

Relational Database Linkage of Scientific Applications and Their Data Files

Ira Rudowsky¹, Olga Kulyba¹, Mikhail Kunin¹, Dmitri Ogarodnikov² and Theodore Raphan^{1,2}

¹Institute of Neural & Intelligent Systems
Department of Computer and Information Science
Brooklyn College of CUNY
Brooklyn, New York, 11210

and

²Department of Neurology, Mt Sinai School of Medicine
New York, New York, 10029

Phone 718-951-4452; Fax: 718-951-4842; E-mail: rudowsky@brooklyn.cuny.edu

Abstract - Research in computational neuroscience has been following a model-based approach where data is comprised of digitized streams of sampled analog data, images and voice. The data are generally contained in files and specialized programs are used to analyze the data. In this study we developed a prototype system for indexing and retrieving information for use by an application that analyzes data. The data in the files consist of channels derived from analog and event driven sources. It is also linked to video images associated with the data acquisition. In this research, we developed an indexing capability that threads into the data acquisition and analysis programs to give the system a broad data base capability. We designed tables and relations within the database for indexing the files and information contained within the file. The system has the potential of giving us retrieval capabilities that include analog, event, and video data types.

I. INTRODUCTION

Research in computational neuroscience comprises not only digitized streams of sampled analog data but also images and voice [1], [2]. We have developed a system, called VMF, whose design is based on the idea that data files contain channels of data streams [3]. The data are generally contained in files that contain the channels and specialized programs have been developed that read the channels, which can then be subjected to a wide range of analysis [4], [5]. Two types of channels are currently implemented. An analog channel is a data stream that is a sampled version of analog data. An event channel identifies specific times that events occur. The program is flexible in that it can handle data on eye movements (See [6] for review) as well as unit recordings of single cell activity in the brainstem of alert animals [7]. The VMF program accesses these files and has the ability to operate on them by a wide range of data analysis modules, correlating the data in different channels or performing specific analysis on a single channel, such as FFT or

wavelet transforms. We have now developed a capability for video data stream channels so that applications associated with the VMF analysis program will be capable of analyzing video data related to eye movements [8], an important new area in research. Inclusion of audio would also be useful in the identification of the file. At present, the only method we have for identifying and retrieving the files is through header information contained in the file, which must be accessed and viewed by eye. Thus, it is important to develop a database system, which can appropriately interact with this file structure to identify and retrieve files so that we can have wider collaboration amongst researchers that would like to access this data.

Work on database management has focused on relational databases as a robust and fast retrieval system that can be upgraded to include video and voice [9]. However, developing schemes for integrating database structures with applications that include data over a wide range of formats has not been adequately addressed. The purpose of this study is to establish a prototype system for indexing and retrieving information and interacting with applications for data acquisition and analysis.

II. BACKGROUND

Early database systems focused on textual data and much work has been done on data structures that could be searched through keys to efficiently and quickly retrieve information [10], [11], [12]. Scientific data has characteristically been stored as streams of digitized data, usually preceded by some header block that describes the organization of the streams and some of these files also include video and voice information [13], [14]. Classical database systems are of limited utility in accessing the data in such files [15], [16]. Moreover, classical database design does not have the capability to interact with applications that use digital

streams representing analog and video information as input. Expansion of data types in relational databases now offer the possibility of including still and streaming video images, characterized as multimedia data [17]. These relational database systems are capable of interacting with data types that have high storage and bandwidth requirements and have the capability to be queried and retrieved based on content [18],[19],[20] [21].

IBM's DB2 and Oracle are the two main commercially available systems that could provide full multimedia support for the scientific application we are proposing. In particular, Oracle has served as the object-relational database for indexing and context-based queries in multimedia databases [22], [23], [16]. Oracle9i also provides a component, interMedia, which is a set of services that facilitates the storage, management and retrieval of multimedia content in an integrated fashion with other enterprise information over the internet. For these reasons, Oracle was utilized in our initial development of the system. We adapted the relational database (Oracle9i) and developed an interface between an application program (VMF) that is presently used for data analysis, so that files containing data streams can be retrieved in a robust manner. At present, a data acquisition program creates and stores data into VMF files as channels, but has no capability for storing indexing information. In this study, we designed a prototype relational database for connecting the VMF application software and the files on which it operates.

III. RESULTS

The relational database was designed to have the capability to index metadata that describe the experiments. A user friendly, graphical front-end application, Database Interface Application (DIA), written in C/C++ was developed to enable the querying of the metadata based on numerous criteria. Text fields and pre-filled, drop down combo boxes are the locations for data entry. The criteria fields include items such as: (a) experiment number (a) subject species (human, monkey, rabbit, rat, mouse), (b) subject gender, (c) apparatus used (a number of rotation devices, linear movement device, human centrifuge device, treadmill information, etc), (d) date range of the experiments, (e) trials numbers within an experiment and (f) medical condition of the subject.

Once the criteria have been supplied, an SQL query is generated and transmitted to the database engine. The resulting set of records is returned to the

client machine and displayed in a graphical, tabular format along with key pieces of indexing information. This enables the researcher to (a) narrow down the possible candidates for viewing and (b) indicate additional experiments that might be considered for review in tandem given their potential inter-relationship based on the selection criteria. This organization enables the user to select an individual record, which will pass the VMF filename to the VMF application program for visualization and analysis of the digitized analog channels and event channels.

A. Database Table Design

The database tables that represent the metadata were normalized to maintain the indexing information related to the VMF files in a consistent manner (Fig. 1). Seven tables were defined: SUBJECTTYPE, SUBJECTCONDITION, SUBJECT, SUBJECTGROUPID, EXPERIMENT, APPARATUS, SUBJECTCLASS (Fig. 1, Gray Headings). Each of these tables has a composite primary key (Fig. 1, Bold fields in EXPERIMENT table) or a single primary key (Fig. 1, Bold Fields). In some instances, a field in a table is linked to a primary key in another table and is referred to as a foreign key. The Foreign keys are not necessarily required to have the same name as the primary key. For example, SUBJECT_ID in the EXPERIMENT table is a foreign key for primary key, SUBJECT_ID, in the SUBJECT table (Fig. 1).

The SQL queries, generated by the Data Interface Application, are submitted to the Oracle database, which accesses these tables and responds to the client with a resulting set of records. The links between the foreign and primary keys enable the search engine to quickly reduce the search space for a query. For example, a query about experiments on apparatus used only by animals accesses the SUBJECTCLASS table, eliminating the need to link through to the SUBJECT table. The robustness of the database design for storing the data and for querying using numerous criteria that have been indexed is demonstrated by the multiple links from the SUBJECT table to other tables (Fig. 1). To classify the type of subject, human or specific animal type, there is a foreign key to the SUBJECTTYPE table. For human subjects, ethnicity information is available via a link to the SUBJECTGROUPID table. Finally, the medical condition of the subject is identified via a link to the SUBJECTCONDITION table. The graphical user interface links the VMF application to the channels in the files.

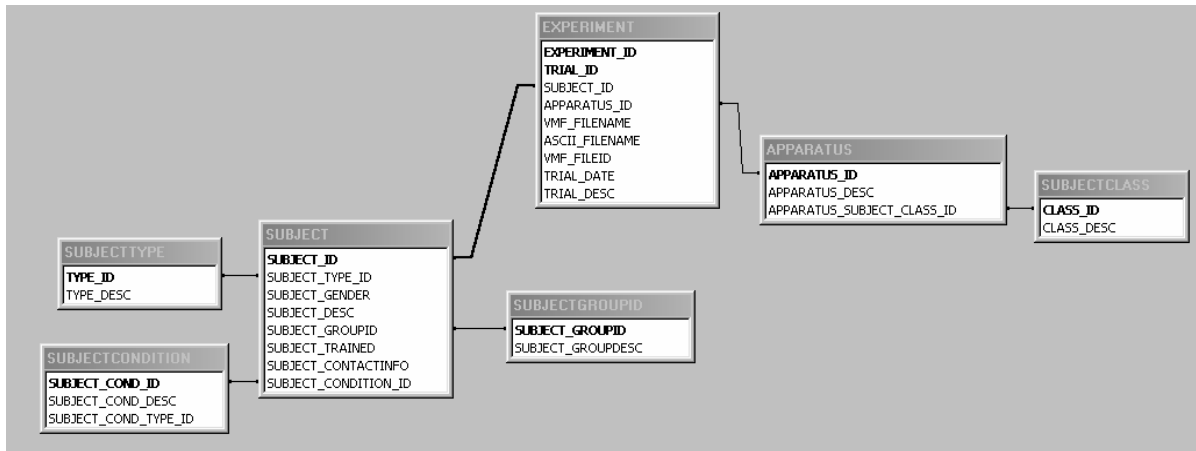


Fig. 1 Flowgraph of database tables.

B. The Data Interface Application

The Data Interface Application (DIA) provides functionality via four tabs - Logon, Query, Result and DBManage, corresponding to four screens. Logon provides application security using Oracle database security features so that access to subject information is limited to those with proper clearance. This is The Query tab provides the capability to formulate SQL queries and submit them to the Oracle database in order to retrieve those experiments that fit the selected criteria. When the submit button is activated, the DIA automatically displays the results in the Results screen. If another screen is selected for examination, the results of the query can be re-accessed by activating the Results tab. The user can view the returned rows from the query, along with some descriptive fields, and then click on a row to display the VMF file via the VMF Analyze program. The DBManage tab provides a series of screens to maintain the Oracle tables in a user-friendly environment.

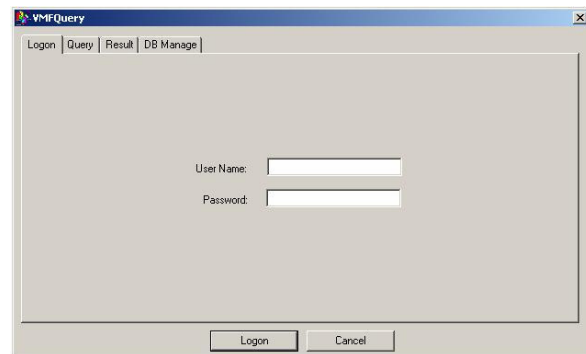


Fig 2. Logon Security Screen

1) *Logon tab:* Medical information obtained from human subjects participating in experiments is subject to the governmental legislation under the Health Insurance Portability and Accountability Act (HIPAA) Privacy Regulations. In order to accommodate these requirements, a Logon feature (Fig. 2) and associated tables are included in our design. Each user is assigned an id and password by the administrator of the system and is placed in a user group that determines access rights. For example, the administrator group has full access (read/write) to all tables, power users have read access to all tables, standard users have access to all tables except those dealing with the personal

information regarding a subject. This access is true for on-line viewing as well as reporting. A user in the standard user group will not be able to view via Result tab or generate a report containing any detailed information regarding a subject. The only piece of information allowed to be seen is the anonymous subject id. Similarly, the Subject table will not viewable in the DBManage tab.

2) *Query tab:* The Query screen (Fig. 3) has a number of fields from which the user can broaden or narrow the search. Most fields have been designed as drop-down combo boxes from which the user selects from pre-filled entries. These entries come from the various tables within the database. Some fields are related to and control the contents of other fields. For example, by selecting Experiment ID, the fields Trial ID, Subject ID, Apparatus, Start Date, End Date, and Location are limited to those occurring within that experiment. By selecting just Apparatus, a much broader result-set will be returned. In addition, Subject Type will be limited to human only or the various animal types. Selecting Subject ID will automatically determine Subject Type.

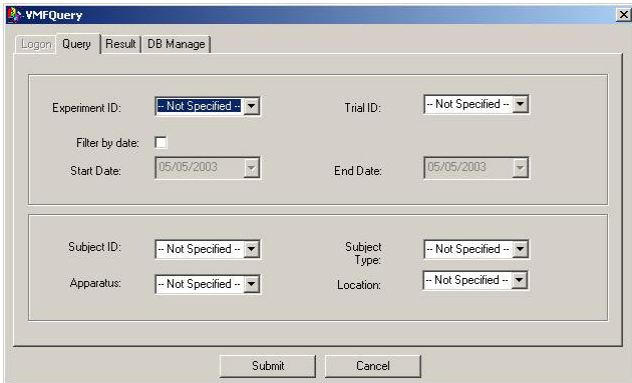


Fig 3. Data Interface Application – Query of all experiments

Date ranges can be entered as well. By first selecting the checkbox ‘Filter by date’, the user can then enter the start and end dates. The program employs Windows’ CDateTimeCtrl and dateTimePicker controls which do not accept invalid or empty dates. By default it displays the current date. If Experiment ID or ExperimentID/Trial ID is selected, the date fields will contain the start and end dates for that set of records. The user can choose, at any point, not to use the date parameters by simply unselecting the checkbox. As the dates are selected from a calendar, the only possible error is if the start date is greater than the end date. A check is made for this error and, if it occurs, a popup message appears that flags this error.

3) *Result tab*: By clicking on the Submit button on the Query screen, a SQL query is constructed by the program and submitted to the Oracle database. The results are returned and displayed in tabular format as shown in Figure 4. The VMF analysis program can be accessed in two ways. One way is to click the appropriate box in the Experiment ID column and then click on the VMF button at the bottom of the screen. Another way to access the VMF is to double click on the row containing the name of the data file to be analyzed. The parameters from the Query screen are displayed as read-only fields at the bottom of the Result screen so that the user will not have to toggle back and forth between the Query and Result screen. Modification of the Query parameters will not update the Result screen until the Submit button is selected. In order to prevent the user from making incorrect assumptions about the data on the Result screen, a message will appear below the table of results indicating the fact that there is a mismatch between the Query and Result parameter displays (Figure 5, shown in red).

4) *DBManage tab*: As depicted in Figure 1, there are a number of tables that form the relational database

supporting this application. Updates, additions and/or deletions are necessary to these tables to reflect the changing experimental environment. For example, additional apparatus may be introduced, new subjects will be included in the testing, changes to subject contact information, corrections to incorrect experiment and/or trial information entered at the time of data entry, etc. In order to provide a user-friendly, non-Oracle based environment to make these changes, a number of screens have been developed to support that capability. Figure 6 depicts the maintenance screen after the APPARATUS table has been selected from the drop-down combo box that lists the tables that can be accessed. Once selected, the rows and columns of the table are displayed (Figure 6) for update. The display window scrolls both horizontally and vertically to enable all columns and rows to be accessed.

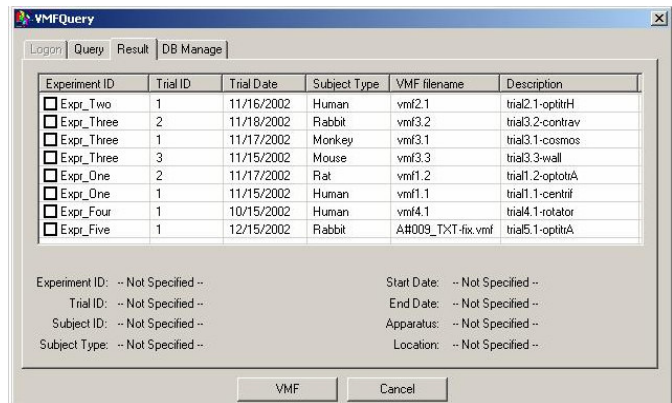


Fig 4. Data Interface Application – Results from query in Figure 2

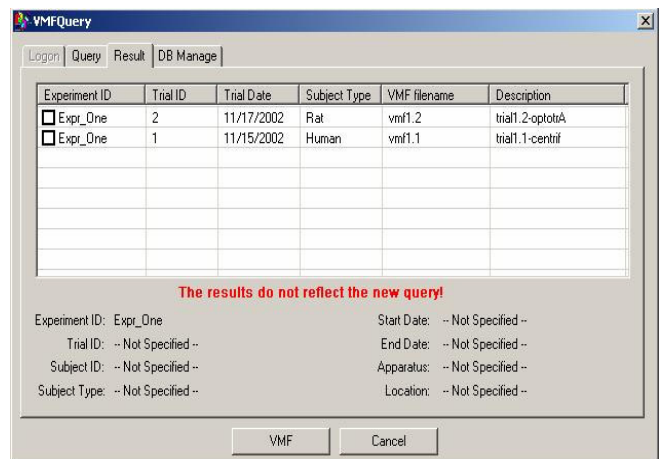


Fig 5 Message that query has been changed but results not updated

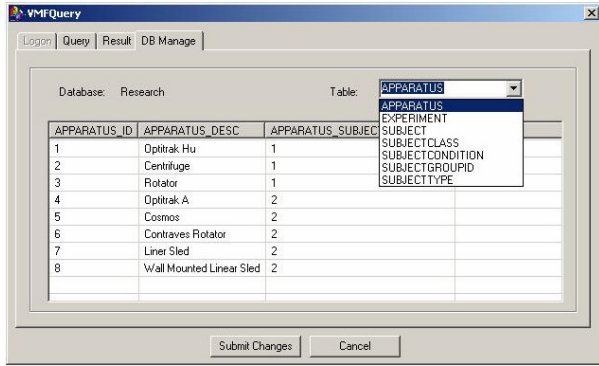


Fig 6. Maintenance screen – table selection

IV. SUMMARY AND CONCLUSIONS:

Results indicate that the relational database is a flexible and easy way to interface data analysis applications to a wide range of data types. Future research will extend to automating the capture of indexing information directly from the Data Acquisition Program. This will eliminate the double entry of data with the increased chance of inconsistent information. The next level of research will extend the querying capability to content based searches in addition to the metadata querying. This will further enhance the analysis performed by the researchers by providing them with a technological tool currently not available. The relational database also has the possibility of including security restrictions to protect subject confidentiality.

ACKNOWLEDGMENT

This work was supported by grants P30 DC05204, DC05222, EY04148 from the NIH and NASA Cooperative Agreement NCC 9-58 with the National Space Biomedical Research Institute.

REFERENCES

[1] Wang, S.S., and Starren, J., A Web-based Secure, Light Weight Clinical Multimedia data Capture and Display System. *Proc AMIA Symp*, 898-902, 2000.
 [2] Cai, W., Feng, D.D., and Fulton, R., Content-based Retrieval of Dynamic PET Functional Images, *IEEE Trans Inf Technol Biomed*, 4(2):152-8, June, 2000.
 [3] Kushiuro, K., et al., Compensatory and Orienting Eye Movements Induced by Off-vertical axis rotation (OVAR) in monkeys. *J Neurophysiol*, In Press, 2002.

[4] Frankewitsch, T., and Prokosch, H.U., Graphical Tool for Navigation Within the Semantic Network of the UMLS Metathesaurus on a Locally Installed Database, *Stud Health Technol Inform* 77:847-51, 2000.
 [5] Smyth, P., Data Mining: Data Analysis on a Grand Scale?, *Stat Methods Med Res*, 9(4):309-27, Aug, 2000.
 [6] Raphan, T., and Cohen, B., The Vestibulo-Ocular Reflex (VOR) in Three Dimensions, *Exp Brain Res*, 145:1-27, 2002.
 [7] Yakushin, S.B., et al., Changes in the Vestibulo-ocular Reflex After Plugging of the Semicircular Canals, *Ann NY Acad Sci*, 942:287-299, 2001.
 [8] Zhu, D., Moore, S.T., and Raphan, T., Robust Pupil Center Detection Using a Curvature Algorithm, *Computer Methods and Programs in Biomedicine*, 59:145-147, 1999.
 [9] Frankewitsch, T., and Prokosch, U., Multimedia Explorer: Image Database, Image Proxy-Server and Search-Engine, *Proc AMIA Symp*, 765-9, 1999.
 [10] Silberschatz, A., Korth, H. and Sudarshan, S. Database System Concepts. McGraw- Hill, 2002.
 [11] Knuth, D., Sorting and Searching. Addison-Wesley, 1973.
 [12] Wirth, N., Algorithms + Data Structures = Programs. Prentice Hall, 1976.
 [13] Eisenhauer, G., Portable Self-Describing Binary Data Streams (1994). Technical Report GIT-CC-94-45, College of Computing, Georgia Institute of Technology, 1994.
 [14] Foster, K. H., Gaska, J. P., Nagler, M., and Pollen, D. A., Spatial and Temporal Frequency Selectivity of Neurones in Visual Cortical Areas V1 and V2 of the Macaque Monkey. *Journal of Physiology*, 365:331-363, 1985.
 [15] Meghini, C., Sebastiani, F., and Straccia, U., A Model of Multimedia Information Retrieval, *Journal of the ACM*, 48(5), pp.909-970, 2001.
 [16] Weber, R., Bolliger, J., Gross, T., and Schek, H.-J., Architecture of a Networked Image Search and Retrieval System, *In Eighth International Conference on Information and Knowledge Management*, Kansas City, Missouri, USA, pp. 430-441, Nov. 1999.
 [17] Ozden, B., Rastogi, R., and Silberschatz, A. Multimedia Support for Databases. *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. pp 1-11, May 1997.
 [18] Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D. and Equitz, W. *Efficient and Effective Querying by Image Content*. *J. Intell. Inf. Syst.* 3(4):231-262, 1994.
 [19] Traina, Jr., C., Traina, A.J.M., dos Santos, R.R., Senzako, E.Y. Support to Content-Based Image Query in Object-Oriented Databases. *Proceedings of the ACM Symposium on Applied Computing* February 1998.
 [20] Ogle, V.E., Stonebraker, M. Chabot: Retrieval from a Relational Database of Images. *IEEE Computer*, 28(9) pp. 40-56, September 1995.
 [21] Chaudhuri, S. and Gravano, L. Optimizing Queries over Multimedia Repositories. *In Proceedings of SIGMOD '96* (Montreal, Canada, June 1996). ACM Press, New York, 1996, pp. 91-102.
 [22] Annamalai, M., Chopra, R., Defazio, S., Mavris, S., Indexing Images in Oracle8i, *In Proceedings of SIGMOD 2000*, Dallas, TX pp.539-547.
 [23] Vaida, M.-F., Domokos, J., Oracle9i In Managing Medical Images and Multimedia Content, *International Workshop, Trends and Recent Achievements in Information Technology*, Cluj Napoca, Romania, May 2002, pp. 16 - 18,