Journal of Integrative Neuroscience, Vol. 3, No. 4 (2004) 363-378
Imperial College Press
www.icpress.co.uk

Short Communication

A RELATIONAL DATABASE APPLICATION IN SUPPORT OF INTEGRATED NEUROSCIENCE RESEARCH

IRA RUDOWSKY
Institute of Neural & Intelligent Systems
Department of Computer and Information Science
Brooklyn College of City University of New York
2900 Bedford Avenue
Brooklyn, New York, 11210 USA
rudowsky@brooklyn.cuny.edu
http://userhome.brooklyn.cuny.edu/irudowsky

OLGA KULYBA[1], MIKHAIL KUNIN[1], DMITRI OGARODNIKOV[2] AND THEODORE RAPHAN[1,2]
[1]Institute of Neural & Intelligent Systems
Department of Computer and Information Science
Brooklyn College of CUNY
Brooklyn, New York, 11210
and
[2]Department of Neurology, Mt Sinai School of Medicine
New York, New York, 10029

Received 17 May 2004
Accepted 2 July 2004

The development of relational databases have significantly improved the performance of storage, search, and retrieval functions and have made it possible for applications that perform real-time data acquisition and analysis to interact with these types of databases. The purpose of this research was to develop a user interface for interaction between a data acquisition and analysis application and a relational database using the Oracle9i system. The overall system was designed to have an indexing capability that threads into the data acquisition and analysis programs. Tables were designed and relations within the database for indexing the files and information contained within the file were established. The system provides retrieval capabilities over a broad range of media, including analog, event, and video data types. The system's ability to interact with a data capturing program at the time of the experiment to create both multimedia files as well as the meta-data entries in the relational database avoids manual entries in the database and ensures data integrity and completeness for further interaction with the data by analysis applications.

Keywords: neuroinformatics; neuroscience; relational database; multimedia data; Oracle; data integrity; security.

1. Introduction

Studies of neural function have resulted in the generation of large amounts of data generated in different laboratories and addressing different questions. These data potentially could be utilized in a shared environment to broaden understanding of brain function and lead to innovations in neuroscience, informatics and treatment of brain disorders [1, 3, 9, 11, 12, 16, 17, 29]. Recent developments of database systems that store information related to neuroscience applications have dealt with the problem of integrating and storing heterogeneous data using an Entity-attribute-value database approach [13, 14, 18-21]. In this approach, the data are organized to associate an entity with an attribute and a value as a row in the database table. This enables a wide variety of disparate data to be embedded within the database. However, because a consistent dictionary of data entry is lacking in this approach, there are redundant data entries and system querying is more complicated than when data are stored conventionally. These systems have also not dealt with the problem of how the database would interact with application programs that perform real time data acquisition and analysis. The aforementioned designs also do not deal with the problem of data security. These latter issues are an important aspect of database design for neuroscience research, since each laboratory already has its own applications for analysis, but needs an interface to these applications, so that the appropriate data can be queried, retrieved and analyzed. The issue of data security is an important issue when data are to be shared across research laboratories. No such global design or its feasibility has been addressed in an organized fashion. The purpose of this study was to develop a prototype relational database based on a relatively small group of laboratories performing behavioral studies ranging from cognition to sensor-motor behavior on humans, monkeys, rabbits and mice to determine whether such an integrative tool could be efficiently implemented.

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 [5, 7]. Classical database systems are of limited utility in accessing the data in such files [15, 30], since the focus is on storage and retrieval of textual data [10, 26, 31]. Moreover, classical database design does not have the capability to interact with applications that use digital streams representing analog and video information as input. Recent work on database management has therefore focused on relational databases as a robust and fast retrieval system that can be upgraded to include video and voice [8]. The data types in relational databases now offer the possibility of including still and streaming video images, characterized as multimedia data [23]. They also are capable of interacting with data types that have high storage and bandwidth requirements and can be queried and retrieved based on content [4, 6, 22, 27]. This affords the possibility of developing interfaces that interact with user written applications.

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 [2, 28, 30]. Oracle9i also provides a component called 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.

At present, a data acquisition program creates and stores data into what we have called VMF files. The VMF program is an in-house developed system to analyze data. The data format is comprised of channels that represent stimuli and responses. The channels can be representations of analog data, that are acquired via an A/D converter or event channels that associate an event with a time of occurrence. The VMF program also contains modules, which operate on the data and perform a wide range of transformations, such as a correlation analysis, power spectral analysis, timing of events, etc. The program also has visualization capabilities so that data can be displayed as time functions. The transformed data can also be displayed in the frequency domain or as one variable against another. While the application has the ability to manipulate and analyze the data in the channels, it has no capability for storing indexing information. One of the design goals was therefore to construct a prototype relational database for connecting the VMF application software and the files on which it operates. However, developing schemes for integrating database structures with applications that include data over a wide range of formats has not been adequately addressed.

In this study we developed and implemented an Oracle9i database system, which can be accessed through a C/C++ front-end. A prototype system was developed that is also capable of running the VMF analysis application to analyze the data once a record has been identified. The interface enables access to non-textual files containing results of experiments on humans and monkeys through relational tables, which we refer to as the meta-data. A robust search capability was implemented that can handle an increasingly varied repertoire of experiments as well as user-friendly and flexible security component that can be easily maintained. The results of our work indicate that the Oracle9i database system can be utilized in interfacing to real-time as well as off-line application interfaces. Initial results on the database design have been presented [24, 25].

2. Conceptual Organization of the System

The design strategy was to link existing data acquisition and analysis programs that researchers utilize to a database that can in turn store links to the data files (Fig. 1). For example, a script has been implemented that saves metadata in the Oracle database in conjunction with data acquired and that is saved in an external file. The user can then access the analysis application through the database interface software to analyze the appropriate data. The tables in the database only contain metadata that reference the linked files. This organization allows the files themselves to contain data of any type that the

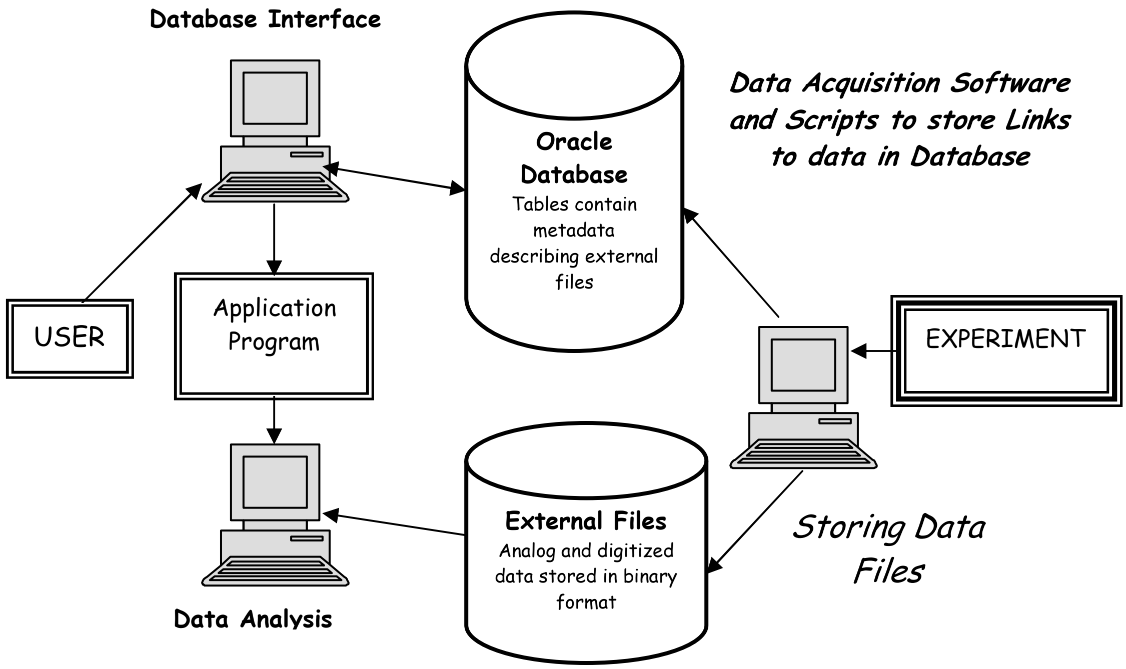

Fig. 1 Conceptual Organization of Database, Applications and Data Files Interaction

researcher has identified and the database has the capability of opening an application that operates on the files (Fig 1). Most of the metadata are in fixed fields, but there is also a comment field that is free formatted. One constraint of this design is that it is incumbent on the researcher to make the free formatted descriptor as complete as possible for appropriately accessing the files to be analyzed. While there is always the possibility of entering incomplete metadata in a free format descriptor, the advantages of such an approach in simplifying the design and avoiding numerous fields far outweighs the small probability that researchers who know their data would enter incomplete metadata descriptors. In addition, the fixed field entries add a robustness to the design and reduces the probability that free-formatted field will introduce errors due to incompleteness. The problem of quality control for data entry is best addressed empirically as individual researchers begin to utilize the system. This a topic beyond the scope of the present paper.

The interface was designed using standard GUI controls with pre-filled drop-down combo boxes. These types of interfaces are commonly used in PC interfaces so that the user can easily adapt to queries and interaction with the system. This type of design has also been used for the results tables, database updates and security. From this perspective, we consider the user interface very friendly. The interface has been prototyped using the Microsoft foundation classes, which limits the client system to run on a Windows based system. The Oracle database server is running under Linux, which has proven to be robust.

At present, the system has been implemented on a server having a 1 MHz. dual processor Pentium III motherboard, with an SCSI interface for disk access. Queries and response times are fairly quick, but no baseline test protocol has been established for estimating runtime specifications. However, Oracle optimizes the queries and we have not experienced any slow access during the testing of the system.

3. System Design

A critical component of the system design was the timely entry of the meta-data relating to each experiment and the concurrent linking of the VMF file associated with each record in the database. In Figure 2, the two dashed lines, numbered 1 and 2, indicate the concurrent actions that are necessary for data integrity. Action number 1 is the creation of the VMF files and action number 2 is the entry of the meta-data associated with the link to each VMF file into the Oracle database. Delaying action 2 to a later time can result in a broken link between a VMF file and its corresponding meta-data record in the relational database defeating the raison d'etre of the relational database and thereby impairing search via the Data Interface Application.

The system has two critical components, written in C/C++, which we refer to as Database Interface Application (DIA). First, a graphical, user-friendly interface was designed so that it is easy to submit queries and retrieve information from the database. This interface has the capability to index metadata that describe the experiments. Second, a maintenance routine was developed so that the tables in the database could be manually updated. The user interacts with the system

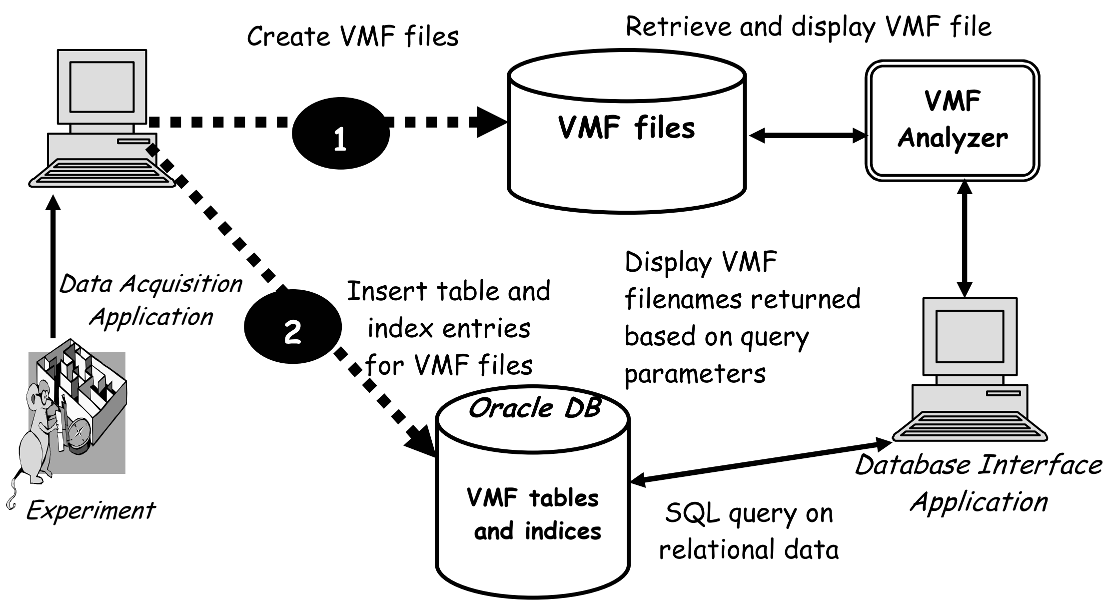

Figure 2. Complete data capture, storage and analysis system. Dashed lines indicate where automated, concurrent actions are necessary

through Text fields and pre-filled, drop down combo boxes, which designate 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

3.1. *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. 3). Seven tables were defined: SUBJECTTYPE, SUBJECTCONDITION, SUBJECT, SUBJECTGROUPID, EXPERIMENT, APPARATUS, SUBJECTCLASS (Fig. 3, Gray Headings). Each of these tables has a composite primary key (Fig. 3, Bold fields in EXPERIMENT table) or a single primary key (Fig. 3, Bold Field in SUBJECT table). 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. 3).

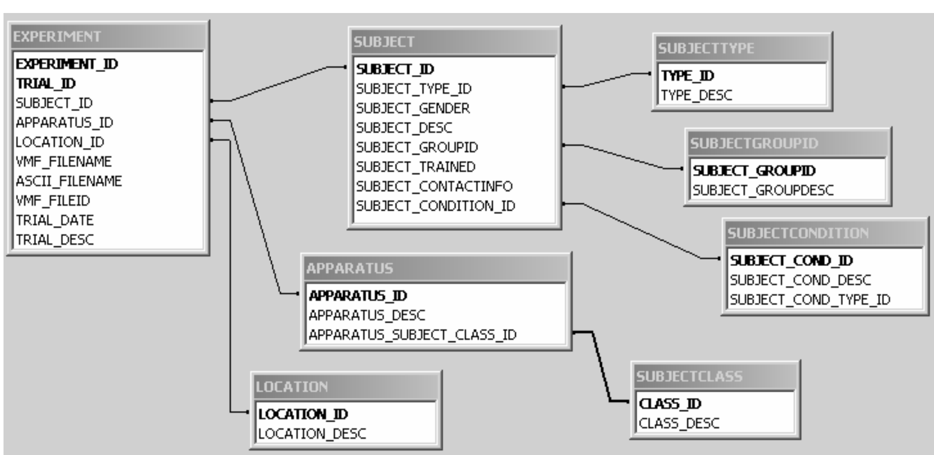

Fig. 3. Flowgraph of database tables.

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

3.2. *The Data Interface Application*

The *Data Interface Application* (DIA) provides functionality via five tabs - *Logon, Query, Result, DBManage* and *Security* each associated with one or more forms. *Logon* enables a user to access the application with the permissions granted via *Security* (see section 3.2.2), which enable the enforcement of access to subject information in accordance with privacy concerns. The *Query* tab (Fig. 4) 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 (Fig. 5). 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. Results indicate that the relational database is a flexible and easy way to interface data analysis applications to a wide range of data types

3.2.1. *Flexible Querying*

During the course of experimentation within a research project, new types of apparatus are being developed, additional types of subjects are being tested and different

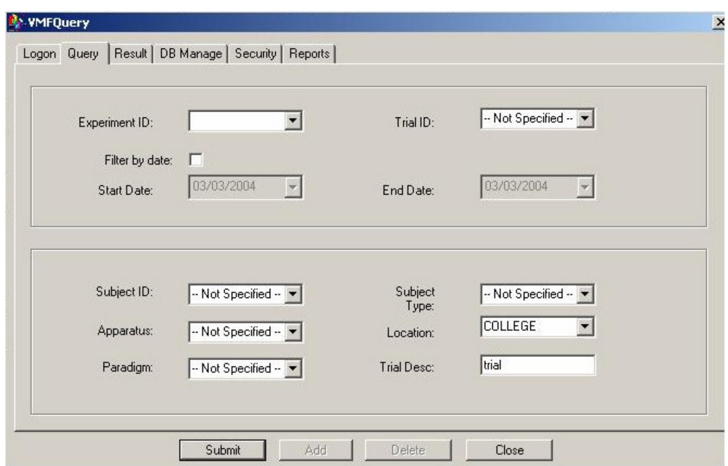

Fig 4. Data Interface Application – Query tab

types of tests are being devised. This had led to a wide variety of trial-specific information that can no longer be accommodated into the existing categories displayed on the query screen but nevertheless must be stored along with the other trial-related metadata. One approach would be to add a field within a table for each new type of test, subject and/or apparatus but this would lead to many fields, which have no entries, leading to an overly complicated user interface and sparsely populated field. The ongoing addition of new fields would also incur the cost and time of programming changes to the database and application code but in the end would be used infrequently as to not warrant the expenditures. To address these problem while maintaining a manageable number of search parameters, a free format text field, COMMENT, has been added to the EXPERIMENT table to hold trial related information that does not fit into the existing query categories. As opposed to the combo boxes that provide a drop down list of entries to choose from, the user is required to type specific text in the *Trial Desc* query text field (Figure 4) – a minor inconvenience for the results obtained. It is also possible to combine key words in searching the COMMENT field by using OR and AND connectors. The program parses the input, constructs the appropriate SQL statement and displays the returned rows in the Results tab (Figure 5). Thus, the COMMENT field together with the ability to perform free-format search on the field is a simple mechanism to provide scalability without adding overhead and complexity to the interface.

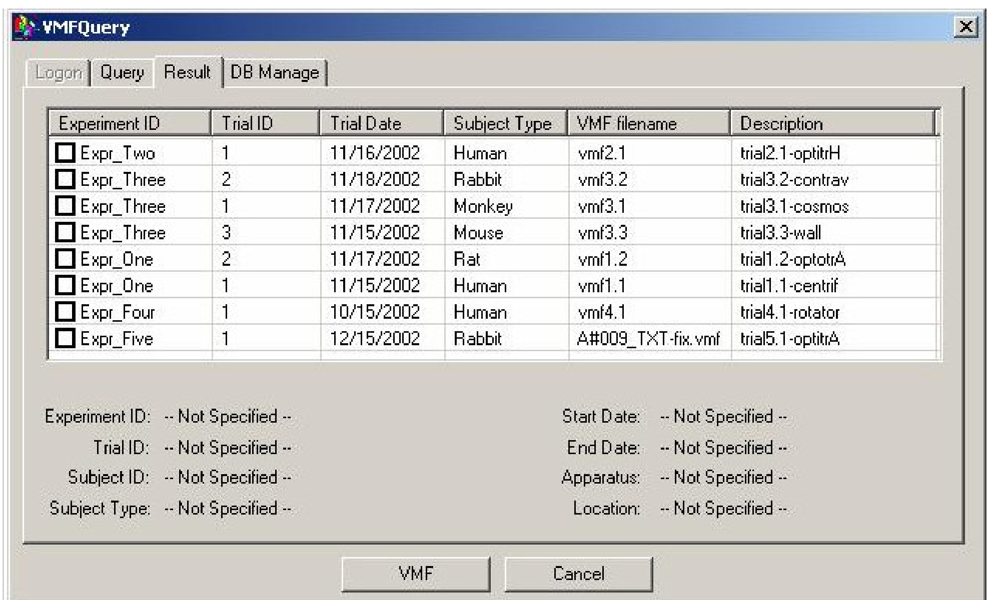

Fig 5. Data Interface Application – Results tab

3.2.2. *Flexible Security*

Oracle9i provides extensive security features but their implementation is not user friendly and would require a database administrator (DBA) to properly administer them. Security in the application is required to control access to database objects, such as tables and rows within tables as well as to restrict functionality of the user interface as well. A security model was designed and implemented [25] that employs our own tables and an interface that is reusable in other applications and extendable to other relational database systems and platforms.

Each *SecurityObject* i.e., any resource that needs to be secured (for example, a table, a particular column or a tab in our application), contains a *SecurityDescriptor*. The *SecurityDescriptor* contains two *Access Control Lists* (ACLs) - one holds user permissions and the other group permissions. An ACL entry consists of three *Access Control Entries* (ACEs), which control allowed, denied, and grantable permissions of a particular user or group. An ACL associates grantees (users and groups) to the permissions that apply to them. *SecurityObjects* are arranged in a hierarchical form so that permission settings may be inherited by a child. Every *SecurityObject* instance has a parent and zero to many children.

This logical design is implemented via five tables that have been added to the database – USERS, GROUPS, USERGROUPS, PERMISSIONS and SECURABLEOBJECTS (Fig. 6). The USERS table contains information relating to individual users of the system while the GROUPS table is used to associate an individual user or group with another group. That is, an individual user can belong to one or more groups and a group can belong to one or more other groups. This information is stored in the USERGROUPS table. An object that is to be secured has an entry in the PERMISSIONS table for each grantee (user or group) that is associated with it. SECURABLEOBJECTS stores the name and type of the objects to be secured. Each bit in the BITMASK column identifies whether a specific right is granted and the entire bitmask string of 1's and 0's defines all rights to the object for a grantee. By ANDing the bitmask of an object for which the user is a grantee together with the bitmasks of all groups the user belongs to, the rights of a user relative to a *SecurityObject* are determined. The classes and tables developed in this research provide an adaptable security infrastructure to grant detailed knowledge of individual experiments only to those individuals or groups permitted under security regulations.

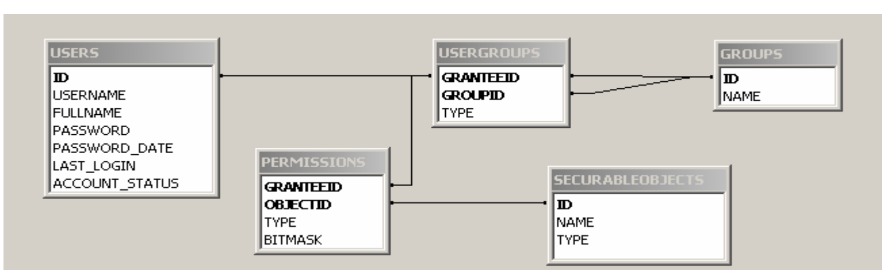

Fig 6. Security tables and their relationships

The *Security* interface is displayed in a tree hierarchy format and only select users are allowed to make changes to permissions. The *Users* branch provides a list of all individual users and their related information (Fig. 7) and the *Groups* branch shows all groups. The rights of an existing user (Fig. 8) and/or group can be updated, new users and/or groups can be added, existing users and/or groups can be deleted and individual users and groups can be added to or removed from a group within the security function (Fig. 9). This allows the user to formulate the security in a modular way by using existing groups to serve as the basis for a new group.

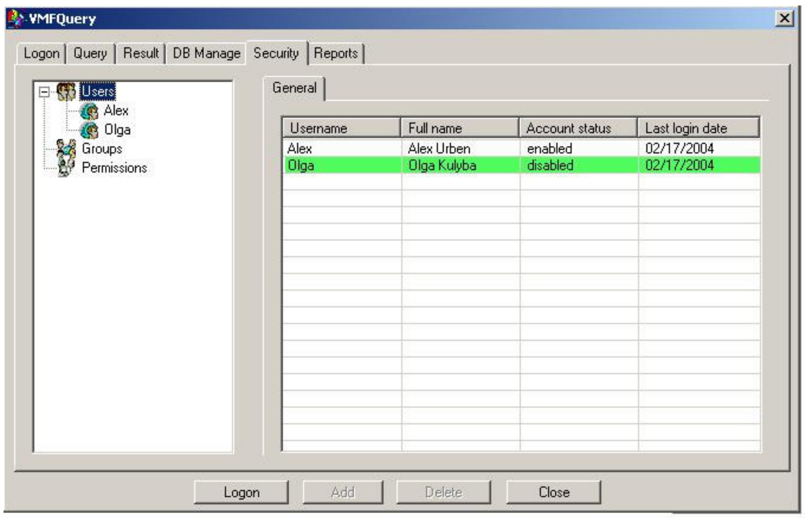

Figure 7. Security – All Users

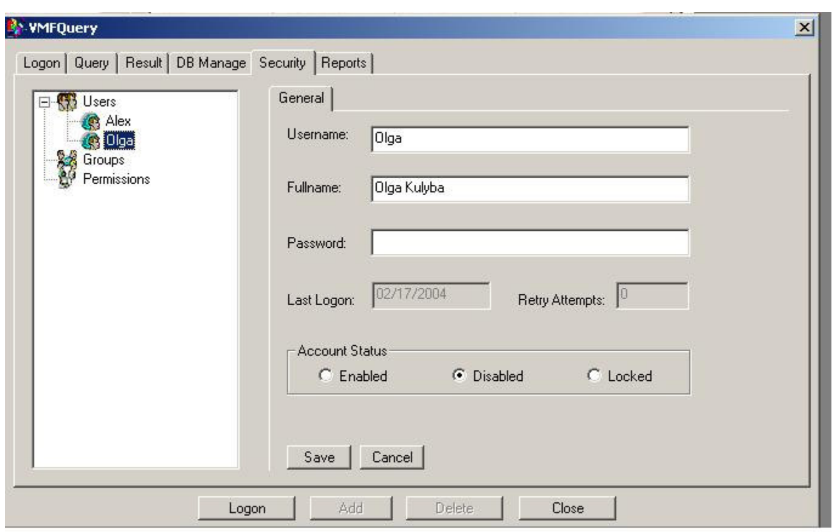

Figure 8. Security – individual user detail

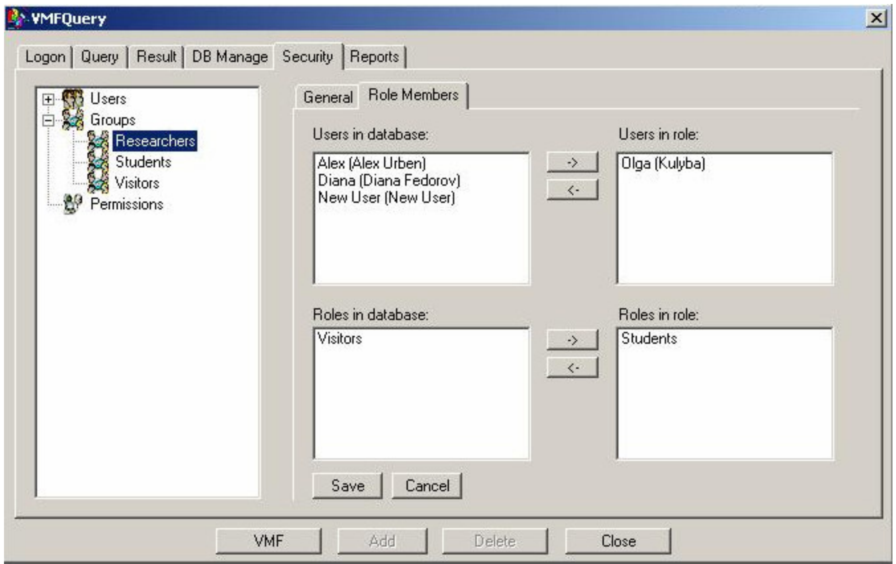

Figure 9. Assigning users and groups to a group

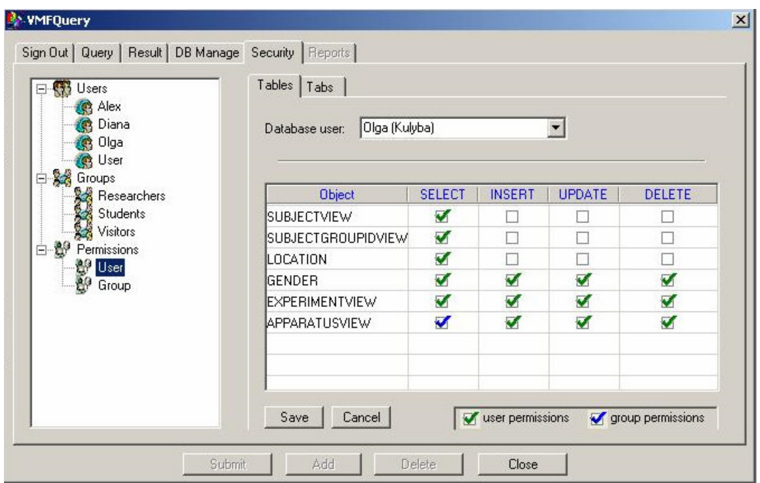

Figure 10. Permissions screen (Tables/Views)

Clicking on the User or Group branch of the Permissions node of the tree (Fig.10) enables the editing of permissions at the individual user or group level for each table and application tab (e.g., DBManage, Query). Security can be administered by a user with appropriate permission and having basic experience in traversing and manipulating a Windows-type graphical interface. By allowing individual users to belong to groups and groups to belong to other groups, the security is flexible and robust. In addition, the design is easily portable to other relational database systems as it does not employ any product specific features of the database.

4. Real-Time data Acquisition

Figure 11 shows the initial step in the data acquisition process for capturing data on body and head movements using an Optotrak System (Northern Digital, Inc.). This system has a three- camera sensor, which senses light emitted from an arrangement of infrared light emitting diodes (IR-LED's). The Optotrak user interface is comprised of a system called Toolbench (Northern Digital, Inc.) that provides the capability to enter basic information about the experiment and to begin the recording session.

For each trial of an experiment, Optotrak writes (1) an 'r' file containing spatial data with a filename in the format 'r#nnn.dat' (where nnn is a three-digit sequence number beginning with 001 and increased by one for each successive trial) and (2) a 'v' file containing analog signals and other measurements as 'v1#nnn.dat' (where the nnn is the same number as the corresponding 'r' file). This is depicted in Figure 12. ToolBench then provides the facility, via the Text Collector function, to merge multiple binary files into one ASCII file with the tab-separated data fields (Fig. 13). These ASCII files are converted to VMF format for further analysis (Fig. 14).

This provides not only a graphical display of the analog and digital data but displays a video recording that runs in parallel to the cursor location across the time axis of the graph under the user's control. Along with the video there is a separate display of a schematic of the eye position coordinated with the graph and video when eye movements are recorded.

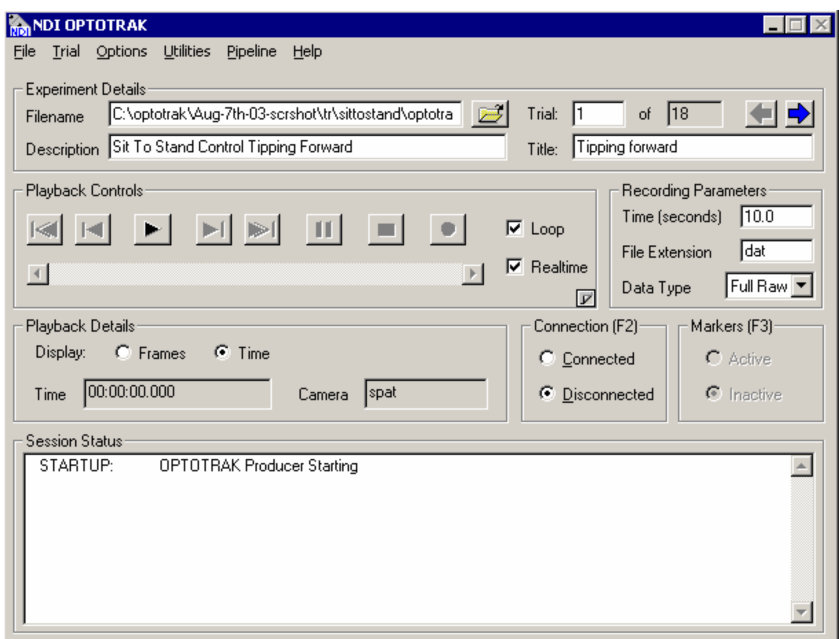

Fig 11. Optotrak acquisition. (Taken as a screen shot from an Optotrak, Northern Digital, Inc, Recording Session) .

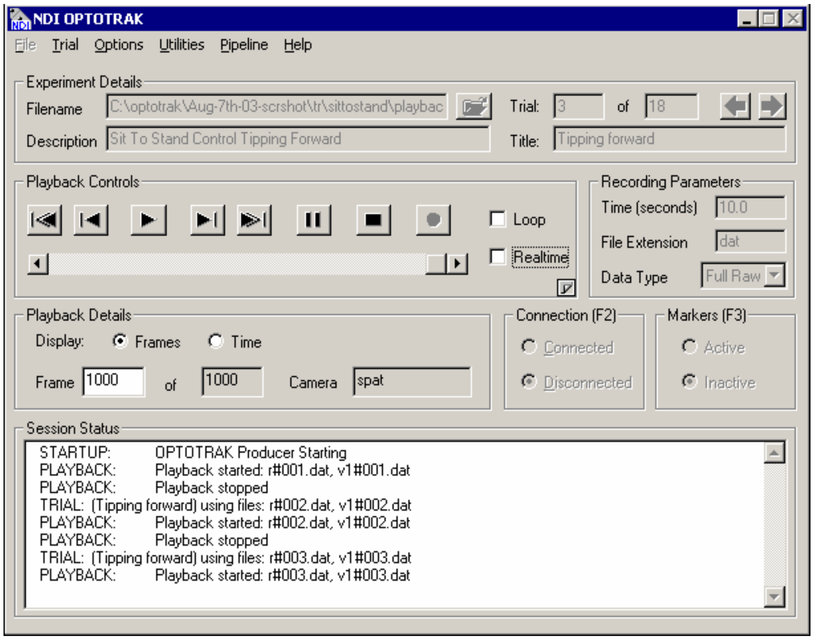

Figure 12. – Optotrak Playback (Screen shot from an Optotrak, Northern Digital, Recording Session).

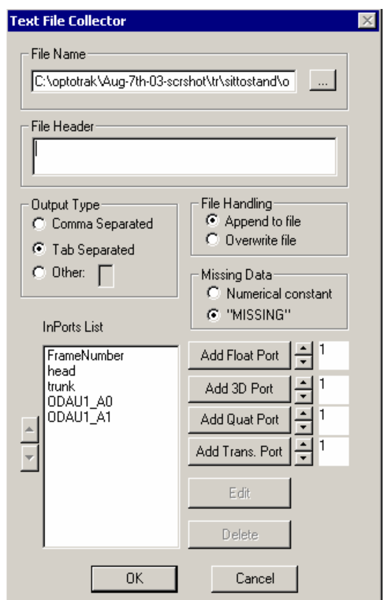

Figure 13. – Text File Collector

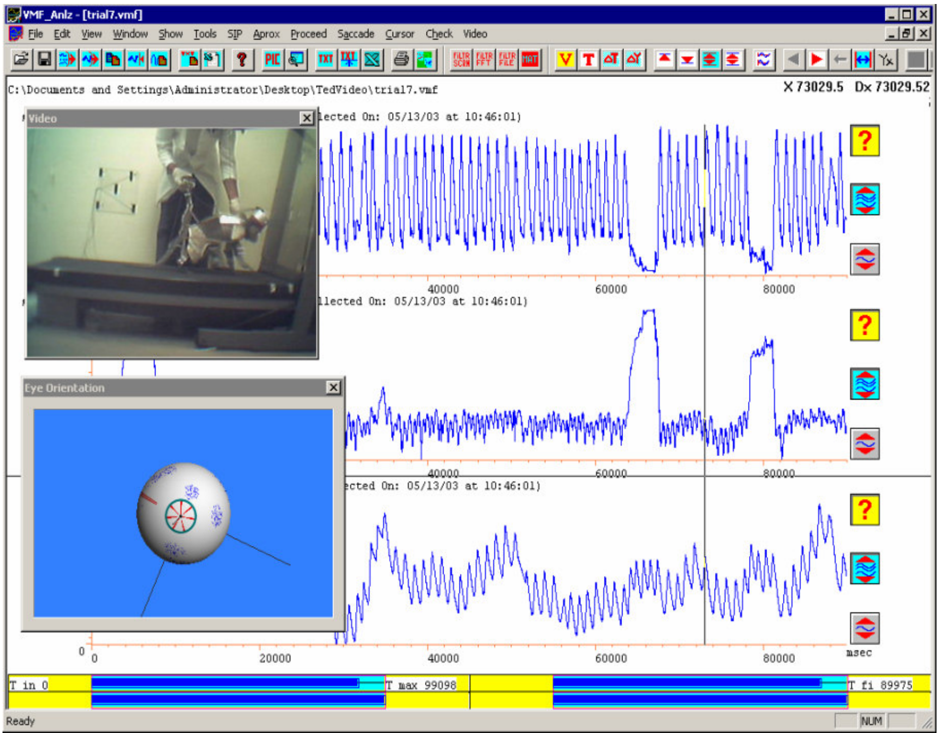

Figure 14. – VMF_Analyzer

When the experiment is designed, a file is created by the user, which includes information describing the equipment configuration as well as a description for each trial to be run during the experiment. During the experiment, a script prompts the user for the experiment id and subject id. These values are stored in an intermediary file for later use in the database update process. After all trials have completed and the ‘vmf’ files have been created, the script gathers the values o the various fields required to insert a new record in the EXPERIMENT table (EXPERIMENT_ID, TRIAL_ID, TRIAL_DATE, SUBJECT_ID, APPARATUS_ID, VMF_FILENAME, LOCATION, PARADIGM and TRIAL_DESC) from either user entered or TOOLBENCH generated information. The realization of the original system design, as described in Figure 1, has been accomplished via the script controlled integration of the data captured at the time of each trial with user entered data and subsequent automated dynamic formulation and execution of the SQL command to update of the database. Meta-data categorizing and describing the experiment and trials has been securely entered into the relational database and available for immediate use by the analysts and researchers.

5. Conclusions

The study demonstrates the feasibility of a prototype database system to support the sharing of data gathered from experiments performed in the study of cognitive and sensory-motor behavior. The integrity of the data is maintained by the automated update of the database with trial specific information immediately after the capture of real-time data has been completed and processed. The researcher is now provided with a secure database for indexing and querying the results of experiments and with an integrated link to existing analysis tools. Scalability and flexibility are naturally provided through the relational tables. Another important feature is the incorporation of a COMMENT field, which does not limit the system for including new experiments or apparatus. While this database design has been organized to support data derived from cognitive and neurophysiological data, it has the potential for being applied to a wider range of data types such images and other types of multimedia data. The graphical interface for querying and setting permissions makes the design appropriate for these kinds of extensions.

Acknowledgements
This work was supported by grants 65397-00 34 from PSC-CUNY, P30 DC05204, DC05222, EY04148 from the NIH and NASA Cooperative Agreement NCC 9-58 with the National Space Biomedical Research Institute.

References

[1] Amari S *et al.*, Neuroinformatics: the integration of shared databases and tools towards integrative neuroscience. *J Integr Neurosci*, **1**(2): 117-28, 2002.
[2] Annamalai M *et al.*, Indexing Images in Oracle8i, *Proc of SIGMOD*, Dallas, TX, 2000.
[3] Becker KG, The sharing of cDNA microarray data. *Nat Rev Neurosci*, **2**(6): 438-440, 2001.
[4] Chaudhuri S, Gravano L, Optimizing Queries over Multimedia Repositories, *Proceedings of SIGMOD '96*, Montreal, Canada, 1996.
[5] Eisenhauer G, Portable Self-Describing Binary data Streams. 1994, College of Computing, Georgia Institute of Technology.
[6] Faloutsos C *et al.*, Efficient and Effective querying by image content. *J. Intell. Inf. Syst.*, **3**(4): 231-262, 1994.
[7] Foster KH *et al.*, Spatial and temporal frequency selectivity of neurones in visual cortical areas V1 and V2 of the macaque monkey. *J. Physiol. (Lond)*, **365**: 331-363, 1985.
[8] Frankewitsch T, Prokosch U, Multimedia Explorer:Image Database, Image Proxy-Server and Search Engine, *Proc AMIA Symp*, 1999.
[9] Geschwind DH, Sharing gene expression data: an array of options. *Nat Rev Neurosci*, **2**(6): 435-8, 2001.
[10] Knuth D, *Sorting and Searching*. 2 ed. The Art of Computer Programming. Vol. 3. Reading, Mass: Addison Wesley. 765-769 1998.
[11] Koslow SH, Should the neuroscience community make a paradigm shift to sharing primary data? *Nature Neuroscience*, **7**(5): 473-481, 2000.
[12] Koslow SH, Opinion: Sharing primary data: a threat or asset to discovery? *Nat Rev Neurosci*, **3**(4): 311-3, 2002.
[13] Marenco L *et al.*, Neuronal database integration: the Senselab EAV data model. *Proc AMIA Symp*: 102-6, 1999.

[14] Marenco L *et al.*, Achieving evolvable Web-database bioscience applications using the EAV/CR framework: recent advances. *J Am Med Inform Assoc*, **10**(5): 444-53, 2003.
[15] Meghini C *et al.*, A model of Multimedia Information Retrieval. *Journal of the ACM*, **48**(5): 909-970, 2001.
[16] Miles M, Microarrays:Lost in a storm of data? *Nat Rev Neurosci*, **2**(6): 441-443, 2001.
[17] Mirnics K, Microarrays in Brain Research: The good, the bad and the ugly. *Nat Rev Neurosci*, **2**(6): 444-447, 2001.
[18] Nadkarni PM, Brandt C, Data extraction and ad hoc query of an entity-attribute-value database. *J Am Med Inform Assoc*, **5**(6): 511-527, 1998.
[19] Nadkarni PM *et al.*, WebEAV: automatic metadata-driven generation of web interfaces to entity-attribute-value databases. *J Am Med Inform Assoc*, **7**(4): 343-56, 2000.
[20] Nadkarni PM, Marenco L, Easing the transition between attribute-value databases and conventional databases for scientific data. *Proc AMIA Symp*: 483-7, 2001.
[21] Nadkarni PM *et al.*, Organization of heterogeneous scientific data using the EAV/CR representation. *J Am Med Inform Assoc*, **6**(6): 478-93, 1999.
[22] Ogle VE, Stonebraker M, Chabot: Retrieval from a Relational Database of Images. *IEEE Computer*, **28**(9): 40-56, 1995.
[23] Ozden B *et al.*, A multimedia support for databases, *Proc. 16th ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Systems*, 1997.
[24] Rudowsky I *et al.*, Relational Database Linkage of Scientific Applications and Their Data Files, *Proc of the IEEE International Workshop on Soft Computing in Industrial Applications*, 2003.
[25] Rudowsky I *et al.*, Flexible Security and Search Capability for a relational Database with Externally Linked Multimedia data Files, *Proceedings of SCI2004 - The 8th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, FL, 2004.
[26] Silberschatz A *et al.*, *Database System Concepts*. New york: McGraw Hill 2002.
[27] Traina Jr. C *et al.*, Support to Content-Based Image Query in Object- Oriented Databases, *Proc. of the ACM Symposium on Applied Computing*, 1998.
[28] Vaida M-F, Domokos J, Oracle9i in managing medical Images and Multimedia Content, *Intl Workshop, Trends and Recent Achievements in Information Technology*, Cluj Napoka, Romania, 2002.
[29] Van Horn JD *et al.*, Sharing neuroimaging studies of human cognition. *Nature Neuroscience*, **7**(5): 473-481, 2004.
[30] Weber R *et al.*, Architecture of a Networked Image Search and Retrieval Syestem, *8th Intl Conference on Information and Knowlwedge Management*, Kansas City, Missouri, 1999.
[31] Wirth N, *Algorithms + Data Structures = Programs*. Englewood Cliffs, NJ: Prentice Hall 1976.