CHAPTER 3.3

User Interaction with NeuroCore

Edriss N. Merchant,[1] David A.T. King,[2] and Jeffrey S. Grethe[3]

[1] *University of Southern California Brain Project, University of Southern California, Los Angeles, California*
[2] *Neuroscience Program, University of Southern California, Los Angeles, California*
[3] *Center for Cognitive Neuroscience, Dartmouth College, Hanover, New Hampshire*

3.3.1 Introduction

For any database to be widely accepted and used it must be an experimental tool as well as a data storage, retrieval, and analysis system. To address this issue, we have begun development of an on-line notebook, a collection of Web-based tools for the researcher. The goal of the notebook is to have a laboratory-independent set of tools for viewing, recording, and comparing data across the country and around the world. To make data easily accessible to anyone with the proper access permissions, we are using the World Wide Web (WWW) for distribution.

The original on-line notebook was developed to investigate the use of the WWW as an interface to the NeuroCore database. This original interface was written using the Informix™ Web Datablade, which enables one to create Web applications that incorporate data retrieved dynamically from an Informix database. To produce a dynamic Web page, a Web browser passes an HTML form to Webdriver, Informix's interface to the Web server. The HTML form data are passed to the Webdaemon, which requests the proper application pages from the Informix database. These application pages are then parsed, and all SQL statements embedded in these pages are executed. A new HTML page is then dynamically constructed and the results are returned to the Webdaemon, which then forwards the page to the Web server through the Webdriver interface.

The original version of the on-line notebook contained various interfaces designed for specific types of queries. As an example of the functionality of these components, two of the query interfaces will be briefly presented. First, a generic query interface is shown. Through this on-line interface (Fig. 1) any textual SQL query can be submitted to the NeuroCore database. The second interface consists of a graphical query based on an histological section contained in the database (Figs. 2- and 3). This notebook interface allows the user to search the data in the database using anatomical coordinates which are input graphically when the user selects the region or location of interest on the histological image. This original prototype version of the on-line notebook helped us choose the WWW as the delivery method for our interfaces to NeuroCore. In designing the current on-line notebook, several Web technologies have been considered, such as Java, Javascript, PERL, and the Informix Web Datablade. The prototype interface also gave us feedback from our users as to features they would like to have in future versions of the on-line notebook.

The major task in designing and building the current on-line notebook was to develop a collection of interfaces that a researcher could use in all aspects of his work. One can view the scientific process as a cycle of events (Fig. 4). Scientific experiments begin with knowledge from journal articles that is used to formulate new experiments. In order to allow researchers greater access to data in the future through on-line journals, a new Model for On-line Publications (MOP) must be developed. Once a new experiment has been formulated, the researcher must then collect new data, which will then be analyzed. This will require several tools. The first stage in using NeuroCore concerns the database itself: Can the current instance of the database store all the data that you wish to store? For example, a database used in previous experiments might allow a user to store data from single unit recordings and behavioral data. In your

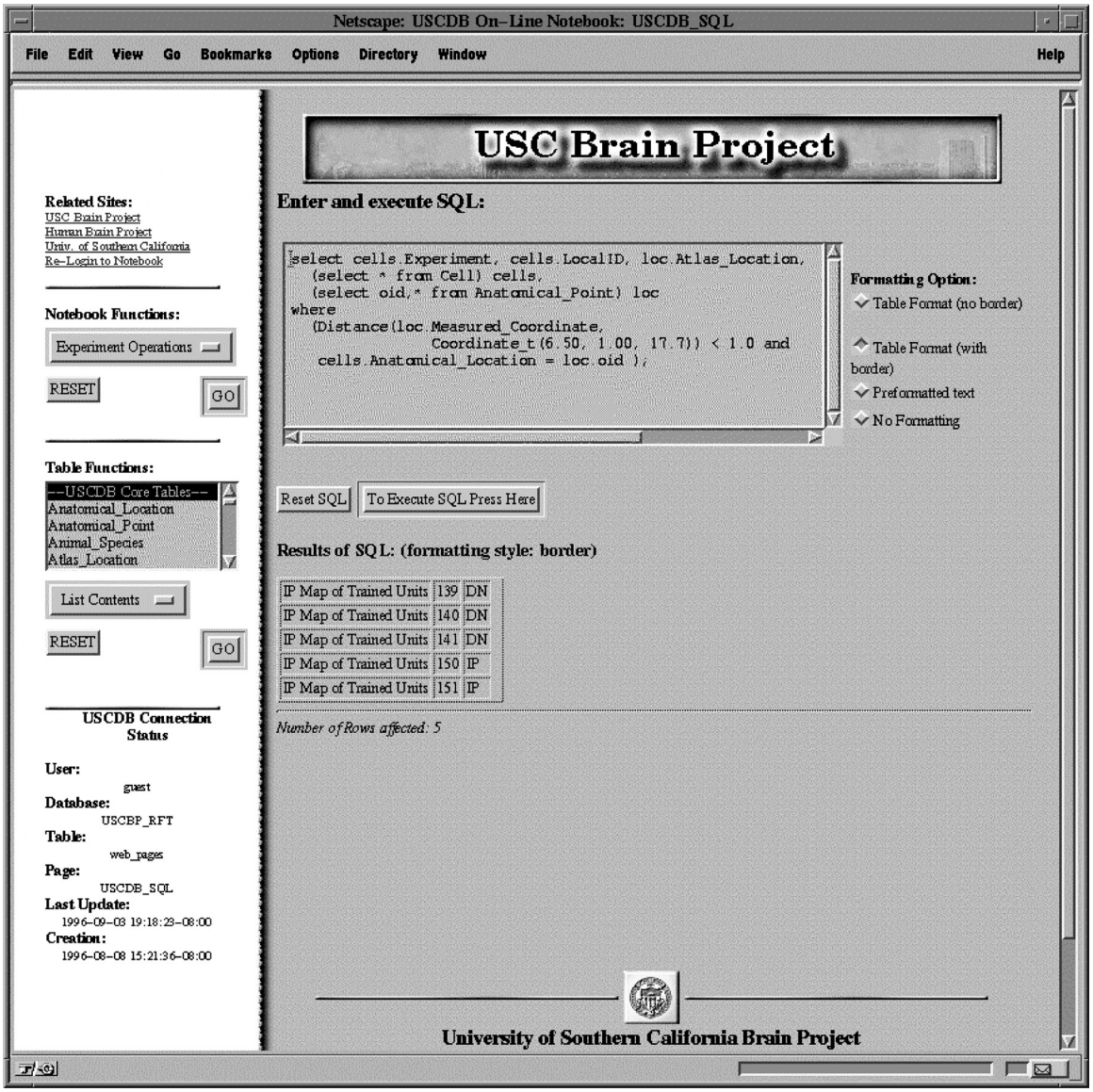

Figure 1 On-line SQL form allows users to interact with the Informix database over the Web using the standard SQL query language. The SQL query shown here searches for single unit recordings, stored in the database, that lie within a certain distance from an anatomical coordinate.

current experiment, however, you might wish to collect multiple unit recordings for which the database entries will contain slightly different information from the single unit entries already in your databases. To examine and modify the database, a schema browser (NeuroCore Schema Browser) will be necessary. Once a researcher has the correct extensions in place, a tool will be needed to enter the data being collected (Java Applet for Data Entry, or JADE). This data entry tool will need to be extensible so that different researchers can use the same foundation for their specific data entry needs. Once a researcher's data have been stored in the database, a couple of tools will be required to support the data management and analysis. First, a database browser (DB Browser) is needed to allow researchers to view and navigate the data stored in their databases. Second, an extensible analysis platform (DataMunch) must be available for users to analyze their data. The results from these analyses will then be used in producing a new journal article. For a database to be used widely by the community, it must be embedded at each stage of this life-cycle. To accomplish this, a suite of tools has been developed as part of an on-line notebook to support the various stages of this life-cycle. The five tools mentioned earlier will be discussed in the remainder of this chapter. Up-to-date information for all tools discussed in this chapter can be found online at http://www-hbp.usc.edu.

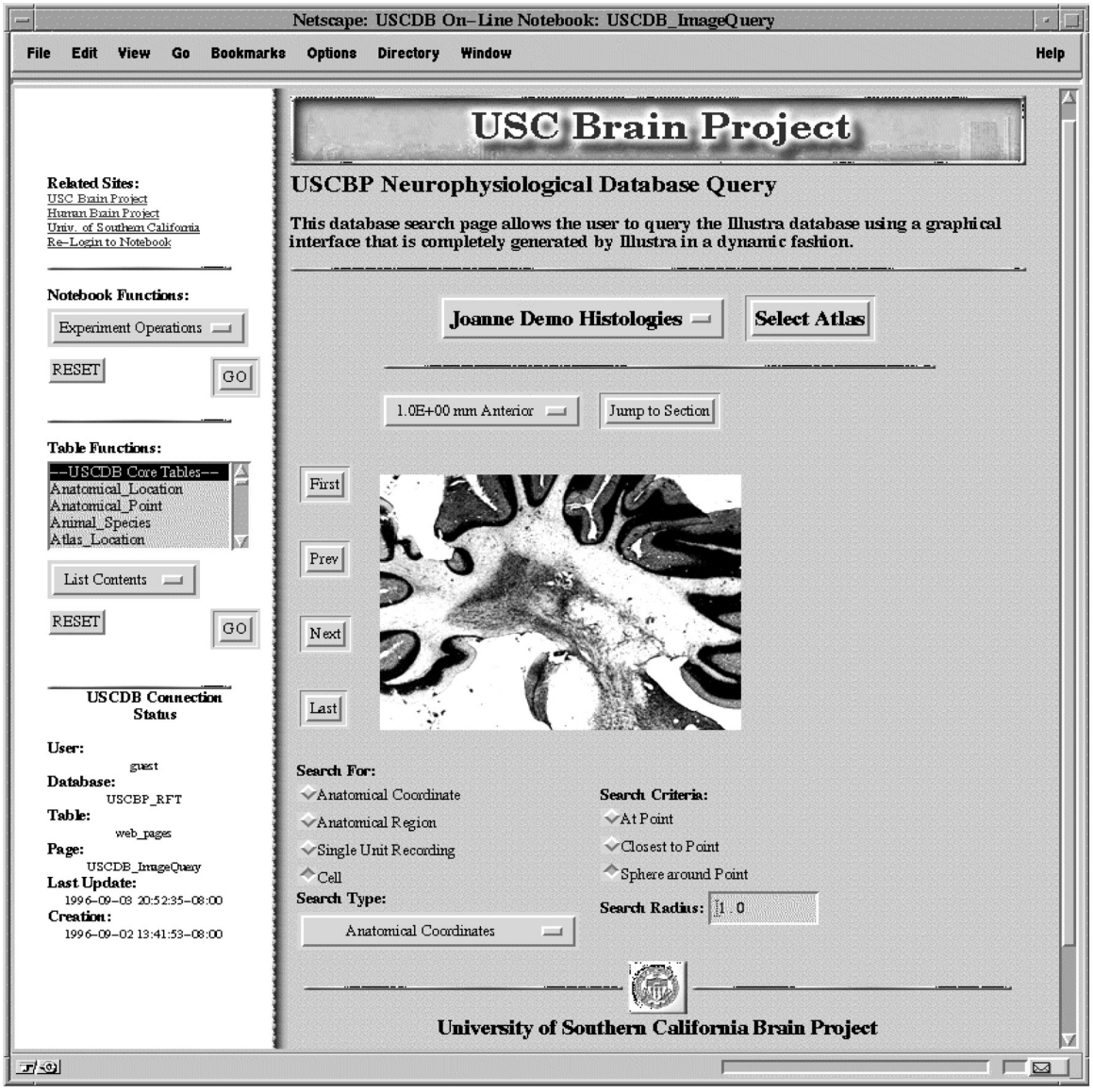

Figure 2 This atlas-based graphical query demonstrates Informix's ability to generate dynamic Web pages as well as the ability to store multimedia data such as images. Through this query, a user can select a search region by clicking directly on the anatomical image. Once the query is executed, the single units recorded in that search region are returned (Fig. 3).

3.3.2 NeuroCore Schema Browser

Introduction

The NeuroCore Schema Browser is a Web-based database client for the NeuroCore database. The goal is a fully functional database client that can be run from a Web browser and can access, retrieve, and manipulate both data and metadata on the USC Brain Project's NeuroCore database architecture. The schema browser allows a user to view the table structure of the NeuroCore database in a hierarchical tree format. The browser interface contains four sections (Fig. 5). The menu allows users to login/logout and change databases. Tables of the NeuroCore database are displayed in the bottom left menu portion of the Schema Browser in a folder tree format, similar to a standard directory tree. By clicking to expand the folder, the child or extended tables are displayed for that particular table. When one clicks on the name of a table, the table information (such as column names, description, data type, etc.) is displayed in the top right portion of the Schema Browser.

Background/Evolution

The Schema Browser was initially developed as a joint venture between the University of Southern California (USC) Integrated Media Systems Center (IMSC) and the USC Brain Project. The requirements and design reports were developed as part of this collaboration, whereas the development of the final tool itself has been conducted by the USCBP.

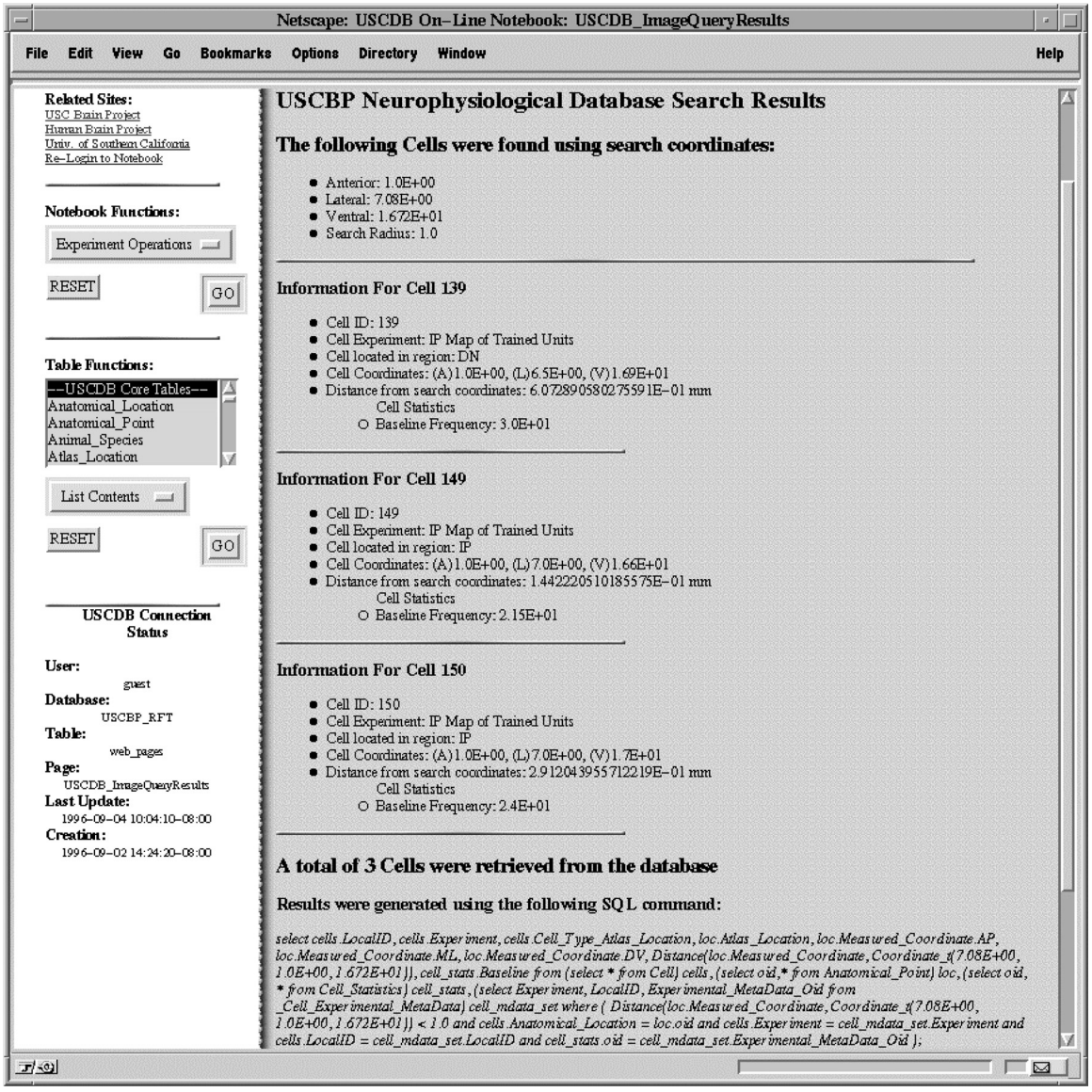

Figure 3 Textual results of graphical query provide information from the cells returned by the query generated in Fig. 2. The actual SQL query used to retrieve the data can be found at the bottom of the captured screen.

Implementation

The Schema Browser currently uses Java and JDBC to display the contents of the database hierarchy. The hierarchy and table information are dynamically loaded from the NeuroCore metadata tables, which the Web browser displays in a hierarchical folder tree format. The original version of the Schema Browser only allowed a user to view metadata associated with a NeuroCore database (i.e., the information regarding the tables and columns that comprise the database). The current release allows users to work directly with the database metadata stored in NeuroCore. This involves being able to view and extend the schema of the database. Because Informix is an object-relational database, this involves not only displaying the tables but also being able to view and manipulate the inheritance hierarchy.

Future Work

One of the most important aspects of the schema browser is the ability to modify the NeuroCore schema. Currently, users are only allowed to add new extension tables to the database through the interface. In future releases, to enter a new schema in the database (e.g., for a new experiment from a particular laboratory), the researcher will have the option of modifying an existing extension set or creating a new extension set. An extension set is a collection of table and datatype definitions that extend the NeuroCore database for a specific experiment or set of experiments. Modification of these extension sets will allow the researcher to save time by using pre-existing schema elements and will allow the researcher to build upon an existing schema. By creating a new set of database extensions, the researcher is able to

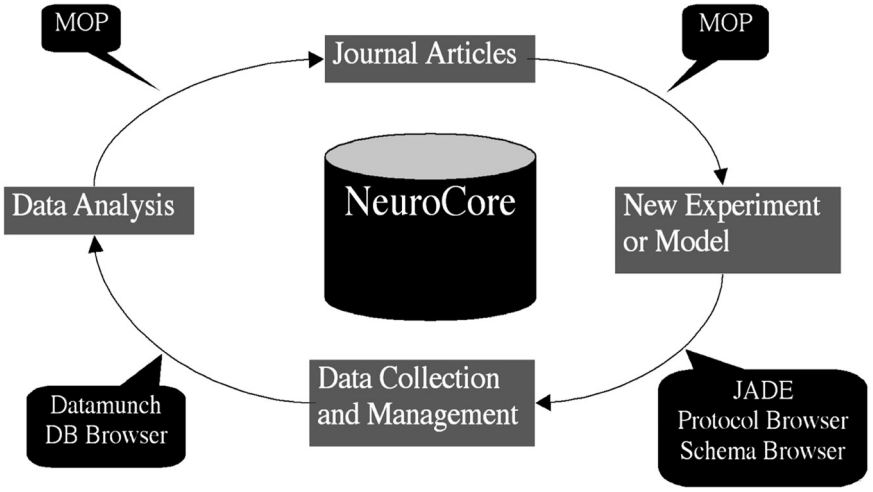

Figure 4 The scientific life-cycle. Scientific experiments begin with knowledge from journal articles that is used to formulate a new experiment. The researcher must then collect new data, which will then be analyzed. The results from these analyses will then be used to produce a new journal article. For a database to be used widely by the community, it must be embedded at each stage of this life-cycle. To accomplish this, various tools must be developed as part of an on-line notebook to support the various stages of this life-cycle.

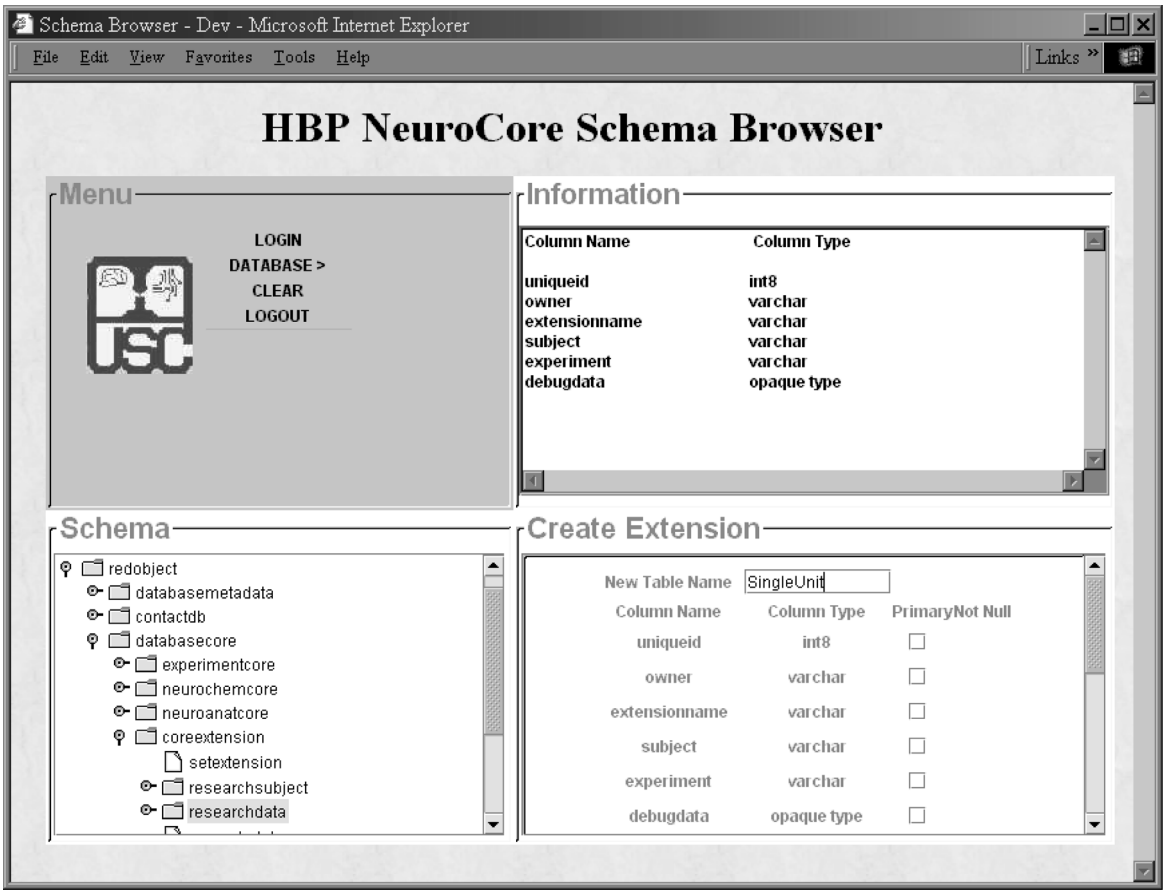

Figure 5 The Schema Browser interface. The bottom left frame shows the table hierarchy in tree format, while the right frame displays the description of the table chosen (top) as well as a form (bottom) to extend the database using the current table as a parent.

extend the NeuroCore schema to customize the database to meet the needs of his laboratory (see Chapter 3.1 for more information on how NeuroCore can be extended for various experiments). In addition to the ability to work with the database schema, the browser will also have to be able to work with objects and abstract data types created in an object-oriented fashion.

3.3.3 Java Applet for Data Entry (JADE)

Introduction

JADE is a Web-enabled tool that allows a neuroscience researcher to enter experiment data directly into a NeuroCore database. JADE is designed specifically for the NeuroCore database schema and offers the same flexibility and extendibility as the NeuroCore design. Like the NeuroCore database schema, JADE is designed to contain certain information elements that are common to all laboratory experiments, for example, all the descriptive information regarding an experiment (i.e., experiment name, experimenter, etc.). JADE can then be extended to meet the needs of a particular laboratory. Researchers can enter specific data regarding their particular experimental paradigms and conditions. One laboratory may be conducting an *in vitro* experiment where the application of various chemical agonists is required, whereas another laboratory may be recording neuronal unit data during a behavioral paradigm using various stimuli (see Chapter 3.1). It is therefore important that JADE be extendible in order to accommodate the needs of a particular laboratory similar to NeuroCore.

Figure 6 JADE three-panel interface: Menu (left), Body (upper right), and Messages (lower right). The Body displays the entry labels and entry fields that are dynamically retrieved from the laboratory database.

The interface for JADE (Fig. 6) consists of three different sections. The first section is the Menu, which contains the options that the user has to work with. The Body section contains the forms with which the user will most often interact; this section contains the labels and text fields for entering data. The third section of the JADE interface is the Messages section, which will send messages to users based upon their actions. JADE has been successfully used by researchers at USC to enter legacy data from traditional paper-based experimental notebooks.

Background/Evolution

The main design consideration for JADE was that the interface needed to have an extendible architecture. In the early development of JADE, interfaces for different laboratories were developed using a specific template for that particular laboratory. The templates were developed by identifying the specific information necessary to be entered for each experiment. The initial development was done for experiments where electrophysiological recordings were obtained (see Chapter 3.1). The information needed for JADE was extracted from descriptor files used by researchers to describe their data for a non-graphical data-entry tool that was being used to upload their data into a NeuroCore database.

The original JADE interface led a researcher through all the questions necessary to create the descriptor files they had been generating. After all the information was received, JADE produced SQL statements for entry into the database. The complete set of SQL statements was then transformed into a single transaction file so that this file could be sent to the database as a single processing block. However, one problem with this first version of JADE was that interfaces for a new laboratory experiment were not easily added to the base architecture. The new experiment would need to have a new template created, and JADE would have to be modified to accommodate this new template. A solution to this was to let JADE query the table definitions of the extension set that constituted a certain experiment. This would allow JADE to be able to produce dynamic interfaces based on the metadata describing the actual tables in an extended NeuroCore database; therefore, JADE must have connectivity to the NeuroCore database for data entry as well as information retrieval of database metadata for the generation of these dynamic interfaces. This more generalized version of JADE is currently under development and is nearing completion.

Implementation

In addition to the metadata tables in the NeuroCore database (see Chapter 3.2, Fig. 6) used to construct dynamic query interfaces, some JADE-specific tables were necessary. JADE would use these tables to build the user prompts accordingly. Because only the data-entry flow for each experiment varies, the core program retrieving and submitting data would remain unchanged. The critical step to create this generic JADE version was to identify the types of database access required by JADE during data entry. The procedure used by JADE to dynamically produce an interface depends on the type of data being entered. More specifically, whether the data are being entered into a single table or multiple tables and whether the data consist of a single entry or multiple entries. In order to accommodate these varying data-entry tasks, additional metadata tables specific to JADE had to be added to NeuroCore. In addition to the detailed description of where data must be entered throughout the database, the order in which data are entered is also extremely important. During the data-entry process, there are certain values that must be defined in describing the experiment before other information can be entered (e.g., to describe the protocol for individual experimental trials, one first must know how many trials there are). With the addition of JADE-specific metadata tables, NeuroCore and JADE have been able to produce dynamic data-entry forms.

Future Work

The current version of JADE has been used for data entry of legacy experiments in laboratories involved with the USC Brain Project. Once the initial design, implementation, and testing of JADE have been completed, further work on JADE has been planned to improve its capabilities. Based upon feedback from users entering data into their own laboratories' NeuroCore databases, a number of future enhancements have been identified:

- Streamline the automatically generated input forms to minimize the display of duplicate information.
- Enhance the user interface when entry fields reference values from other tables.
- Build an interface to populate the tables that JADE queries for building the prompts. Currently, the information is entered manually by a database administrator. A user interface would help administrators to view and update information for JADE.

3.3.4 Database Browser

Introduction

The Database Browser offers a neuroscience researcher the ability to view, browse, query, and display results of an experiment stored in the NeuroCore database. The interface was designed for the NeuroCore database schema and offers the same flexibility and extendibility as the NeuroCore design. By using the NeuroCore schema as a foundation, any database built

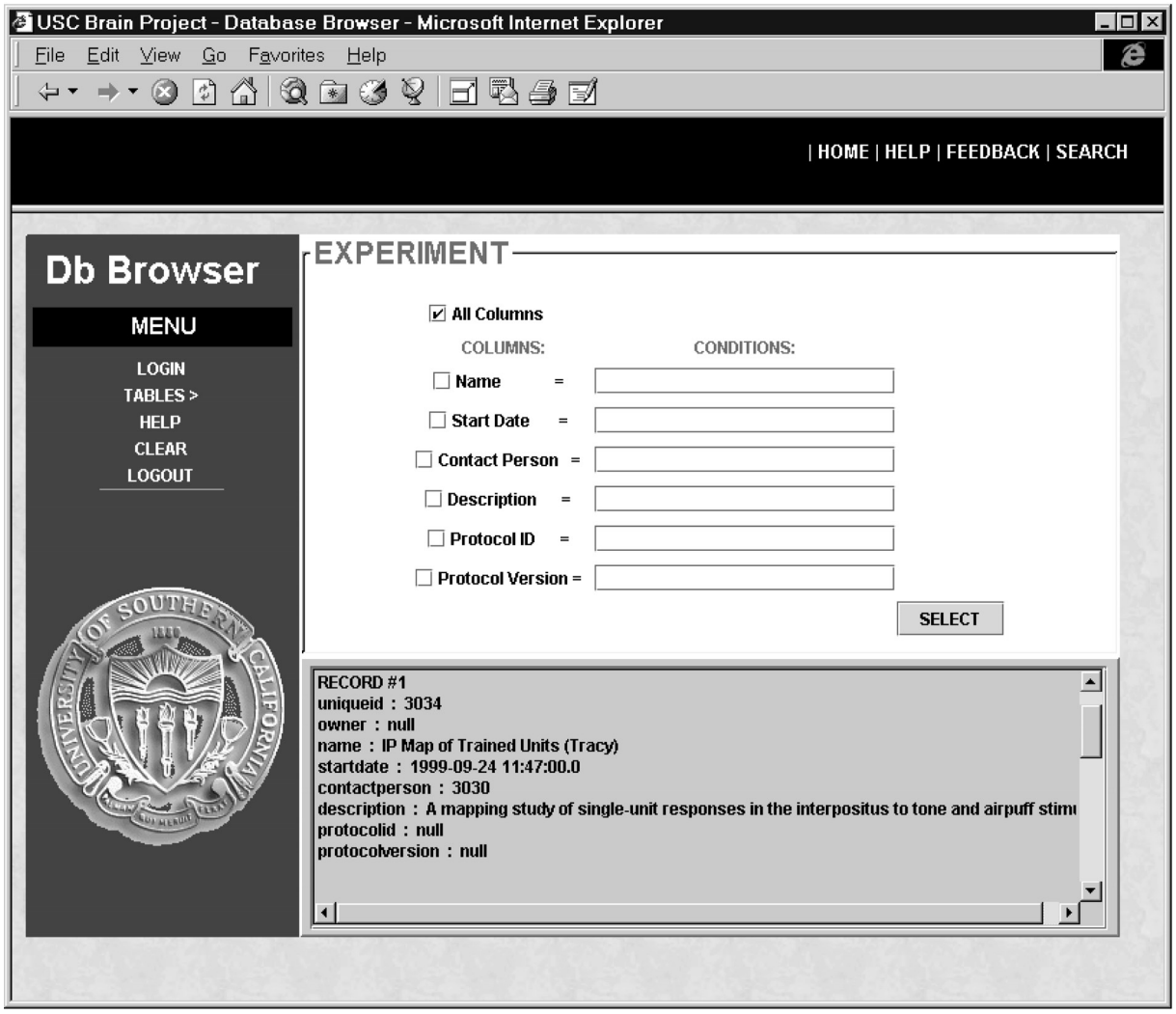

Figure 7 Database Browser three-panel interface: Menu (left), Body (upper right), and Messages (lower right). The Body displays the Columns and Search Criteria Text fields, while the Messages/Results section displays the results of the query performed.

using NeuroCore can use the Database Browser, as all NeuroCore databases contain the same core information and structure.

The user interface for the Database Browser consists of three sections (Fig. 7). The left-hand portion is the Menu for the Database Browser. The researcher has the options to Login, Access Table pages, get Help, Clear the contents of the other two sections, or Logout. The upper right-hand portion is the Body of the user interface, which changes based on the user's actions and is used to enter information or view the database contents. The lower right portion of the page, the Messages/Results section, is used to interact with the user. It will display information or instructions based upon the user's actions.

In order to facilitate searching, a user selects a particular table to search. The user selects the table name from the menu and is shown the columns for that table. The user is given a choice of checkboxes and text fields in two separate sections displayed side by side (Fig. 7). The section on the left contains a list of the columns that the user may select (with checkboxes) for the search results. For example, in the Experiment table, a user selecting the "Name" checkbox will be shown only the names of the experiments in the database. By the same method, a user choosing the "Name" and "Contact Person" checkboxes will be returned the name and contact person for all experiments stored in the database.

The section to the right contains a number of text boxes which correspond to a column of the table. These textboxes are used to allow a user to enter information or keywords for a search query. For example, using the same Experiment table, a user entering "Michael Arbib" in the text box corresponding to the Contact Person column will be shown the experiments in the database for which "Michael Arbib" is assigned as the Contact Person. The text box feature allows the user to narrow the search results on a table.

Another feature incorporated into the Database Browser is a specialized query selection. A particular laboratory may need a certain query performed

Figure 8 Specialized query result: This query returns the cell ID and corresponding trial and subject ID number for a recorded cell. The query is specialized for the needs of the USC Thompson Laboratory.

occasionally based upon the data stored in the database. For example, one query that has been incorporated is to identify the cells that are recorded during an experiment. The user who selects this query from the menu and selects an experiment from the database is then returned a listing of all the cells and their corresponding trial for that experiment (Fig. 8).

Implementation

The Database Browser was created initially as an interface so that researchers could verify their data were correctly stored within the NeuroCore database. A lot of work on an external interface had already been completed involving JADE so the Database Browser was built on this foundation. The Database Browser is a Java Applet that connects to the Informix database via JDBC.

Future Work

The initial development of the Database Browser has been completed, and further work has been identified to improve the tool. The program is being enhanced to dynamically display the table information and the necessary checkboxes and text input fields based on the information stored in the NeuroCore database metadata tables (see Chapter 3.1, Fig. 6). Currently, the Database Browser stores and retrieves the table information within its program. In order to add a new table for querying, the program must be changed to accommodate the new table. The same principle for storing the table information within the database and retrieving that information to build the interface that is incorporated in JADE needs to be applied to the Database Browser. This feature will improve the maintainability of the tool. Another feature that has been identified for future work is the dynamic retrieval of specialized queries for a particular laboratory. The query to retrieve cells corresponding to an experiment may not be used by all laboratories. In addition, new specialized queries may need to be added to the Database Browser. The queries would be stored within the database and could then be retrieved at a later time by the Database Browser when needed by a user.

3.3.5 DataMunch

Introduction

DataMunch is a free, open-source-code application that runs in the Matlab™ scripting language. These routines are used primarily to analyze two types of data:

- "Spike" data, that is, extracellular single or multiple-unit waveforms, spike times, or bin values obtained from single or multiple cells
- Continuously recorded behavioral data, such as (the magnitude of) eyeblink or limb movement

DataMunch (King and Tracy, 1999) was designed to analyze data from experiments using a classical eyeblink conditioning paradigm (see Chapter 3.1). The analysis program itself works in batch mode, allowing the user to define what data should be examined and what operations should be performed on the selected data.

A set of analysis records created from a collection of data is called an "analysis set." One is able to search each analysis set DataMunch creates with various tools in order to find collections of cells that match whatever combination of statistical criteria were defined. These selected data can then be analyzed and graphed in various ways:

- Plots of the distribution of spike-train onset times (or post-stimulus firing rates or response types, etc.) for all cells having strong correlations with behavioral onset times
- Histograms for each cell with this (or some other) combination of measurements
- Scatterplots of one statistic against another for all cells selected by the search parameters

In summary, DataMunch can take a set of data and produce a set of analysis records. It provides the user with a great deal of control over how data are analyzed as well as how they appear when plotted (see Fig. 9). A researcher can easily search through an analysis set with a convenient search tool to test hypotheses and, based on the results of this search, can create new plots that summarize part or all of the analysis set.

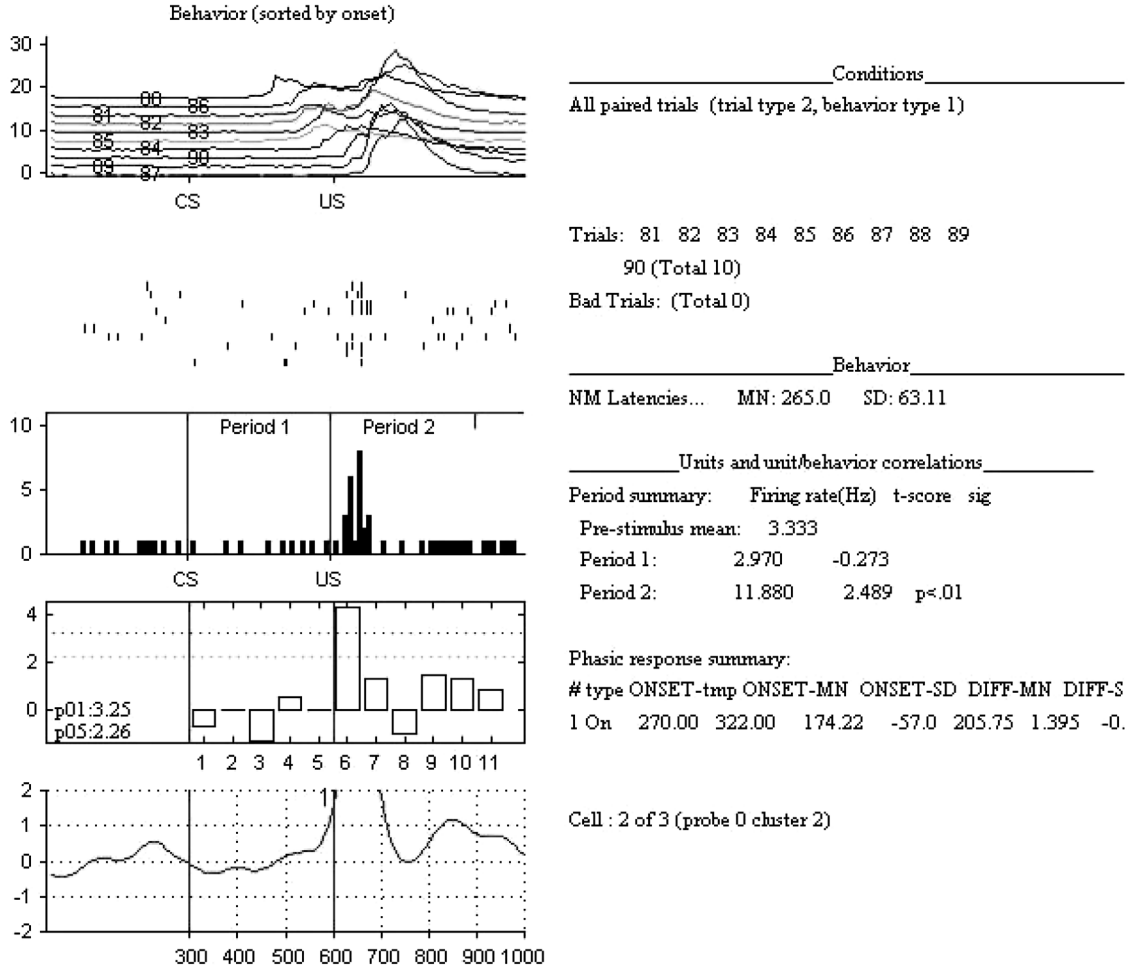

Figure 9 DataMunch: A typical example of output from an analysis of data from a classical conditioning experiment.

Background and Evolution

DataMunch is the data analysis package that members of the Thompson Laboratory have developed to analyze single- and multiple-cell spike data and behavioral (eyeblink) data. It was originally designed to recognize several popular file types, plot behavioral curves, spike rasters, summed histograms, and normalized firing rates for each cell. The program also calculates a set of statistics and displays them both within the plot and also in a separate spreadsheet suitable for further analysis. DataMunch was originally designed to work "locally," analyzing files on a single machine; however, current efforts are expanding DataMunch's capabilities so that it will be able to connect directly with a NeuroCore database and ultimately allow a convenient user interface through the Web. This functionality is being added using database and Web connectivity toolkits developed specifically for Matlab.

Future Work

There are two major efforts currently underway to connect DataMunch to the NeuroCore time-series database (for information on the time-series database, please see Chapter 3.1): data transfer between Matlab and DataMunch and an enhanced querying tool that interfaces DataMunch with the database. The querying tool will allow users to create advanced search statements by choosing variables such as response type (increases vs. decreases in firing rate), response timing (onset of neural responses), etc. For each variable, the user can enter a value or range of values to search for, and cells that fit this criteria are returned by the database for plotting and statistical analysis. Choices are also offered as to what kind of plots should be displayed and what should be done with the statistics returned.

3.3.6 Discussion

The development of the tools comprising the on-line notebook is an ongoing process. With each successive release, feedback from users highlights new functionality that can be incorporated within this framework. One such requested addition is the addition of a graphical way to reconstruct the protocol for a given experiment,

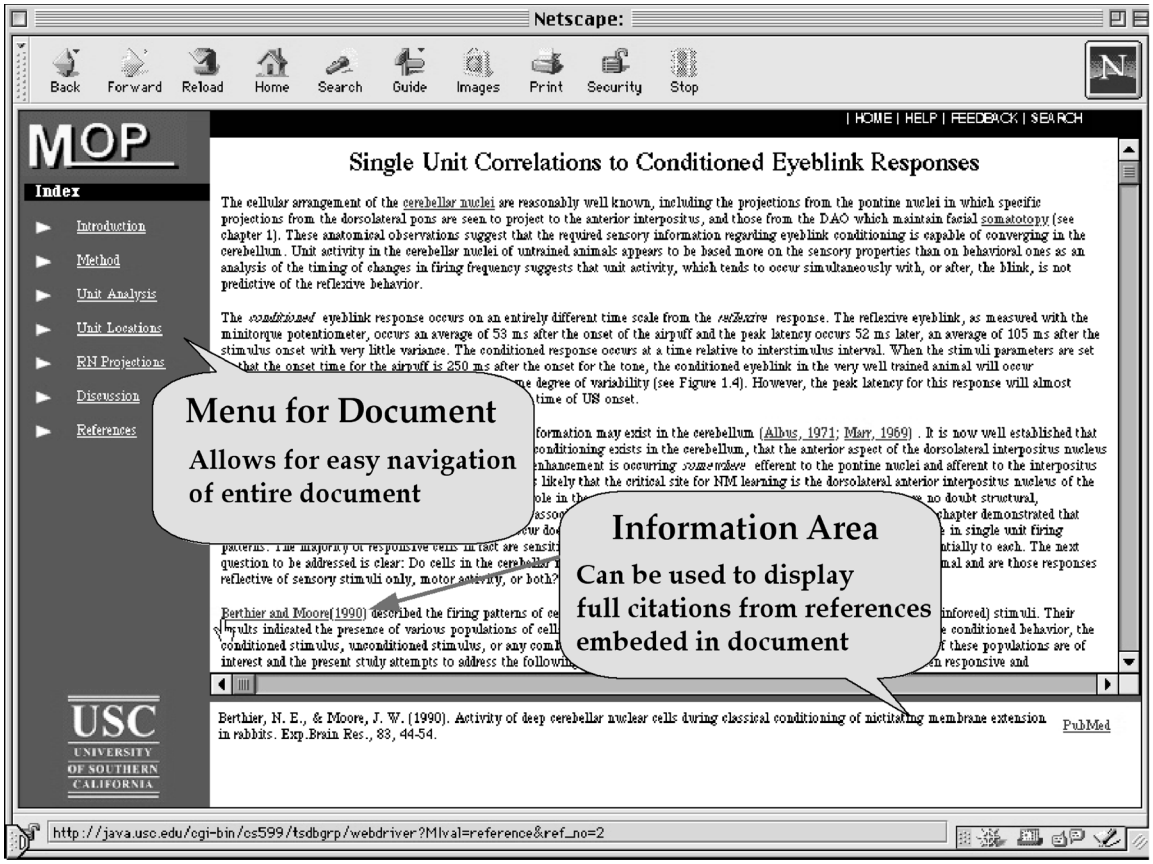

Figure 10 A prototype on-line journal interface. This interface uses features of HTML and Java to enhance the user interface for a journal article. Here, we have taken a chapter from a thesis that contains the data we have stored in the database that was developed (Tracy, 1995). This document has a navigation bar on the left for easy navigation and an information area on the bottom which, in this example, is giving the user information concerning a specific reference in the document. This allows the user to view select information without leaving his current view.

session, block, or trial. This Protocol Viewer would allow a user to graphically view protocols related to the data collected, to edit protocols, and to create new protocols. The Protocol Viewer is currently in the design stage of development; however, there is one aspect of a researcher's interaction with an on-line database that is of critical importance to the success of any scientific database, namely, the interaction between published journal articles, the data they contain, and the databases housing this data.

A scientist's life revolves around publication (Fig. 4). When designing a new experiment, many articles are read for background and information on experimental procedures. The scientist must then collect, store, and analyze all his data. These data will then be published as a journal article, which completes the cycle. Currently, the NeuroCore database and on-line notebook will allow researchers to collect and store, retrieve, and analyze their data. However, a critically important aspect of this cycle is the journal article itself. Science revolves around the publication and reading of peer-reviewed journal articles. With many journals going on-line, it is imperative that the databases we construct are able to interface with the on-line journal of the future. To this end, we have developed MOP (Model of On-line Publishing) as a prototype of such an interface. MOP highlights some important features that should be included in tomorrow's on-line journals.

The Concept of an Embedded Type

Each journal article contains many types of information (e.g., images, graphs, references). For each embedded type a set of functionality needs to be defined so that users can interact with these embedded objects. For example, a citation in the article, when selected, could display the full reference in another window without affecting the presentation of the article. A user could then interact with this reference independently of the article itself. Another example would be a graph containing various statistical data. In this instance, selecting the object would provide the user with a description of the graph object as well as specific menu options related to that graph object (Fig. 11). However, by activating the object (i.e., double-clicking the object), one could launch a separate window where users could interact directly with the graph and the data contained within it (Fig. 12).

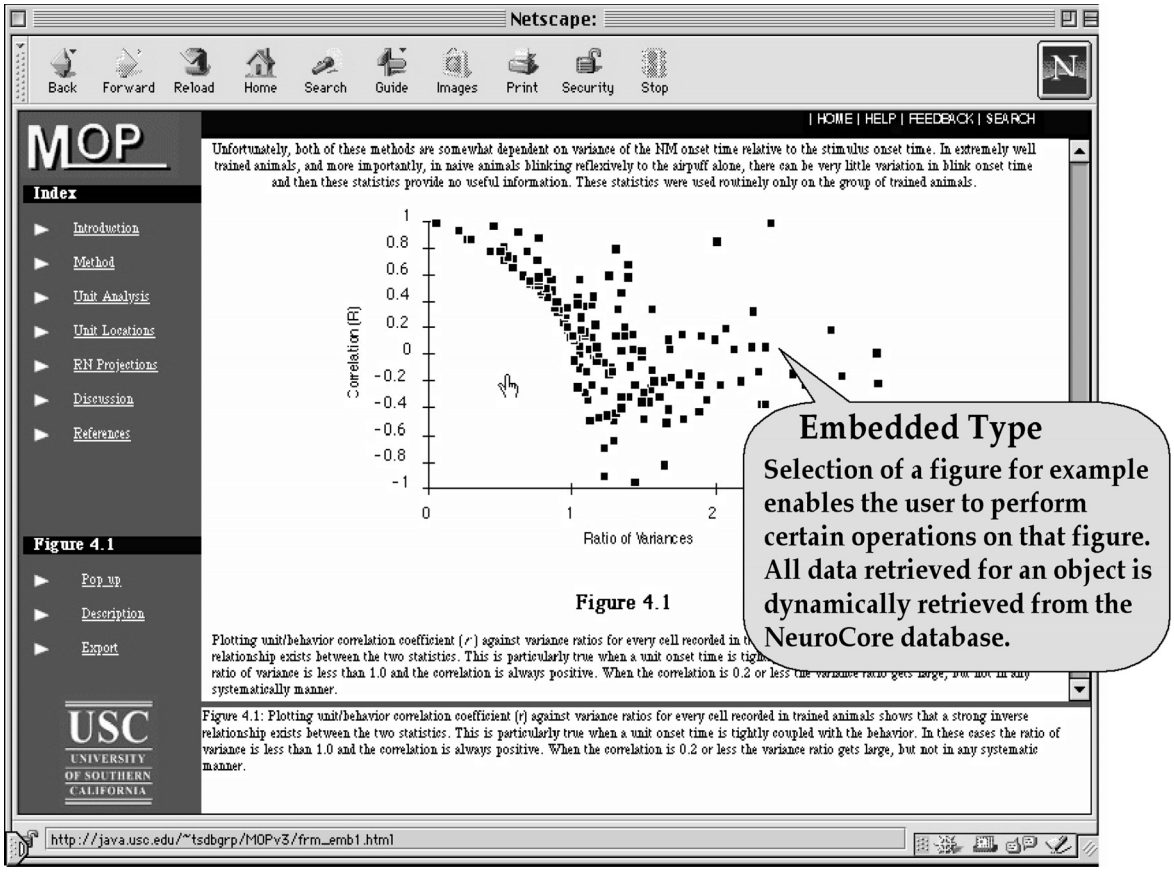

Figure 11 An embedded type in a document. In this example, the chart is an embedded type that can be operated on. In the lower left corner of the interface is a special contextual menu for this specific chart object. In the status window on the bottom of the interface is a complete description of the embedded object.

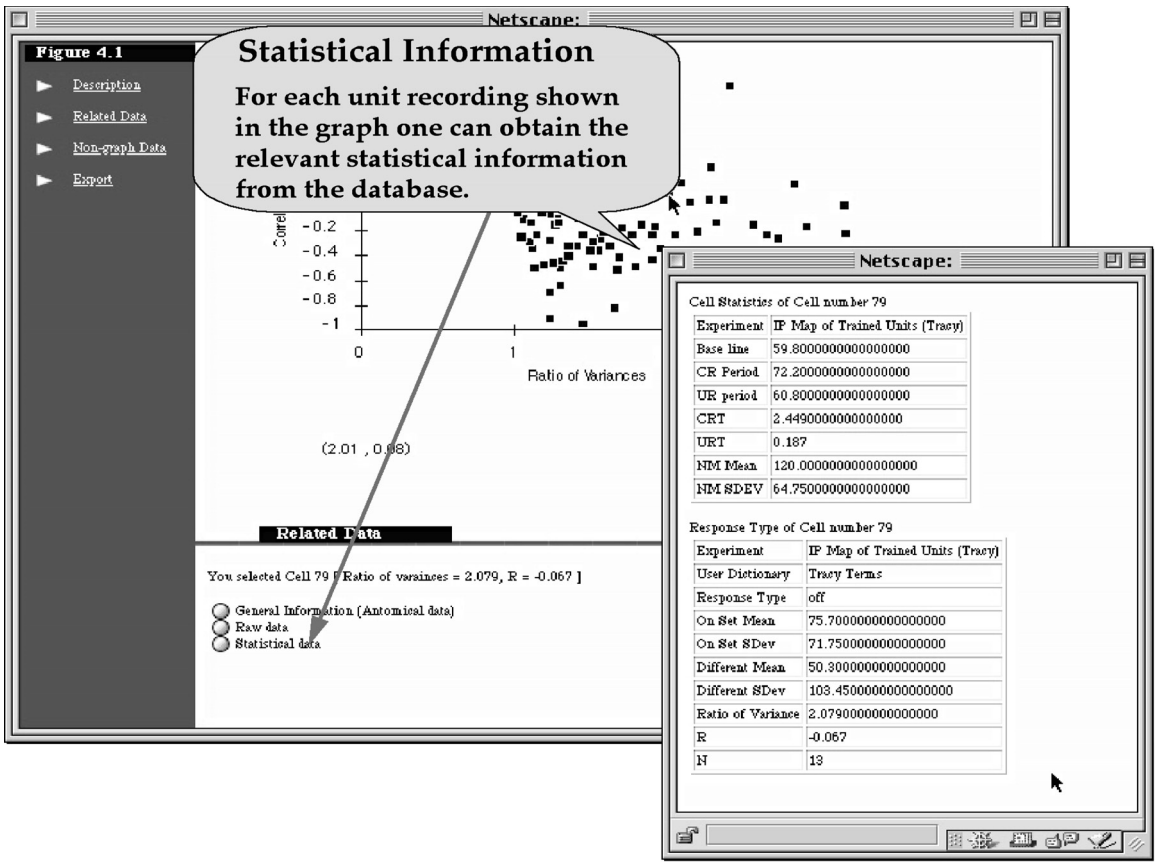

Figure 12 Once an embedded object has been activated, a new window with that object is launched. This allows access to other information stored in the database related to specific data within the object. In this example, a user has selected a specific data point and then has asked for more information concerning that data point from the database. In this instance, the user has retrieved statistical data concerning a specific cell found in the database.

A More Intuitive Interface Than What Is Found in Today's On-Line Journals

Most on-line journals do not take full advantage of the new Web-based medium they are using. Many still look and feel like a printed journal article, with the exception of some hypertext links to on-line reference information. To make an on-line journal more effective, one needs to take advantage of this new medium. For example, one could provide status windows and contextual menus to ease navigation of the document and references (Fig. 10).

The Ability To Drill Down into the Data That Generated a Specific Graph or Chart

With both journals and databases on-line it is now feasible to connect the summary charts and figures found in journal articles to the actual data that generated them (Fig. 12). This will allows users to re-analyze the underlying data themselves, examine the data to find data points that were clipped or not shown in the figure, and to find other data related to individual data points.

References

King, D. A. T., and Tracy, J. (1999) DataMunch Web Site. http://www.novl.indiana.edu/~dmunch.

Tracy, J. (1995). Brain and Behavior Correlates in Classical Conditioning of the Rabbit Eyeblink Response, Ph.D. thesis. University of Southern California, Los Angeles.