

A WEB-BASED SEARCH ENGINE FOR THE RETRIEVAL OF EVOKED POTENTIAL RECORDS

Michael J. Pougnet, BScE, PhD candidate & Dennis F. Lovely, PhD, PEng
*Department of Electrical & Computer Engineering, University of New Brunswick, NB,
Canada*

Abstract – At CMBEC30 a data management protocol for evoked potential records was introduced. This protocol presented a standardized approach to the acquisition, storage, and retrieval of evoked potential data. Since then, various updates have been made to the design in an attempt to improve data security and ease-of-use. This paper discusses the improvements made for data retrieval.

INTRODUCTION

Data management for biological signals is generally non-standardized. Although acquisition procedures between different research centres may be similar in nature, standard file types and data structures are rarely seen. In addition, many laboratories leave management and storage of acquired signals solely to the researcher. These factors limit the reusability of files and also result in lost data. For the last three years, a standardized approach to data management for evoked potential signals has been in development at the University of New Brunswick (UNB) [1]. This protocol uses a custom acquisition program that standardizes data structures and file types, a file manager that transfers acquired files to an offsite database, and a search engine that allows registered users to download files. The purpose of this new approach is to eliminate the loss of data, and allow for an easier sharing of evoked potential files between researchers and institutes.

BACKGROUND

Biological signal processing is a major area of research at UNB. Investigations into somatosensory evoked potentials (SEP) is just one field of specific interest. Acquired from the peripheral nerves, these signals are induced by way of electrical stimulation. Some problems that exist with SEPs include low signal-to-noise ratios and corrupting stimulus artifacts. Ongoing research at UNB involves applying specific signal processing techniques to eliminate some of these aforementioned problems. In many cases, the acquisition of various high resolution SEPs is needed to properly evaluate these new techniques. Because SEP recordings are large in nature (~10MB), and loss of data is a common occurrence due to human error,

data management is of major concern. In an attempt to alleviate these concerns, UNB researchers have been developing a standardized practice for data management.

Until 2007, standardization of SEP data at UNB was limited to the methods in which signals were acquired. Acquisition software, written in MATLAB as a graphical user interface, provided the necessary burst sampling routines to collect SEP data [2]. This software gave a proprietary file extension and file structure to the SEP recordings. Previews of the recording were also created automatically and saved as a Joint Photographic Experts Group (JPEG) image with the data file. At this point, data storage and data sharing were left to the discretion of the researcher.

In 2007, significant steps were taken in an attempt to further data management and data sharing at UNB. These steps incorporated the previously designed acquisition software with a MySQL/File-transfer-protocol (FTP) data server for file storage and a JAVA-based search engine for data retrieval [3]. This new protocol worked by transparently connecting to a data server upon opening of the acquisition software. Each time a signal was acquired, its file would be transferred to a repository on the data server via FTP. Relevant experimental settings from each file would then be used to populate a MySQL data table. This data table would then form the basis for search criteria used in the JAVA-based search engine.

To recover files from the database, a custom designed search engine, named **SEPsearch**, was created. This JAVA-based application gave users the ability to filter a database of SEP files based on four important experimental settings (recording polarity, recording site, stimulation site, stimulation type) extracted from the MySQL data table. The user could then download desired files via a direct connection to the FTP data server.

SEPSEARCH

Since its initial design, **SEPsearch** has undergone various changes and improvements. First, a transition from a JAVA application download to a browser-run webpage was completed in order to comply with UNB firewall restrictions. This change also restricts users

from a direct connection to the data server. Increased flexibility in providing search criteria was also completed. Now, users can better tailor their searches as they are not restricted to only four search options. Search options are populated from the MySQL data table in order to provide flexibility when new search criterion is added. The FTP server for data storage was also substituted for a data table in the MySQL database. This was done to limit the number of open connections to the data server. Finally, upon completion, the utilization of the UNB Integrated Technology Services (ITS) will be implemented to provide services such as updates and backups.

All of the modifications to **SEPsearch** have yielded an easier to use and more secure data management protocol. The protocol secures both the acquisition computer and the data server behind a firewall, and restricts all database searches through a web-based search engine. Connections made to the web server are done via the search engine. The web server then interrogates the data server with the user's queries and returns the results back through the web server. This new data management protocol is shown in Figure 1.

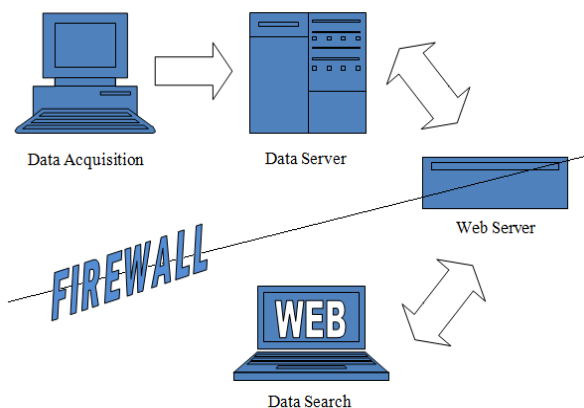


Figure 1: Data Management Protocol.

For a better understanding of how **SEPsearch** works, an explanation of both the MySQL database and the structure of the web-based search engine are necessary.

Database Overview

This project uses a MySQL database server. MySQL software is free-to-use and is fully supported by the ITS at UNB. For the purpose of development, the current database server is running on an office PC behind the UNB firewall. As previously mentioned, upon finalization of this project, this database will be moved to ITS where servicing will be done regularly.

The database server is divided into six tables. These tables are *level2*, *level3*, *level4*, *searchcriteria*,

users, and *data*. Each table is used to retain important information needed for the search engine to run.

The *level "x"* tables hold filenames and relative information about each file in the database. This information includes the various experimental settings previously discussed such as stimulation site, recording site, recording polarity, etc. The numeric variable "x" corresponds to a security level. For example if $x=4$ then the search engine user has a security level of 4 and will complete all of his or her searches on the *level4* table. Likewise if $x=2$ or $x=3$ the search engine user will be directed to the *level2* and *level3* pages respectively. For more information regarding the security levels see the earlier publication [1].

The *searchcriteria* table is a new feature of the database system. Rather than the search criteria being hard coded into the web server, this table contains a compilation of all the different search items and options. In this way additional search items and options can be added without modifications to the search engine. An example of an item and option set held in the *searchcriteria* table is:

Search Item	→ Recording Polarity
Search Options	→ Monopolar, Bipolar

The *users* table includes pertinent information about each registered search engine user. Information in this table may include, but not be limited to, first and last name, security level, and university or clinic affiliation. User passwords are also kept in this table, but are secured by encryption.

The *data* table acts as a file repository for the search engine. Here, all data acquisition files are stored as Binary Large Objects (BLOBs). This method of data storage also represents a new feature of the data management protocol. Previously, a secondary file transfer protocol (FTP) server was used in conjunction with the MySQL database server. Although all search engine fields were stored in the MySQL data tables, actual data files resided on the FTP server. By incorporating the use of BLOBs, the management protocol has been streamlined. Data file security has been increased while the number of open connections needed by the search engine has been reduced. It is also important to reiterate that upon finalization of the project UNB ITS will adhere to their own protocol of backing up any files residing on their servers, which will now include the evoked potential data. The *data* table also holds the preview images of each file and other information such as file size.

Website Overview

The search engine website is divided into six forms. These are as follows:

- SEPsearchlogin** - initial entry point
- SEPsearch** - search criteria entry
- SEPsearchresults** - list of files found
- SEPsearchpreview** - graphical preview on a file-by-file basis
- SEPsearchgraphics** - streams JPEG image to **SEPsearchpreview**
- SEPsearchdownload** - list & transfers files to user

A simplified flowchart of how these forms interact is shown in Figure 2.

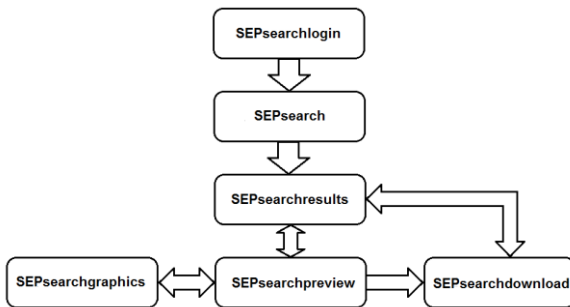


Figure 2: Flowchart for web-based search engine.

Each of these six forms consists of an “.aspx” and an “.aspx.vb” file. The “.aspx” files hold the client-side (or frontend) code. These files are written in HTML with ASP.NET controls [4]. The “.aspx.vb” files hold the server-side (or backend) code. These files are written in VB.NET and are responsible for interrogating the data server. The backend can also dynamically alter the frontend based on query results.

SEPsearchlogin is the first window the user sees upon opening the website. Here the user is prompted

for login credentials. If verified through the *users* table of the database, the user will gain entry to the main search window (**SEPsearch**).

At the **SEPsearch** window the user has three options for defining his or her search criteria. These options are searching the database for keywords, obtaining all data files, or using various “dropdown” lists to tailor search results for specific acquisition parameters. Again, it should be noted that each dropdown list within the **SEPsearch** window is dynamically populated by the fields found in the *searchcriteria* table of the MySQL database. Figure 3 shows a typical screenshot of the **SEPsearch** window.

Once the user has selected an option for searching, a call is made to the web server to interrogate the data server for matching results. The user is then directed to the **SEPsearchresults** window where all results from the data server are listed. Here the user can see important information about each search result and flag one or more of the files for download using the checkbox. To assist in the decision of downloading a file, the user can preview the data in a graphical format using the “Preview” button. A screenshot of the **SEPsearchresults** window is shown in Figure 4.

If a user decides to preview a selected file, he or she will be directed to the **SEPsearchpreview** window. This window shows the plot of sample data taken from the chosen file. It also displays comments from the researcher regarding the quality or the specificity of the file. The **SEPsearchgraphics** form works with **SEPsearchpreview** window in displaying the sample data plot. Although **SEPsearchgraphics** has both frontend and backend code, the frontend is never shown. **SEPsearchgraphics** simply uses backend code to acquire and stream sample data plots from the data server to the **SEPsearchpreview** window. Figure 5 shows a screen shot of the **SEPsearchpreview** window.

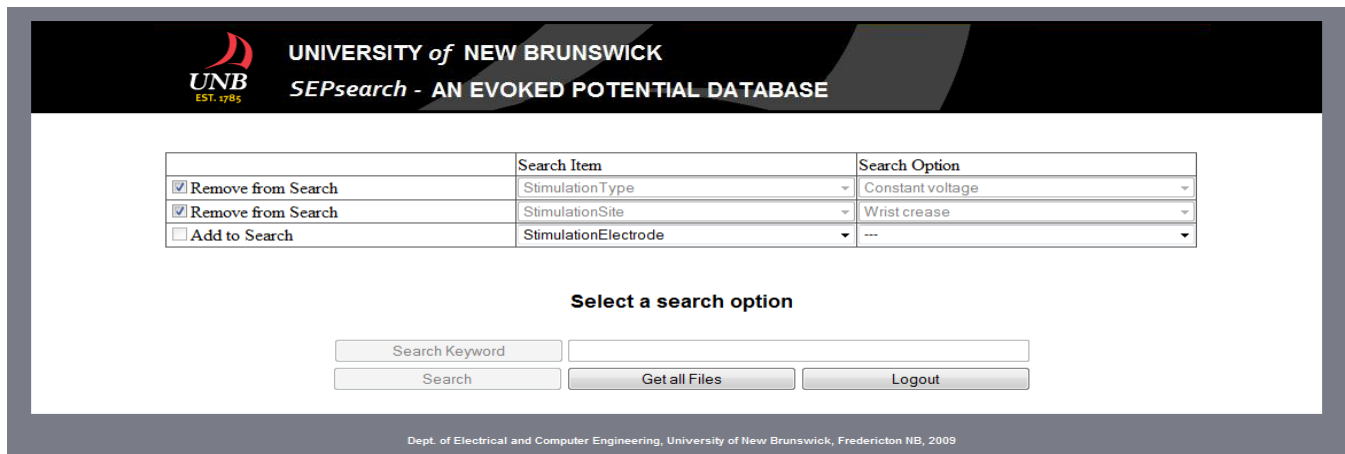


Figure 3: SEPsearch screenshot.

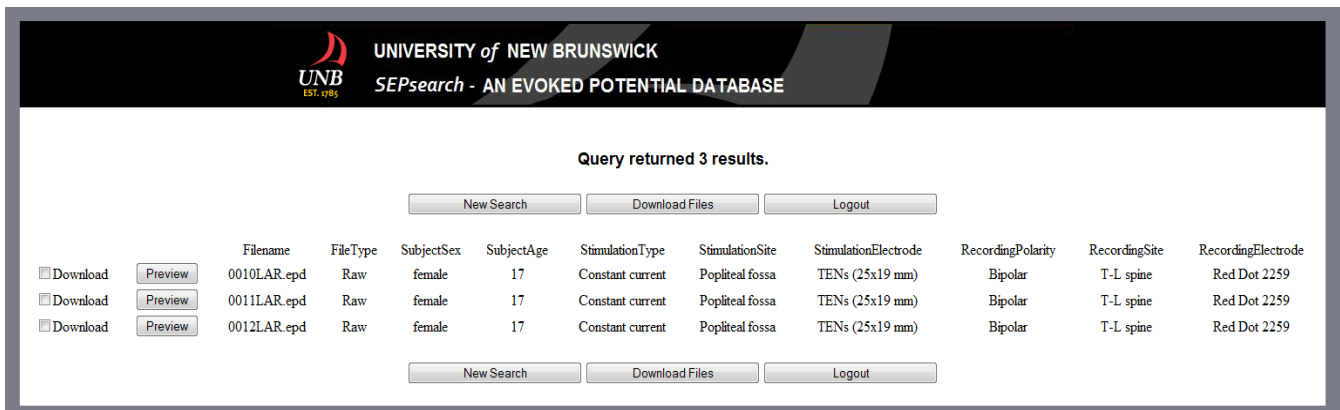


Figure 4: SEPsearchresults screenshot.

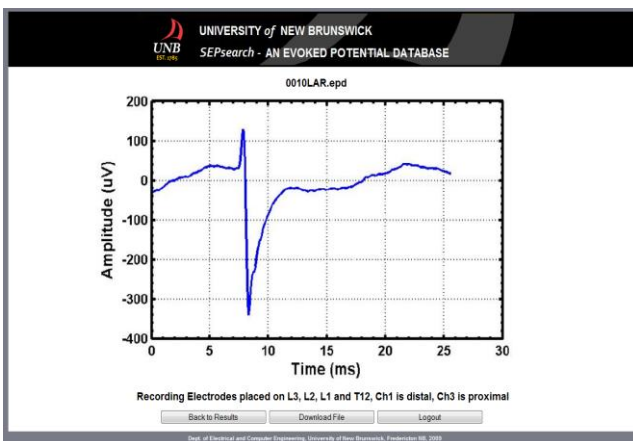


Figure 5: SEPsearchpreview screenshot.

Once all the required files have been flagged for download, clicking on the “Download Files” button will direct the user to the **SEPsearchdownload** window. This window simply initiates the download and lists the files that are being transferred.

Upon completion of download, database users can manipulate data files using custom designed viewing and processing software made available by UNB. If a researcher desires to manipulate the data files using their own software, files can be opened via the “*Import Data*” option in MATLAB. Here the user will find all channels of data, a header file explaining how the data is saved, and a list of experimental settings used during recording.

CONCLUSION

Building on work first seen at CMBEC30, an updated search engine has been discussed. This updated engine provides a new data management protocol that will increase both data and user security. The new search engine design also aims to make UNB recorded data files more accessible to other researchers. This is done by the transition from a

JAVA application download to a browser run webpage. Other improvements from the initial design include the removal of an FTP server for data storage and the utilization of the UNB ITS to both update server applications and backup up existing data files. Finally, steps have been taken to ensure the server can be easily modified in the future as search options and items are dynamically created from an existing MySQL database table.

FUTURE WORK

Although the project is in its latter stages, more work is still to be done. Primarily this project needs to be moved from the development environment to a proper web host. This will inevitably result in minor problems and will most likely lead to slight code changes and new or modified designs.

A second future area for development would be the finalization of a previously suggested WEB based registration process for users of the system. While the existing service does have a user registration, a more robust registration process is desired.

Once the evoked potential database has been fully tested and is functional, expansion into other biological signal databases may follow. Due to the extensive research into both gait and electromyography at UNB, management of such large quantities of data is essential. Through the flexible infrastructure developed for the management of evoked potential data, it should be a fairly simple task to include these new data types into this framework.

REFERENCES

- [1] M. Pougnet, D.F. Lovely, “Real-time database management of evoked potential records,” presented at the 30th Annual Conf. CMBEC, Toronto, ON, Canada, 2007.
- [2] The Mathworks Inc., Natick, MA 01760, USA
- [3] Sun Microsystems Inc., Santa Clara, CA 95054, USA.
- [4] Microsoft Corporation, Redmond, WA 98052-6399, USA.