most/docs/estf0612.zip>

6-19-03: P. Benson Shing, University of Colorado at Boulder:

"Pseudo-Dynamic Algorithms"

<http://www.neesgrid.org/most/docs/ESTF0619.ppt>

7-3-03: Laura Pearlman, Information Science Institute, and Erik Johnson, University of Southern California:

"NTCP: NEESgrid TeleOperation Control Protocol"

<http://www.neesgrid.org/most/docs/ESTF0703.ppt>

7-10-03: Joe Futrelle, National Center for Supercomputing Applications, and

Amr Elnashai, Hatem Seliem, and Young Suk Kim, University of Illinois at Urbana-Champaign:

"NEES Metadata Services"

<http://www.neesgrid.org/most/docs/ESTF0703.ppt>

7-17-03: Nestor Zaluzec, Argonne National Laboratory, and Dan Kuchma, University of Illinois at Urbana-Champaign:

"Telepresence Manager (TPM) Services and the E-Notebook"

<http://www.neesgrid.org/most/docs/ESTF0709.ppt>

8-5-03: Carl Kesselman, University of Southern California/Information Sciences Institute, and

B. F. Spencer, University of Illinois at Urbana-Champaign:

"Lessons Learned from the MOST Event"

<http://neestpm.mcs.anl.gov/NEESDocs/Zaluzec-ESFT-20030717.pdf>

8-21-03: Thomas A. Finholt, Dan Horn, and Sung Joo Bae, School of Information, University of Michigan:

"Analysis of the user experience during the July 30, 2003 MOST experiment"

http://www.neesgrid.org/most/docs/030821%20MOST_user_experience_analysis_FINAL_08_21_03.ppt

8-28-03: Paul Hubbard, Argonne National Laboratory, and Liang Chen, University of Illinois at Urbana-Champaign:

"DAQ and NSDS for MOST"

<http://www.neesgrid.org/most/docs/DAQ%20and%20NSDS%20for%20MOST1.ppt>