



ELECTRONIC SEISMOLOGIST

January/February 2002

Steve Malone

E-mail: steve@geophys.washington.edu

Geophysics, Box 351650

University of Washington

Seattle, WA 98195

Phone: (206) 685-3811

Fax: (206) 543-0489

FOREWORD

The Electronic Seismologist (ES) has been goofing off for several issues, mostly sitting around and complaining about having too many earthquakes or stations to deal with and not enough time to generate beautiful prose about things electrically seismological. The ES is sure that his many (read, three) avid readers have been upset with the hiatus in ES columns and are now threatening dire actions (read, finding something useful to do). Fear not, for from the dark, dank north come rescuing words in the form of an interesting article on a unique seismic analysis system.

Most all of us who are responsible for interpreting seismograms for the purpose of producing basic earthquake catalogs would like to be able to do our work at the place and time of our choosing. Of course, seismic analysis is usually done on a fancy computer in a dark computer room near the wiring disks containing the waveform data. High-speed networking is breaking down this model, allowing for more flexibility. The ES has even been known to engage in seismogram review from home in the middle of the night or from a continent and nine time zones away from his lab. An early, simple, but still powerful technique for separating the display from the data source is the X11 system. Other techniques involve shipping data volumes over the network to workstations with complete processing systems on them. There are a growing number of systems that lie between these models, where some of the processing is at the data source and some at the user end. The topic of this month's column is such a system.

Again the group at the University of Bergen (Norway) has produced a novel set of computer software that should be interesting to all. After reading the article they submitted the ES quickly wanted to try out their system. It was easy. Just connect to their Web site, click a few buttons, and off one goes. The ES ran it from his Linux box, from a Sun, and from a Macintosh laptop; over a 100baseT Ethernet connection and a 28Kbaud dial-up to a commercial ISP. It worked just as described!! Read the following and try it.

SeisWeb: A CLIENT-SERVER-ARCHITECTURE-BASED INTERACTIVE PROCESSING TOOL FOR EARTHQUAKE

ANALYSIS

B. Moreno
Centro Nacional de Investigaciones Sismológicas
Calle 17 No. 61 e/4 y 6 Vista Alegre
Santiago de Cuba, Cuba

L. Ottemöller and J. Havskov
Institute of Solid Earth Physics
University of Bergen
Alleg. 41
5007 Bergen, Norway

K. A. Olsen
Department of Informatics
Molde College
N-6401 Molde, Norway

Introduction

With advances in computer and information technology, client-server-architecture-based tools for accessing earthquake data through the Internet have become feasible. The *SeisWeb* tool was designed to investigate the feasibility of remote access to seismological databases through the Internet. The *SeisWeb* software was written in Java. It is platform-independent and will work through a Web browser or as a stand-alone application. The tool was designed to be database-oriented and to facilitate the most common basic functions of seismological processing. We hope that its simple graphical interface and easy access will make seismology more accessible to the public, increasing both interest and understanding. In this note we discuss the client-server architecture, processing and transfer speed, the graphical interface, and security.

With the ongoing expansion of the Internet, the World Wide Web (WWW) has become an enormous source of information for people all over the globe. Seismologists use the Internet extensively for many purposes: data exchange, data transfer from the field to a central recording site, dissemination of information on recent earthquakes, and educational information. Several interactive tools for access to parametric and waveform data have been developed by various institutions. At present most of these tools are data retrieval tools, but the development of tools for processing seismological information across the Web has begun.

Among the data retrieval tools, *Wilber*, *AutoDRM* (Kradolfer, 1993; 1996), and *SeismiQuery* (references to all software mentioned are given under the URL references) are the most widely used. The concept is to download data and process it locally using software already installed. This is satisfactory for most professional users, but it is not a practical solution for the occasional user or for a nonspecialist who lacks the appropriate software on his own computer. Currently, *Seisgram2K*, the Seismicity Viewer, and the Digital Seismology Tutor present a first generation of Java-based interactive processing tools. In these tools the data are downloaded from a server to the client and the processing is done on the client side, which

means that these tools combine the task of extracting the data with processing. These tools are not linked to a parametric database.

Up to now, the processing done to obtain earthquake information has not been transparent to the public. This could change with the development of a simple-to-use Web-based processing system. Such a system could be offered to nonspecialists to display seismograms, even to pick phases and amplitudes, and to determine location and magnitude. The purpose of this project has been to investigate the issues relevant to Web-based processing and to develop some prototype software (*SeisWeb*). In the software's simplest implementation, a *SeisWeb* user will be able to inspect data on a remote database, and in a full implementation perform interactive processing. Since our effort is also directed toward occasional users and nonexperts, the plan is first to implement only the most general functions of an earthquake analysis system.

Design Concept

Client-server Architecture

The professional (or nonprofessional) user who wants to access seismological data works locally using clientside software. The data, together with the processing software, are stored on the server side. Obviously, some physical connection using communication protocols between the client and server must exist. The Internet's transfer protocol, TCP/IP, is the most widely used, and we assume that such a connection between client and server exists.

There are various possible ways that data and processing can be handled between client and server. We propose a system in which the data are kept on the server side and the processing is also done on the server side. The client software handles only the communication with the server, displays the data obtained from the server, and interacts with the user. An alternative solution would be to do the processing on the client side, in which case the data need to be transferred to the client. There are advantages and disadvantages with both systems. The "serverside processing" approach maximizes the use of existing software and guarantees the same results as obtained when working on the server directly. At the same time it minimizes the size of the client-based application. Another advantage of the serverside processing is that the complete data set does not need to be transferred, but only that needed to display the results. This leads to the disadvantage of the serverside processing, which is that many messages between client and server need to be sent. An optimization of the amount of data transmitted between client and server is therefore required. We consider that the advantages of the serverside processing outweigh the disadvantages, particularly considering future increases in communication speed.

Speed

The speed of the client-server-based system reflects both processing time on the server and the transfer rate between client and server and strongly depends on the software implementation. The processing time on the server

is the same as when working directly on the server, which means the speed should be sufficient, since powerful computers are normally found on the server side. However, the performance could decrease if a large number of users were working simultaneously using the same server.

The transfer rate between client and server represents a bottleneck. A low transfer rate may make the proposed system impractical. The system must therefore be designed to transfer as little information as possible between client and server and to make use of data compression where possible. The largest amount of data that needs to be transferred is waveform data, which easily may exceed several megabytes for one event. Due to the processing being done on the server, however, it is not necessary to transfer the full waveform data. Instead, it is sufficient to transfer only the data needed to present visual images of seismic traces. Especially when working with multitrace data, a large reduction in data size is to be expected.

Graphical User Interface

The design of the graphical user interface (GUI) is essential, especially if the software is also meant for use by nonspecialists. The success of the development effort can be measured by the users' acceptance. The GUI is required to be simple and clear enough to be self-explanatory.

In any system development task, it is essential to study the users' needs. To assess user needs, a questionnaire was prepared and sent to professional seismologists. The questionnaire was designed to determine what seismologists use the Internet for and to obtain the users' ideas on what a Web-based processing tool should look like. Twenty-six responses were used to define the main functions needed. These functions were implemented in our Web-based processing tool design.

Nonspecialists were not involved in the questionnaire because of the difficulty in finding occasional users with experience in retrieving seismological information through the Internet. We believe that the needs of seismologists will be satisfactory for the occasional user as well. Nonspecialists played the main role in refining the GUI, however. This process was performed using the "Thinking Aloud" protocol technique (Lindgaard, 1994). This method permits us to understand how the user approaches the interface and what considerations the user keeps in mind when using the interface. The main aspects considered in this study are the spatial distribution and connectivity of the options and the labeling of buttons identifying the options.

Implementation of *SeisWeb*

Following the design concept given above we developed new software called *SeisWeb*. The main objective of the software is to provide interactive access to both parametric and waveform data, and to provide functionality for basic processing of earthquake data. Access to both parametric and waveform data is event-based. *SeisWeb* was implemented and tested with the *SeisAn* software and database (Havskov and Ottemöller, 1999). *SeisWeb*, however, is independent of *SeisAn* and can work with other processing and database systems.

The main functions supported are in *SeisWeb* are:

- display parametric data such as earthquake location, origin time, source parameters, and phase readings
- search the database
- extract parametric information
- locate earthquakes
- produce epicenter maps
- extract waveform data
- display seismic traces in single- or multitrace mode
- basic processing of the seismograms (filtering, instrument correction, phase picking, amplitude reading)
- update database on the server side (login and password needed)

Platform Independence

SeisWeb is written in the Java programming language and thus is platform-independent (Weber, 1998). This means that *SeisWeb* can run on various desktop computer systems, allowing the user to connect to the database on the server from the desktop of his choice. *SeisWeb* can run either as an applet through a Web browser or as a local application. Applets are automatically downloaded upon request from the server and run on the client side in memory. Since the processing is performed on the server side, the applet remains relatively small in size, allowing fast download. There is practically no difference in speed or functionality if the software is running as an applet or as an application.

Client-server Implementation

The *SeisWeb* applet or application runs on the client side. On the server side, *SeisWeb* interacts with applications located on a Web server that handles the communication protocol and data processing. *SeisWeb* was tested with the *Apache* Web server software. The transfer of requests and data between client and server is based on the Common Gateway Interface (CGI) (Gundavaram, 1996).

Several commands for communication between *SeisWeb* and the server were defined. These commands are independent of the processing software and database structure on the server side. However, the processing software must be able to understand the requests from *SeisWeb*. A description of the *SeisWeb* commands running on the server side are available from <http://www.ifjf.uib.no/seismo/seisweb/commands.html>.

In principle, the processing software could be located on the Web server itself and the request interface implemented in the local processing software. However, to allow for greater flexibility we developed a script that interfaces between the client's requests and the serverside processing software. This script would need to be replaced when implementing another processing and database system. As an example, to request parameter data in the time interval February 1998 to April 1999, the following *SeisWeb* command is sent:

```
extract_par -base BERGE -time 19980201 19990430 -outfile
```

earthquake_list.out

The script on the server side, which is a CGI application, executes the processing software "extract_par" and sends back to *SeisWeb* the content of the output file "earthquake_list.out." An overview is shown in Figure 1.

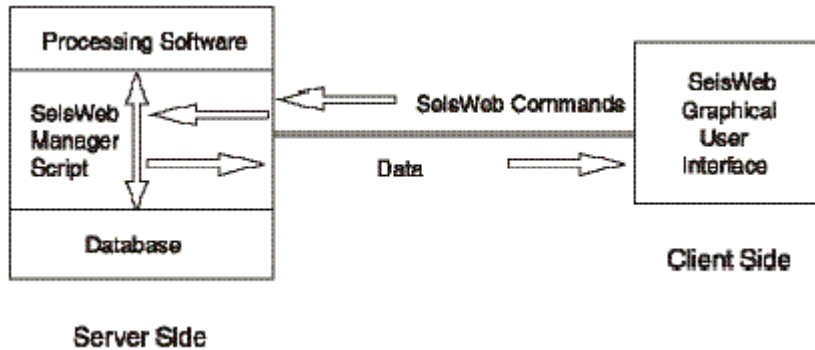


Figure 1. Overview of *SeisWeb* in client-server architecture.

User Interface

SeisWeb was equipped with a graphical user interface in the *Windows* style. User interaction is achieved through push buttons and dialog boxes. The function labels are self-explanatory. Additional help is given through an online help function. The user starts by requesting event data from a list of available databases (Figure 2). The search can be limited using search criteria. *SeisWeb* then displays a listing of events as returned by the server (Figure 3). The user can view and modify the parametric data associated with a particular event and locate the event. Epicenter maps can be generated for individual events or a number of selected events. This process on the server side is performed by a script that generates maps using *GMT* (Wessel and Smith, 1999) and standardized commands.

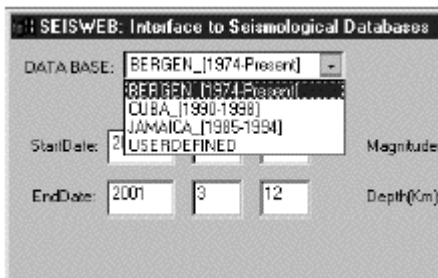


Figure 2. Selection of database to work with.

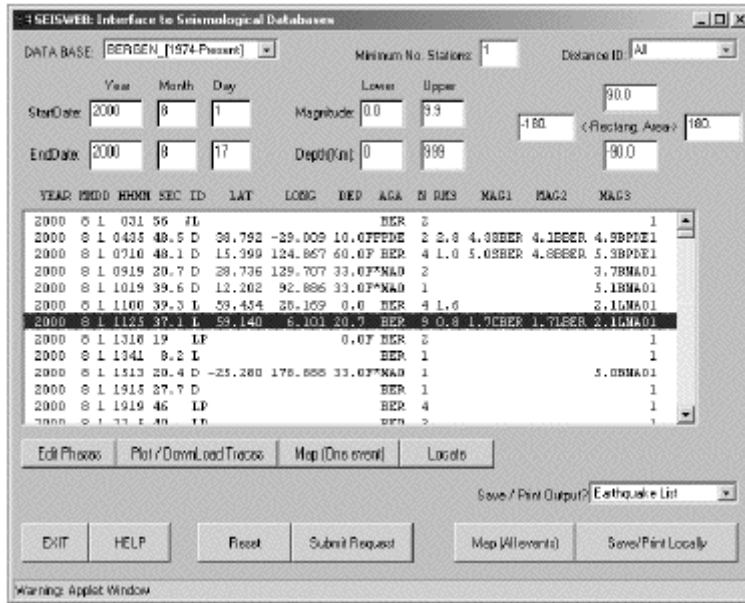
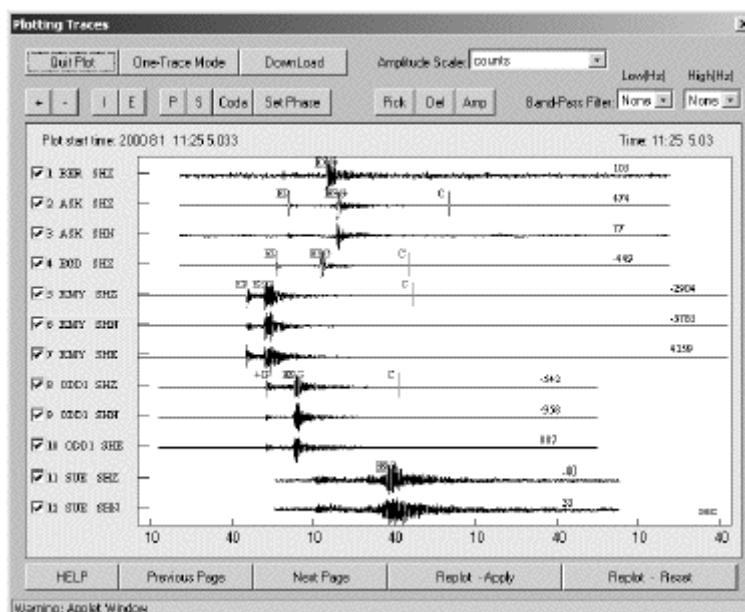


Figure 3. Main window of SeisWeb.

Based on the listing, individual events can be selected for trace plotting (Figure 4), interactive processing, and downloading of the data. To increase the speed during the waveform data transfer, only the points needed to create a visual image of the trace are transferred. This reduces significantly the amount of data to be transmitted. The user can switch between single- and multitrace mode and perform basic processing like filtering, converting traces to ground motion, and picking amplitudes and phases. Based on the processing, the parametric information is updated in memory within the applet. The user thus can perform a complete routine analysis based on waveforms and/or parametric data and then calculate the hypocentral location and magnitude.



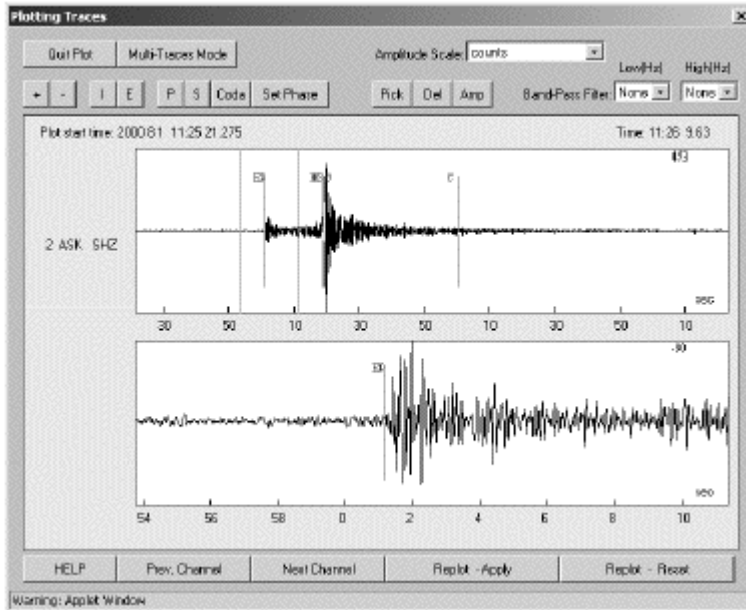


Figure 4. Windows for plotting waveforms. (Top) Multitrace mode; (Bottom) single-trace mode.

SeisWeb distinguishes between two types of databases: (1) predefined databases, which are defined on the server through *SeisWeb*, and (2) user-defined databases, which are owned by the user who has access to his account through the Web server. The user-defined database needs to be specified (database, login, password) by the user interactively and is not predefined through *SeisWeb*. Using *SeisWeb* on a user-defined database, the user can modify his data and update the database.

Formats

Both parametric and waveform data formats are in ASCII to simplify communication between different computer platforms. The most common ASCII format used for data exchange is probably GSE (GSETT-3, 1997), which was selected for the transfer of waveform data between the client and the server. The GSE parametric format, although widely used for data exchange, is a little hard to read and was therefore not used in *SeisWeb*. Instead, the Nordic format (Havskov and Ottemöller, 2000), another widely used format, was selected. The GSE waveform format includes the definition of compressed formats. For use with *SeisWeb*, the CMP6 compression was selected. For data download, both GSE and SeisAn format are supported.

Security Implementation

System security has been implemented on various levels of access to the database. The levels of access are set up on the server side in advance. Access can be restricted, *e.g.*, to not allow for waveform data download or database modification on the server side. The modification of data on the server side is possible only in user-defined databases. In this case the user

has to enter the database, login, and password. *SeisWeb* checks that the login and password are the same as defined on the server, using a modern encryption algorithm (Schneier, 1996; Rhee, 1994).

Discussion

The design concept behind *SeisWeb* as an interactive processing tool in earthquake seismology is novel in the sense that it is based on a client-server architecture where most of the processing is performed on the server side. In other applications (e.g., *Seisgram2K*) the processing is performed on the client side. The main advantage of our design concept is that the tool can work with various processing and database systems. To work with a remote database and processing system, the user does not have to know the database structure and the processing system. All that is needed is a computer with an Internet connection and Web browser software. With the increase of telecommunication speed and advances in computer technology this approach may become more common.

Another novel aspect of *SeisWeb* is that the parametric and waveform data are associated and can both be inspected interactively. It is obvious that the professional seismologist could benefit from a tool like *SeisWeb*, since it will allow him to inspect and process earthquake data interactively without downloading it. *SeisWeb* will also provide easy access to seismological databases and basic processing tools to nonseismologists. Giving direct access to the data may increase the nonseismologist's interest and understanding and could be beneficial to educational projects.

The most critical aspect of *SeisWeb* is the speed of Internet connections, since with low transfer rates its use becomes impractical. We assume that the server has the capacity to process the data at least as fast as the local machine would, so delays when compared to clientside-based processing are caused by the communication between client and server. On the slow end, *SeisWeb* was found to work reasonably fast with a 28.8K baud modem connection or 64K baud ISDN connection.

Conclusion

We have developed an interactive processing tool for earthquake seismology based on client-server architecture. We found that with the current transfer rates and computer systems, client-server-based processing across the Internet is already feasible. The design allows *SeisWeb* to work with various processing and database systems. Basic routine processing functions were implemented with an easy-to-use graphical user interface. Considering the large number of seismological databases that could be accessed through the Internet, there is a good potential to make these data available for interactive use to professional seismologists and to nonexperts. *SeisWeb* is not only a working tool in earthquake analysis but also an educational training tool for better understanding of seismology.

How to Get and Test *SeisWeb*

SeisWeb can be tested and obtained from the Institute of Solid Earth Physics, University of Bergen, Web server:

<http://www.ifjf.uib.no//seismo/seisweb/seisweb.html>.

ACKNOWLEDGMENTS

We would like to thank Kuvvet Atakan and Margaret Grandison for their fruitful comments and reviews of the manuscript. Susan Hough, John Ebel, and Steve Malone made valuable suggestions for improving the overall presentation of the manuscript. We also wish to express our gratitude to all who filled in the questionnaire.

REFERENCES

- Group of Scientific Experts Third Technical Test, GSETT-3 (1997). Provisional GSE 2.1, Message Formats & Protocols, Operations Annex 3, May 1997.
- Gundavaram, S. (1996). *CGI Programming on the World Wide Web*, O'Reilly & Associates, Inc.
- Havskov, J. and L. Ottemöller (1999). Electronic Seismologist: *SeisAn* earthquake analysis software, *Seism. Res. Lett.* **70**, 532-534.
- Kradolfer, U. (1993). Automating the exchange of earthquake information, *Eos, Trans. Amer. Geophys. U.* **74**, 441-448.
- Kradolfer, U. (1996). *AutoDRM: The first five years*, *Seism. Res. Lett.* **67**, 30-33.
- Lindgaard, G. (1994). *Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems*, Chapman and Hall, London, U.K.
- Rhee, M. Y. (1994). *Cryptography and Secure Communications*, McGraw-Hill Book Co., Singapore.
- Schneier, B. (1996). *Applied Cryptography, Second Edition, Protocols, Algorithms, and Source Code in C*, John Wiley & Sons, Inc.
- Weber, J. L. (1998). *Special Edition Using Java 1.2*, Fourth Edition, QUE, United States of America.
- Wessel, P. and W. H. F. Smith (1999). *The Generic Mapping Tools Technical Reference and Cookbook*, Version 3.2, School of Ocean and Earth Science and Technology, University of Hawai'i, 115 pp.

URL References

Apache Software Foundation: <http://www.apache.org>

AutoDRM: <http://seismo.ethz.ch/autodrm/>

Digital Seismology Tutor: <http://lbutler.geo.uni-potsdam.de/TutorII/>

Seisgram2K: <http://www-geoazur.unice.fr/PERSO/lomax/seisgram/SeisGram2K.html>

Seismicity Viewer: <http://www-geoazur.unice.fr/PERSO/lomax/nlloc/soft2.10/seismicity.html>

SeismiQuery: <http://www.iris.washington.edu/SeismiQuery>

SeismoSurfing: <http://www.geophys.washington.edu/seismosurfing.html>

Wilber: <http://www.iris.washington.edu/cgi-bin/wilber1frame.pl>

SRL encourages guest columnists to contribute to the "Electronic Seismologist." Please contact Steve Malone with your ideas. His e-mail address is steve@geophys.washington.edu.

Posted: 1 November 2002

