13th World Conference on Earthquake Engineering
Vancouver, B.C., Canada
August 1-6, 2004
Paper No. 583

DEVELOPMENT OF A PROTOTYPE WEB-BASED SFSI SIMULATION TOOL

Robert NIGBOR[1], Ali ASGHARI[2], Larry McMICHAEL[3], Kyran MISH[4]

SUMMARY

Coupled to experiment design for a NEES-sponsored test facility in Southern California, a prototype simulation tool for soil-foundation-structure-interaction analysis has been developed which provides a web-based, interactive, user-friendly, finite element analysis framework for the purpose of remote computation and visualization of finite element results of a fully instrumented test structure. A structured database has been deployed to facilitate data manipulation through which most of the transactions in terms of storage and modeling are meticulously managed. The development of the prototype simulation tool is based upon the Model-View-Controller design pattern. The project is based on an open-source, platform independent philosophy that makes the adaptation and distribution of the tools developed possible. The simulation applet can be accessed at http://nees.usc.edu.

INTRODUCTION

The integration of the web with the field of simulation should provide lots of ideas for how simulation is to change. With the web being the driver of much of today's software technologies, one needs to study how neatly it is feasible to embed simulation interface in browsers while the main computation is remotely processed remotely on another machine(s). Handling different users simultaneously, efficiently input/output manipulation, handy graphical interface design, soft-wares compatibilities, platform independency to allow the program to be executed flawlessly on common commercial operating systems, developing a simply expandable structure and so many other issues should be investigated at the primarily stage of such multidisciplinary

[1] Research Associate Professor, University of Southern California. Email:nigbor@usc.edu
[2] Graduate Student, University of Southern California, USA. Email:asghari@usc.edu
[3] Post Doctorate, University of Oklahoma, USA.
[4] Presidential Professor of Structural Engineering, University of Oklahoma, USA.

projects to get the best outcome possible. Considering all the above tips as mostly software/hardware concerns, understanding the physics of the problem itself is of great importance that is not to be overlooked by the complexity of the computer methodologies. The objective of this paper, however, is to present a prototype web-based simulation tool that has been developed to assist in the design of soil-foundation-structure interaction (SFSI) experiments at the Garner Valley Downhole Array (GVDA) test facility (Fig.1) in Southern California. This web-based application is to handle the finite element analyses of a predetermined soil-structure-interaction model on a server and to display the results to the remote user through a web browser. At the back end, there are various web-oriented methodologies incorporated in this process to control the integrity of the analysis whilst at the front end the user interface allows users to simply specify changes to the configuration of the test structure in order to predict the influence of variations in the mass, stiffness, and bracing patterns on the interactions experienced between the test structure, foundation, and the soil. Results from these analyses will be effectively stored in a database to be investigated by whom the data is available for to determine appropriate structural configurations and load conditions for experiments to be conducted at the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) sponsored GVDA test facility. Although the primary purpose of the simulation tool is to provide users a simple interface to perform finite element analyses of the GVDA test structure, the technology developed for the project has broader applications as an open-source framework for the remote computation and visualization of finite element results.

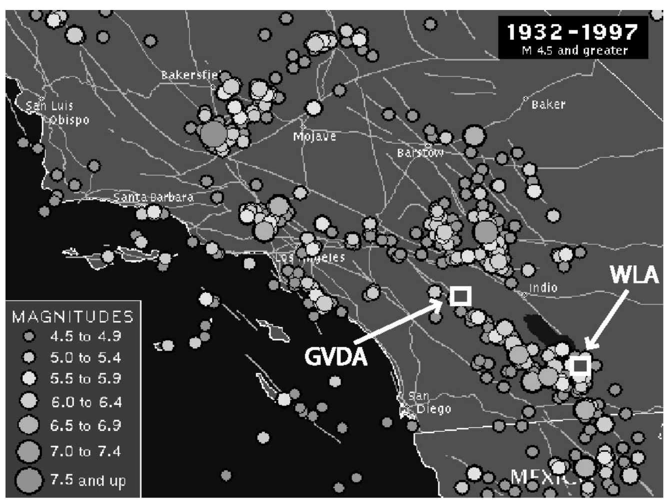

Figure. 1: Map of southern California showing locations of WLA and GVDA and epicenters of earthquakes with magnitudes greater than 4.5 between 1932 and 1997.

Though each component of the simulation tool may be the product of a different development environment, the compatibility of each component which interacts (either directly or indirectly) with the other components must be practically verified; nevertheless having bugs in such a multi-layer program seems to be inevitable. An open development environment has been utilized for the prototype simulation tool in order to promote both the future extension of the prototype tool's capabilities and the ability for other NEES projects to use and deploy the tools developed. The development environment consists of a variety of open-source and freely distributed tools, which can be deployed on a variety of computer platforms (e.g. Windows, UNIX, Linux). The development of the prototype simulation tool was based upon an object-oriented methodology using the Model-View-Controller (MVC) pattern.

BACKGROUND

The state-of-practice for engineering characterization of Soil-foundation-structure interaction effects for routine structures has not undergone recent advancement, though the state-of-the-art in SFSI analyses has evolved steadily over the last three decades. Despite the fact that in early 30s many analytical approaches initiated to consider the effect of soil interaction on the behavior of the buildings but due to complexity of the solutions and the lack of advanced computational devices most of the efforts by many scientist such as Sezawa and Kanai (1935) Biot (1932) and etc. just were limited on papers and never influenced the real world problem till late 70s that more realistic approaches came out and mainly by assuming that the foundation is rigid to simplify the problem; of course even by now there is just few papers the essentially investigate flexible foundation.

Nevertheless some simplified substructure-based SFSI provisions are included in the NEHRP(BSSC, 1997) and Applied Technology Council (ATC, 1978) codes, but these provisions have not been calibrated against field performance data. Due to the common perception that to ignore SFSI effect is conservative (which is a naive justification fueled by the lack of field performance data), the SFSI provisions in the NEHRP and ATC codes are voluntary and are often neglected in practice.

The extend to which soil-foundation-structure interaction alters the response of a building depends mainly on the dynamic stiffness of the soil relative to the structure. In a more general sense, soil-foundation-structure interaction is a collection of phenomena in the response of structures caused by the flexibility of the foundation soils, as well as the response of soils caused by the presence of the structure and it has long been recognized as an important factor, though the physics governing SFSI are not well understood yet. To help address this problem, an instrumented test structure will be constructed at the GVDA test facility to study the physics of SFSI under controlled field settings. The test structure (Fig.3) is composed of a superstructure featuring a thick, reinforced concrete, roof slab supported by a steel, moment resisting, frame and a foundation consisting of a thick, reinforced concrete, floor slab (Figure 1). The steel frame has a configurable bracing system that can be used to modify the stiffness and damping characteristics of the overall structure.

The roof slab has been designed to support additional mass, which can be used to modify the period of the structure. Active loadings can be applied to the structure by either a small shaker installed on the underside of the roof slab or by an external shaker connected to the structure via a mounting bracket on the roof slab.

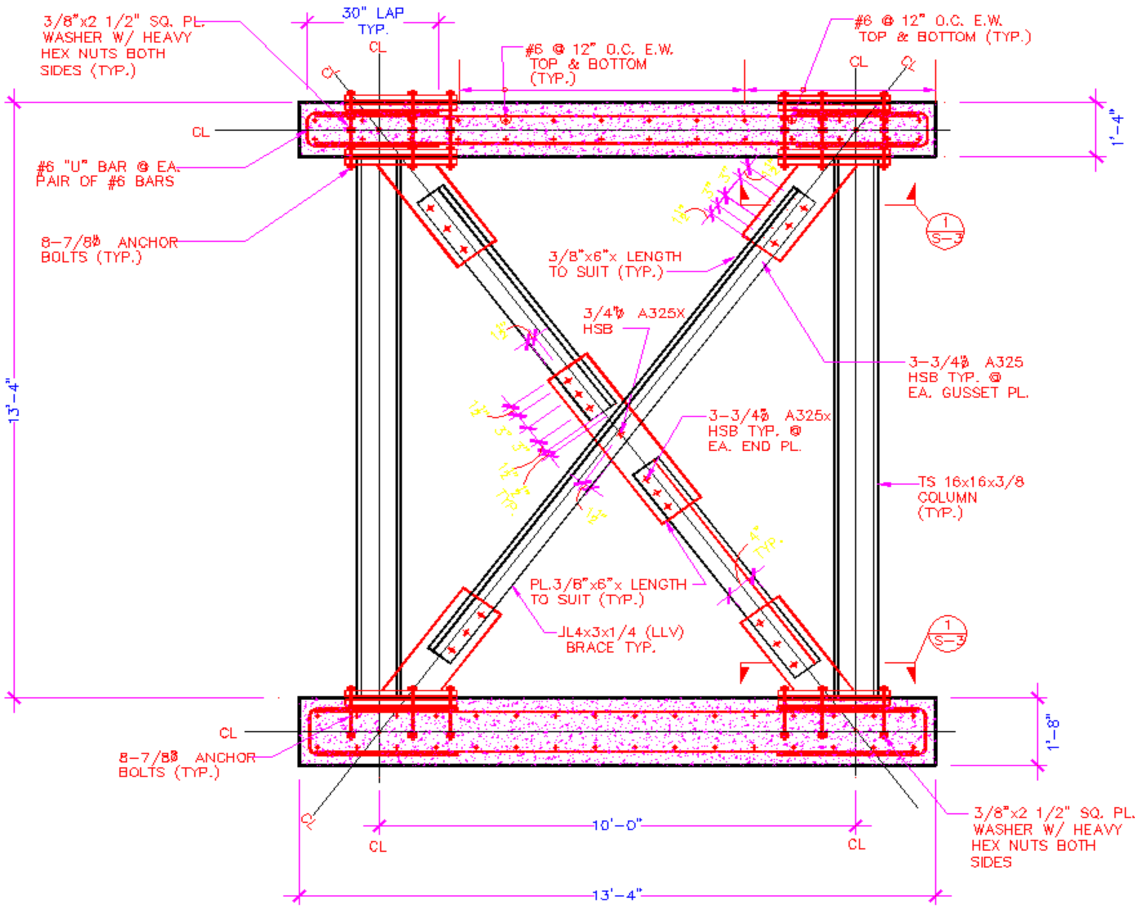

Figure 3. Instrumented test structure at the NEES-sponsored GVDA test facility.

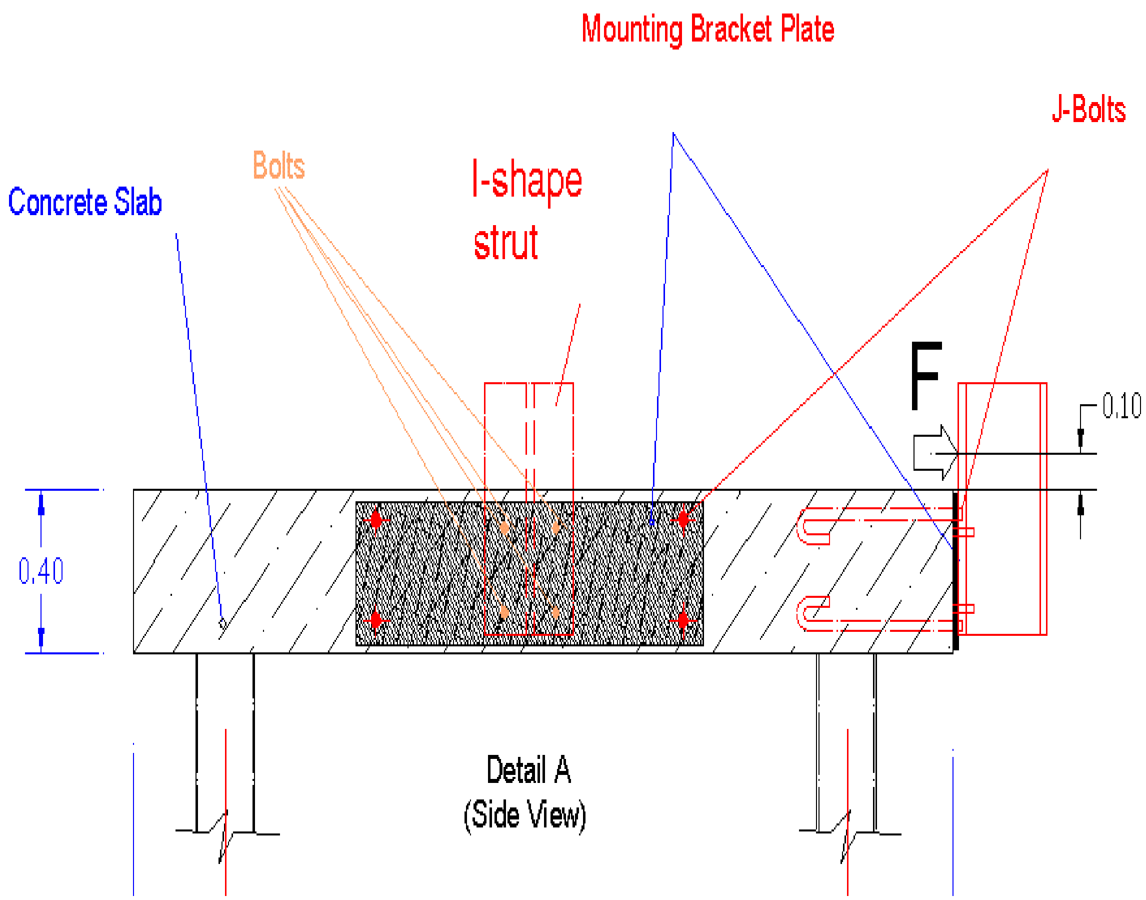

Figure 4. Mounting brackets at the roof of the test structure at GVDA test facility.

Additionally, the test structure will subject to passive loadings resulting from earthquake ground motions. The GVDA site is also equipped with both surface and downhole arrays of accelerometers (figure 5). It will be instrumented with accelerometers, strain gauges, pressure cells, and uplift displacement monitors that will be connected to a real-time data acquisition system as illustrated in figure 6. The test structure at the GVDA test facility is scheduled to be operational in mid-2004.

The prototype simulation tool will be used to predict the response of the test structure to various structural configurations and applied loads. The numerical results can then be stored and retrieved through a database management system (DBMS) package such as MySQL which is an open-source DBMS software, structured upon relational database. Studying the results may lead to deciding which configuration and loading will induce the desired behavior in the structure during an experiment.

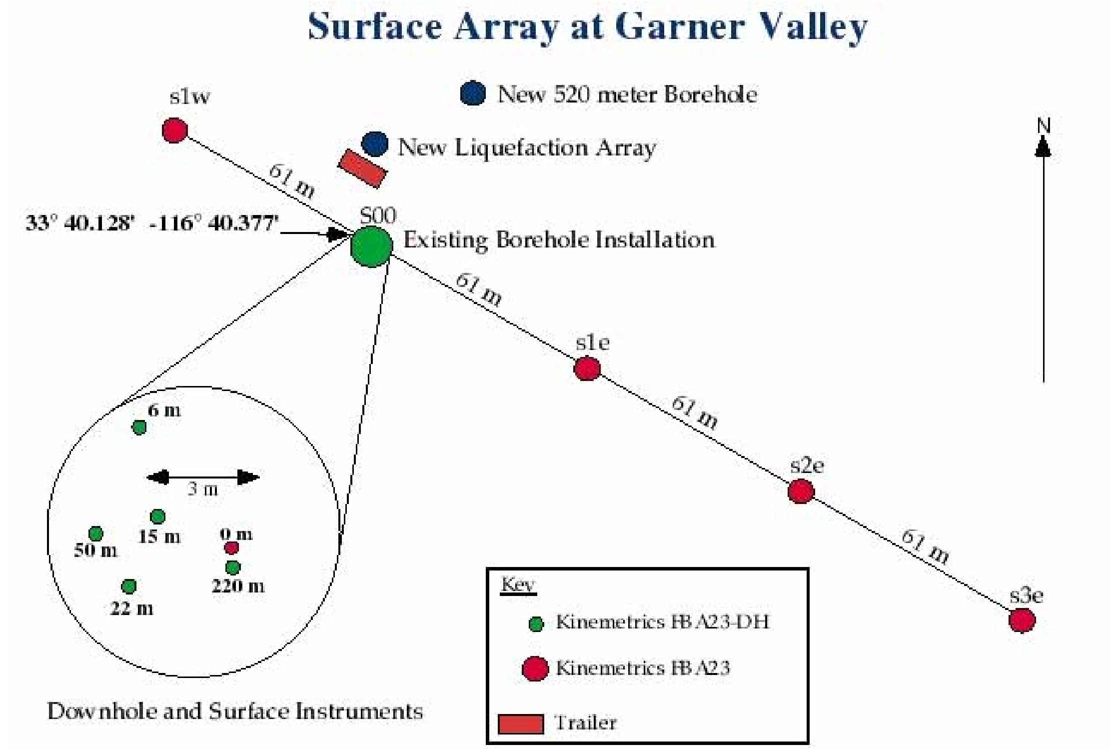

Figure 5. Map view of the GVDA test site showing the relative location of the surface stations, liquefaction array and boreholes.

The most important design aspect of the prototype simulation tool is the close coupling of a web-based, interactive, user-friendly, finite element analysis capability with the design of experiments for the instrumented GVDA test facility. However, making the tools developed to achieve that capability as general, flexible, and extensible as possible increases the value of this project to the educational and research communities.

The accessibility provided by a web-based application, along with the maturation of network-enabled development technology (e.g. the Java servlet), is beginning to generate widespread interest in the development and deployment of engineering applications on the Internet. A wide variety of web-based engineering applications have been developed from simple demonstrations of the technology (Flagpole Project, 2002), to web-based interfaces to numerical analysis programs (von Oehsen et al, 2002), to frameworks for the collaborative development of a numerical analysis program (Peng, 2002).

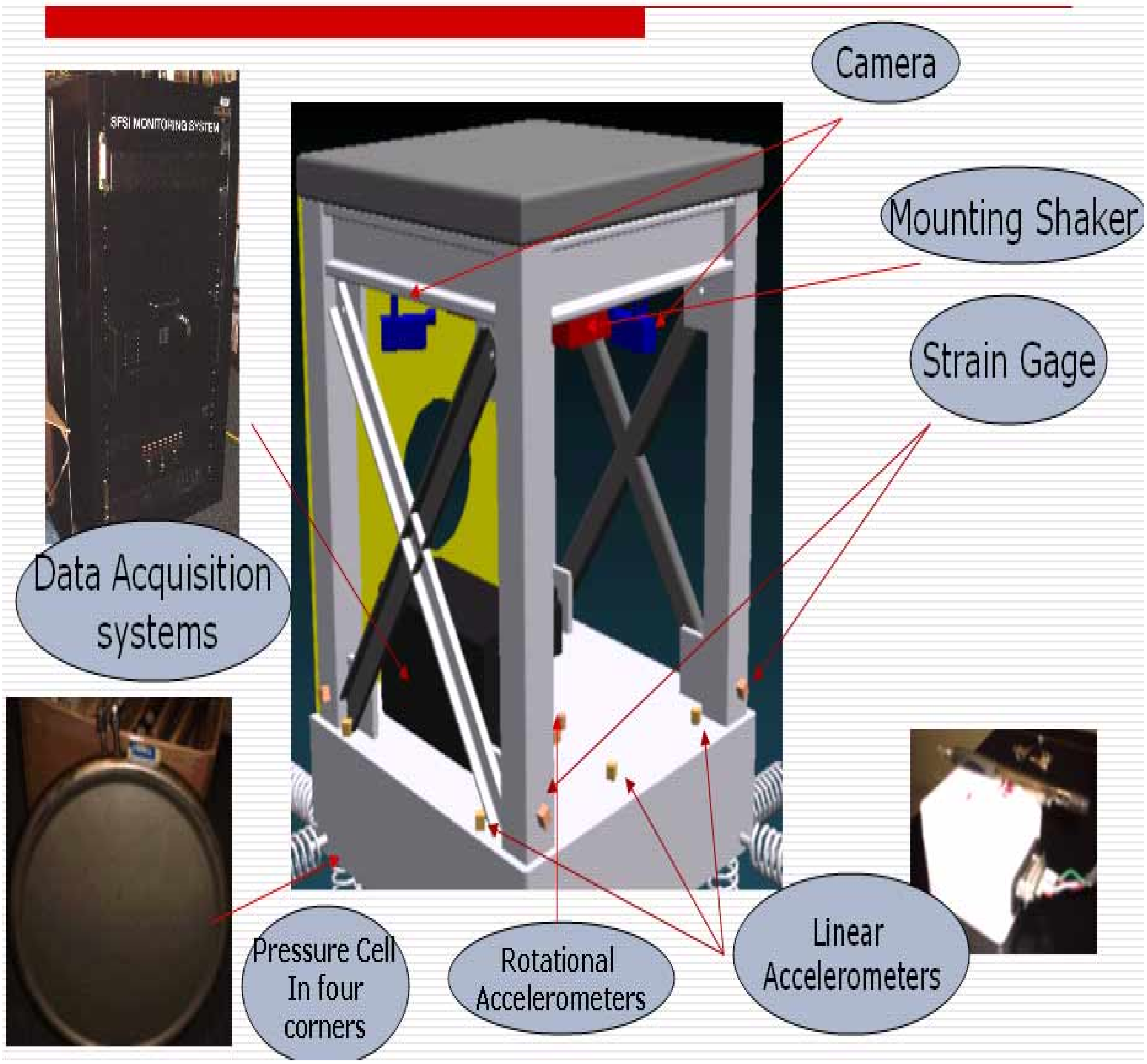

Figure 6. The schema of fully instrumented structure at the GVDA test facility.

OBJECT-OREINTED, OPEN-SOURCE DEVELOPMENT

The entire technology base upon which the simulation tool was developed was rooted in an open-source, object-oriented environment. The Java programming language was chosen as the most appropriate programming language in which to develop the simulation tool. The Java 2 (j2sdk1.4.1_03)Platform (Sun Microsystems, 2003) has become the industry standard for web-based interactions. The Struts framework (Apache Software Foundation, 2003a) was used to manage the communication between the user and the server. Storage of the state (current configuration and results) of each user's model is maintained by a database management system (MySQL, 2003) to neatly manipulate the data. The simulation tool can be deployed using a web server (Apache Software Foundation, 2002) and servlet container (Apache Software Foundation, 2003b) available for both Windows and UNIX based operating systems.

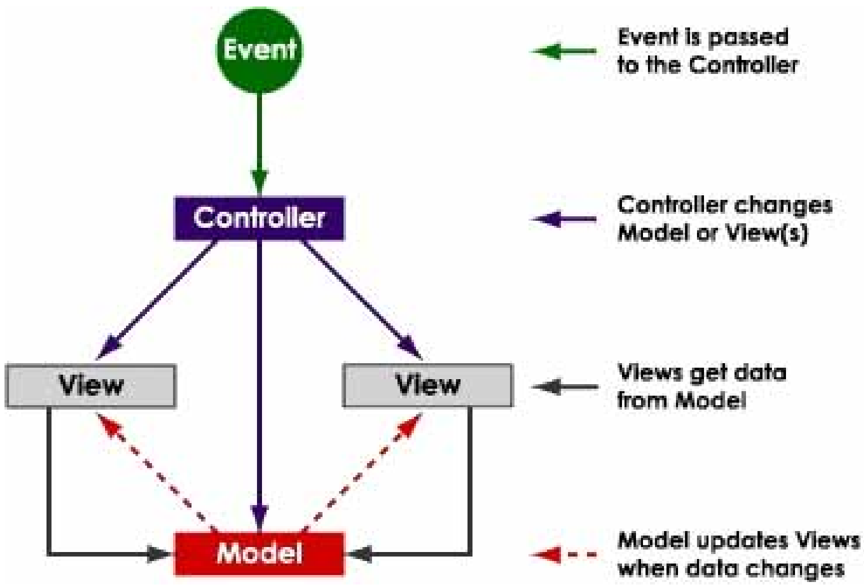

Figure 7. **Schematic of the Model-View-Controller design pattern**

The development of the simulation tool was based upon the Model-View-Controller (MVC) design pattern (Figure 2). In the MVC paradigm the user input, the modeling of the external world, and the visual feedback to the user are explicitly separated and handled by three types of object, each specialized for its task. The view manages the graphical and/or textual output to the portion of the bitmapped display that is allocated to its application. The controller interprets the mouse and keyboard inputs from the user, commanding the model and/or the view to change as appropriate. Finally, the model manages the behavior and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).

The simulation tool can be thought of in general terms as consisting of three major components: a user interface (e.g. java applet or strut), server-side application code (servlet) , and an application binder. The user interface (View pattern) provides the ability for the user to interact with the prototype simulation tool via a web browser. The user interface provides an intuitive and simple means for the user to: define the geometry of the test structure and the load conditions selected from dozens of available database objects; perform the finite element analysis on the server; access the results of the analysis; save the input and output files for future use. The user interface was developed using the Swing components. The Swing components, which are part of the Java Foundation Classes (JFC), can be used with either JDK 1.1 or the Java 2 platform. Also a blend of Hyper Text Markup Language (HTML) and JavaServer Pages (JSPs) is to control the content and layout of the web pages displayed to the user.

Stand-alone vs. Client-Server application

The server-side application code (Model pattern) determines the results that correspond to a user's request. A server-side application is a program or resource that runs on the server, as opposed to a client-side application that runs on the user's computer. The application code is responsible for maintaining information on the current state of the user defined model, finite element analysis, and analysis results for each user; most of these transactions are carried out via several web protocols which are also responsible for integrity and concurrency of the process. A typical stand-alone application will store the state information in a combination of computer memory and files on the hard disk, depending upon how frequently the information is changing, e.g. the geometry of the structure and the applied loads may be stored in computer memory while the model is being developed and then, when the mel is complete, a file containing all the input data necessary for the finite element analysis may be written to the hard disk. However, a web-based application may have several users defining and changing the state of their respective models at the same time. It was therefore decided to use a relational database to store the state information for each user while the model is under development, in order to avoid potentially severe demands on the server's memory and to avoid frequent, relatively slow, writes to a file. The simulation tool will employ several server-side applications, but the primary application code is the finite element analysis engine. The finite element analysis engine selected for the simulation tool is OpenSees (PEER, 2003); OpenSees, the Open System for Earthquake Engineering Simulation, is an open-source development effort coordinated and maintained by the Pacific Earthquake Engineering Research Center (PEER). The current capabilities of OpenSees include a variety of element and material types, linear and nonlinear equation solvers, and time dependent load specifications. OpenSees is developed using the C++ programming language; make files are available for the Windows, Unix, and Linux platforms so there is no platform dependence introduced by the choice of OpenSees.

The application binder (Controller pattern) acts to coordinate information and resources between the server hosting the simulation tool and the user's computer (typically referred to as the client). The application binder is a server-side web application that receives information from the user interface, processes the request, and returns the appropriate information to the user. The application binder must therefore be able to interact with other applications on the server (e.g. execute the finite element analysis) in addition to coordinating communication over the Internet between the client (user) and the server. The Java programming language was chosen as the most appropriate development environment for the application binder. In addition to being a full-featured, object-oriented, programming language, the protocols and interfaces required for network communication (i.e. the Internet) and database interactions are embedded within Java.

REMOTE COMPUTATION AND VISUALIZATION FRAMEWORK

In general terms, the technology developed to implement the simulation tool can be thought of as a framework for the remote computation of finite element analyses and the remote visualization of the results. The user interacts with the simulation tool through a web browser and the Internet. The user defines the configuration of the structure and the loads to be applied using a graphical user interface (GUI). The user then submits a request for the finite element analysis to be performed on the remote server. The server is responsible for the finite element analysis, processing the results, and returning the results to be displayed locally to the user via the web browser. This approach allows the intensive numerical calculations associated with the finite element analysis to be offloaded from the user's computer; the server may perform the calculations itself or it may forward the request for the analysis to another computer, or cluster of computers, depending upon what approach maximizes the computational resources available.

Information flow through the framework can be summarized in the following manner. The user specifies changes to the state of the model (e.g. loads, boundary conditions, bracing elements) through the GUI displayed in the web browser (View). The data supplied by the user is collected via HTML Forms' elements (e.g. input boxes, check boxes, buttons) and passed to the Controller. Based on the user input, the Controller determines to which server-side application (Model) it will pass the information. The server-side application updates the state of the model and notifies the Controller when it has completed its task; the Controller then updates the View so the user can visualize the changes to the model. When the model is complete, the user initiates a request for a finite element analysis by clicking on a button in the web browser; clicking the button causes the Controller to invoke a servlet on the server. The servlet calls a Java program that initiates the finite element analysis (C++ executable) and waits for its completion; once the analysis is complete, the results can be displayed on the user's web browser.

PROTOTYPE SFSI SIMULATION TOOL

The prototype simulation tool (Figure 8) provides a simple interface to perform finite element analyses of the GVDA test structure; the primary purpose of these analyses is to design experiments to be performed at the GVDA test facility in order to develop a better understanding of the interaction that occurs between a shallow, slab, foundation and the soil. The simulation tool also provides users with background information on SFSI and a tutorial on the use of the simulation tool. This information serves not only to assist researchers in learning to use the simulation tool to design experiments, but also serves as an educational resource for students. The prototype simulation tool can be accessed at http://nees.usc.edu.

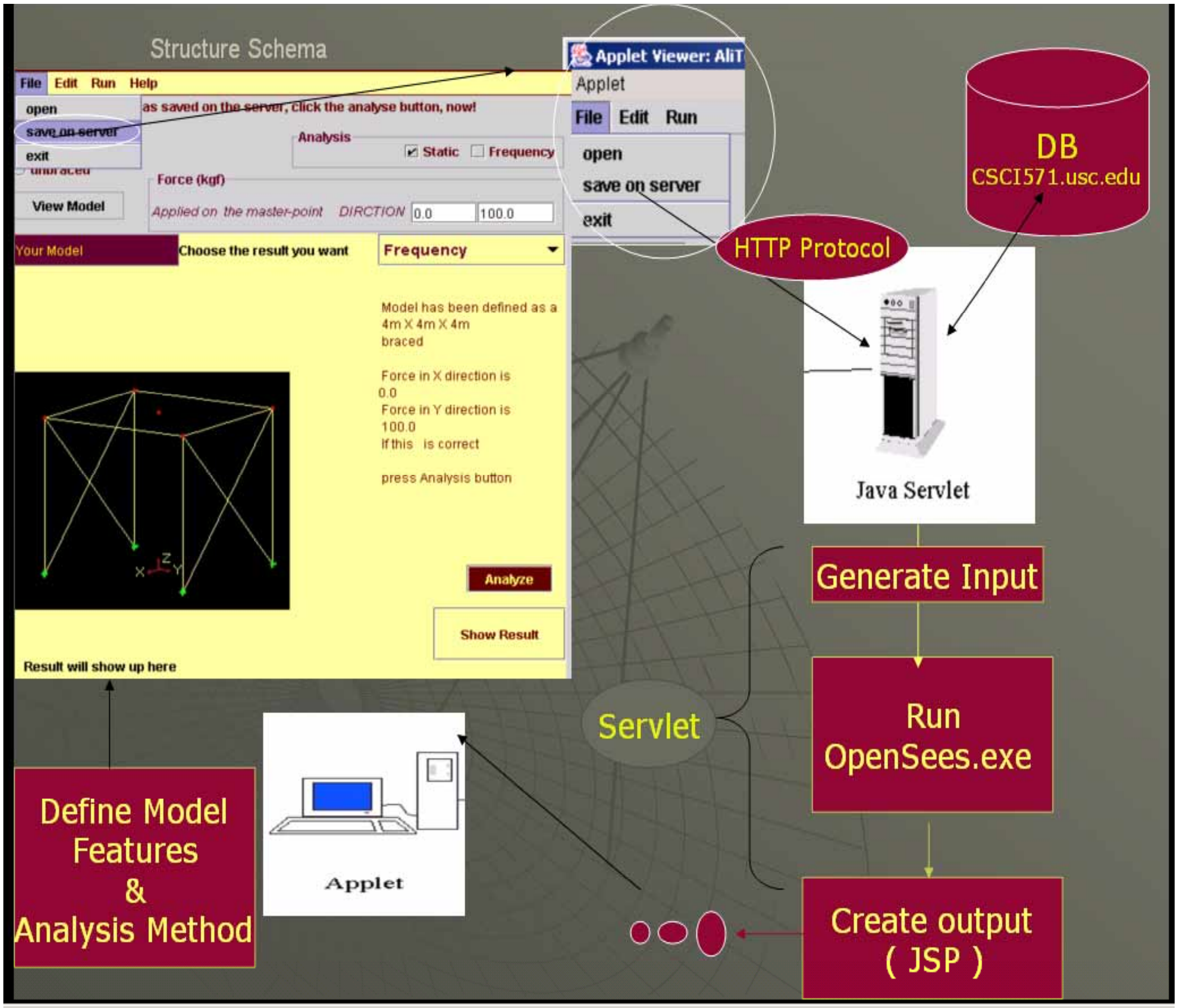

Figure 8. Overview of information flow through the prototype simulation tool.

The close coupling between the simulation tool and the GVDA test facility reduces the amount of information required from the user to perform a finite element analysis. Since the local site conditions and the base configuration of the test structure are predetermined, most of the geometrical and material properties required by the finite element analysis engine can be predefined and can be hard-coded in the server-side program. The user will need to specify only the location and properties of any desired bracing, the addition of any mass, and the loading to be applied in order to perform a finite element analysis. To provide a handy interface for the user the available options for the user will be pulled out from a database so that he/she can conveniently select from one of the existing options. This feature will also save client-side validity check of the input data that is transferred to the server.

In addition to being able to represent the geometry of the GVDA test structure, the finite element analysis engine was required to possess a number of additional capabilities. The GVDA test structure will experience passive loadings resulting from earthquake ground motions and active loadings resulting from the application of time dependent loads to the roof of the structure. Consequently, in addition to representing the structural behavior of the test structure, the finite element analysis engine must be able to reflect time dependent load conditions. The finite element analysis engine must also include the ability to represent SFSI, which will be the primary source of uncertainty in the finite element analysis. Therefore, it is important that the finite element analysis engine be capable of incorporating new representations of SFSI as the experiments lead to a better understanding of the behavior. These requirements led to the selection of the OpenSees finite element analysis program.

The OpenSees program is particularly well suited to this project. OpenSees has a number of available element and material types to define the physical representation of the GVDA test structure, as well as the ability to provide an initial representation of SFSI using soil springs. As the physics of SFSI are better understood, the open-source nature of OpenSees provides a mechanism for researchers to incorporate more complex representations of SFSI, such as rocking and uplift, within the finite element analysis program. It is also important to note that while modification of the OpenSees program requires a detailed understanding of finite elements, to simply run an analysis the user is not required to have any detailed knowledge of finite element analysis in general, or of the OpenSees program in particular. The abstraction of the complexities of performing the finite element analysis to pushing a button in the web browser allows a user with a background in experimentation, rather than analysis, to easily use the simulation tool to design SFSI experiments.

CONCLUSION

The prototype simulation tool is a web-based application that provides a simple, user-friendly, interface to perform finite element analyses of the GVDA test facility. Even though the simulation tool is closely coupled to experiment design at the GVDA test facility, the technology developed can be viewed in more general terms as a framework for remote computation and visualization. The open development environment of the simulation tool utilizes state-of-the-art resources that are also open-source and freely distributed; this facilitates both the adaptation of the tools for a particular project and the distribution of the tools among NEES-sponsored projects. The highly accessible nature of a web-based analysis tool also provides an excellent platform for Educational Outreach and Training purposes, a purpose well suited to the NEES mission. In this manner, the simulation tool could serve as both an educational resource for students and a development environment for researchers.

ACKNOWLEDGEMENT

This project is a coordinated effort within the George E. Brown Jr. Network for Earthquake Engineering Simulation, NEES, between the BYU/USC/UCSB Equipment Site and NEES System Integration teams, and is funded by the National Science Foundation.

REFERENCES

1. Apache Software Foundation. (2002). "The Apache HTTP Server Project." <http://httpd.apache.org/> (Oct. 15, 2003).
2. Apache Software Foundation. (2003b). "Apache Tomcat." <http://Jakarta.apache.org/tomcat/index.html> (Oct. 15, 2003).
3. Flagpole Project. (2002). "Software for the Flagpole Project." <http://flagpole.mit.edu/software.html> (Oct. 15, 2003).
4. MySQL. (2003). "MySQL Database Server." <http://www.mysql.com/products/mysql.index.html> (Oct. 15, 2003).
5. Pacific Earthquake Engineering Research Center (PEER). (2003). "Open System for Earthquake Engineering Simulation." <http://opensees.berkeley.edu/> (Oct. 15, 2003).
6. Peng, J. (2002). "An Internet-Enabled Software Framework for the Collaborative Development of a Structural Analysis Program." PhD thesis, Dept. of Civ. And Envir. Engrg., Stanford University, Stanford, Calif.
7. Stewart, P. Jonathan.(November 1998) "Empirical evaluation of inertial Soil-Structure-Interaction effects" Report No. PEER-98/07 Pacific Earthquake Engineering Research Center.
8. Sun Microsystems. (2003). "Java 2 Platform, Standard Edition." <http://java.sun.com/j2se/> (Oct. 15, 2003).
9. Trifunac, M.D. Todorivskam .M.I and T-Y Hao, Full-Scale Experimental studies of soil-structure interaction- A review Proc. Snd U.S.-Japan Workshop on Soil-Structure Interaction, March 6-8, 2001, Tsukuba City, Japan.
10. von Oehsen, J. B., Cyr, E. C., Cox, C. L., and Malloy, B. A. (2002). "ldmac Page 3/12/2004." *Proc., Internet Accessible Mathematical Computation 2002 workshop*, Assoc. for Computing Machinery, Lille, France.