CHAPTER 1.1

NeuroInformatics: The Issues

Michael A. Arbib
University of Southern California Brain Project and Computer Science Department, University of Southern California, Los Angeles, California

1.1.1 Overview

We see the structuring of masses of data by a variety of computational models as essential to the future of neuroscience; thus, we define *neuroinformatics* as the integration of: (1) the use of databases, the World Wide Web, and visualization in the storage and analysis of neuroscience data with (2) computational neuroscience, using computational techniques and metaphors to investigate relations between neural structure and function. The challenge to be met is that of going back and forth between model data (i.e., synthetic data obtained from running a model) and research data obtained empirically from studying the animal or human. Research will pursue a theory-experiment cycle as model predictions suggest new experiments and models improve as they are adapted to encompass more and more of these data.

We view it as crucially important to develop computational models at all levels, from molecules to compartments and physical properties of neurons up to neural networks in real systems constrained by real connections and real physiological properties. These can then be tested against the empirical data and that is why it is so valuable to maintain an architecture for a federation of empirical databases in which the results from diverse laboratories can be integrated, and to provide an environment in which we can develop computational modeling to the point where we can make quantitative verifiable or disprovable predictions from the model to the database.

The University of Southern California Brain Project (USCBP) approach to neuroinformatics is thus distinguished not only by its concern with the development of models to summarize and yield insight into data, but also in that we are developing general *architectures* for the support of neuroinformatics. To focus our work in software development, we are building *The NeuroInformatics Workbench*™, a collection of neuroinformatics tools which we summarize below. But, it is important to realize that many other groups will be developing neuroinformatics tools, so part of our work addresses the key issue—for databases, simulators, and all the other tools discussed in this volume—of *interoperability*, ensuring that tools and databases developed by different subcommunities can communicate with each other despite their idiosyncrasies.

Simulation, Databases, and the World Wide Web

Our approach to neuroinformatics is shaped by three technologies: (1) The classical use of computers for executing programs for numerical manipulation (this lies at the heart of our work in modeling and simulation); (2) the development of database management systems (DBMSs), which make it easy to generate a wide variety of databases (organized collections of structured facts) stored in a computer for rapid storage and retrieval of data; and (3) the World Wide Web, which has been transformed with startling rapidity from a tool for computer researchers into a household utility allowing resources appropriately stored on one computer hooked to the Internet (the "server") to be accessed from any other computer on the Internet (the "client") provided one has the URL (universal resource locator) for the resource of interest.

THE WORLD WIDE WEB

The World Wide Web has indeed become so familiar that we will assume that every reader of this book knows

how to use it, and we will repeatedly provide URLs to the databases and tools described in the pages that follow.

DATABASES

On the other hand, we will not assume that the reader has any deep familiarity with databases. Chapter 1.2 introduces the basic concepts. *Relational databases* (introduced in the 1970s) provide a very structured way of describing information using "tables." The current standard for a data manipulation language for querying and modifying a database is SQL (a contraction of SEQUEL, the Structured English QUEry Language introduced by IBM). *Object-based databases* (introduced in the 1980s) organize as "objects"—a rich variety of formal structures. Key structures then include objects, classes (collections of objects with something in common), and inter-relationships which structure the semantic connections between objects and classes. *Object-relational databases* (introduced in the 1990s) combine the "best of both worlds" of relational databases and object-based databases. Our approach to databases in this volume is adaptable to any object-relational DBMS; the implementations available on our Website employ a specific object-relational DBMS, namely the Informix Universal Server. Chapter 5.1 will take up the theme of "Federating Databases."

PROGRAMS FOR NUMERICAL MANIPULATION

We shall not assume that the reader has mastery of a specific programming language such as Java or C++ but will rather provide a view of the simulation environments we have built atop these languages (Chapters 2.2 and 2.3). The expert programmer can follow the URLs provided to see all the "gory details." Here we simply note that Java is an object-oriented programming language (see Chapter 2.2 for the explanation of "object-oriented") that runs on the Web. Most Web browsers today are "Java enabled," meaning that the browser provides the "virtual machine" that Java needs to run its programs on the client's machine. *Applets* are programs that run under a browser on the client machine but as a security measure do not write to the disk on the client machine. *Applications* are programs that do not run under a browser (unless as a plug-in) but can write to the user's disk. Our work on the NSLJ simulation environment (Chapter 2.2) emphasizes the use of Java.

The Challenge of Heterogeneous Data

The brain is to be studied at multiple levels, from the behavior of the overall organism through the diversity of brain regions or functional schemas down through specific neural circuits to neurons, synapses, and macromolecular structures. Consider some of the diverse data that neuroscientists use. For example, the study of animals integrates anatomy, behavior, and physiology. In studying the monkey we may note that there are hundreds of brain regions and seek to provide for each such region the criteria by which it is discriminated from other regions—whether it be gross anatomy, the cytoarchitectonics, the input and output connections of the region, or the physiological characterization, or some combination, that drives this discrimination. Then, for a variety of behaviors of interest—whether it be eye movements, various aspects of motor control, performance on memory tasks, etc.—we may seek to characterize those regions of the brain that are most active or most correlated with such behaviors and then characterize the firing of particular populations of neurons in temporal correlation with different aspects of the task. For example, we have studied the role of the intraparietal sulcus in the control of eye movements (modeling data on the lateral intraparietal sulcus, LIP) and in the control of hand movements (modeling data on the role of the anterior intraparietal sulcus, AIP, and area F5 of the premotor cortex). In such modeling studies, we seek to understand what must be added to the available database on neural responsiveness and connectivity to explain the time course of cellular activity and the way in which they mediate between sensory data, the animal's intention, and the animal's movement.

Increasingly, our studies of animals can be related to the many insights we are now gaining from new methods of human brain imaging, such as those afforded by position emission tomography (PET) and functional magnetic resonance imaging (fMRI). Such methods are based on characterization of very subtle differences in the regional blood flow within particular subregions of the brain during one task as compared to another. As such, it is difficult to determine whether the fact of *lowered* significance in a particular region implies *non*-significance for a task. Moreover, the resolution of human brain imaging is very coarse in both space and time compared to the millisecond-by-millisecond study of individual cell activity in the animal. It is thus a great challenge for data analysis and for modeling to find ways, such as the Synthetic PET method for synthesizing predictions of PET activity developed at USC (Chapter 2.4), to relate the results of the observations of individual neural activity in the animal to the overall pattern of comparative regional activity seen in humans.

All this reinforces our point that the comparison of models and experiments is a crucial and continuing challenge, even though much neuroscience to date has paid relatively little attention to the role of explicit computational modeling of brain function. However, this inattention to explicit models is diminishing, as modeling occurs at many levels, such as: (1) the systems analysis of circuits using, for example, the NSL (Neural Simulation Language) developed at USC to compare such things as the effects of different hypotheses in bringing the activity of model circuitry of the cerebellum and related areas in accordance with observations in the Thompson labor-

atory during classical conditioning experiments; (2) the use of the GENESIS language developed at Caltech and the NEURON language from the University of North Carolina and Yale to relate the detailed morphology of single cells to their response to patterns of input stimulation; and (3) the EONS library of "essential objects of the nervous system" developed at USC to model activity in individual synapses in explicit detail. A challenge for future research is to better integrate the tools developed for the different levels into an integrated suite of multi-level modeling tools.

A crucial challenge, then, is to provide a powerful set of methods for comparing the predictions made by a model with relevant data mined from empirical databases developed under the Human Brain Project and related initiatives in neuroinformatics. We see the key, both for the construction of databases of empirical data and for the comparison of empirical data with simulation results, to be the notion of the *experimental protocol*. Such a protocol defines a class of experiments by specifying a set of experimental manipulations and observations. As a basis for further comparisons, we translate such a protocol into a simulation interface for driving a simulation of the empirical system under analysis and displaying the results in a form which eases comparison with the results of biological experiments conducted using the given protocol.

Federating a Variety of Databases

We here offer two typologies of databases to indicate the different ways in which we will organize the data and the related models and articles, but first we present the notion of federated databases.

Federation

We do not envision there being a single repository of all the data of neuroscience. The way the Web is going, even a single field such as neuroscience may see hundreds, possibly thousands, of databases. There were over 20,000 presentations at the last meeting at the Society for Neuroscience. We expect there to be both personal or laboratory databases and public databases maintained by particular research communities say, people working on cerebellum or on cerebellum for classical conditioning, etc. Each subcommunity may have a shared public database linked to their private databases, thus, workers in neuroinformatics have to understand how to build a *federation* of databases such that it is easy to link data from these databases to gain answers to complex problems. The challenge is to set up databases so that they can be connected in a way that gives the user the illusion of having one wonderful big database at his or her disposal. When a query is made for data from a database federation, the data required may not be in any one of those databases but will be collated from a set of these databases. Users then have the choice of whether to keep the data in their computers as part of their own personal databases or to post the data on one of the existing databases as new information for others to share.

The classic idea of a database federation is to link databases so that each may be used as an extension of the other. We envision a federation linking a multitude of databases accessed through the Web. Our primary strategy has been to design NeuroCore, a database construction system based on an extendable schema (information structure) for neuroscience databases (Chapters 3.1 and 3.2), which makes it easy to link databases that share this common structure. More generally, we envision a "cooperative database federation" linking the neuroscience community. In this approach, the import schema of a given database specifies what data you want to bring in from other databases, and the export schema says what data you are prepared to share and how you will format them for that sharing purpose (Chapter 5.1). In order to be able to connect and access other databases, certain "hooks" have been included in the core database schema to foster such communication. This allows the database to reference and access other databases concerned with published literature as well as on-line electronic atlases. Another possible avenue for database federation in the future is with other neurophysiological database systems using platform-independent transfer protocols such as the TSDP (Time Series Data Protocol) developed by Gabriel and colleagues (Payne *et al.*, 1995). *Databases may be virtual*, integrating partial views gleaned from multiple databases. For example, a database on the neurochemistry of synaptic plasticity might actually be a federation of databases for different brain regions. Moreover, *databases must be linked*: our NeuARt technology (Chapter 4.3) enables an atlas of brain regions to be used to structure data both on the location of single cells (a link to a neurophysiology time series database) and for standardizing slice-based data (such as stains of receptor activity in a brain slice recorded in a neurochemistry database).

Typology 1: The Types of Data Stored

Article Repositories Many publishers are now going on-line with their journals. There are going to be many such *Article Repositories*, including preprint repositories, technical report repositories, and so on. Article Repositories provide an important class of databases—repositories for articles in electronic form, whether they are journal articles, chapters, or technical reports. Even if articles migrate from linear text to hypertext, such narratives about the data—"This is the recent experiment that I did," "Here is my review," etc.—are going to be very important and will often provide the way for humans to get started in understanding what is going on in some domain, even if they will eventually search specific datasets of the kind described below.

Repositories of Empirical Data What most often comes to mind when one talks about databases for neuroscience is what we call a *Repository of Empirical Data*. This is where we get data from different laboratories and make them available either to laboratory members or more generally. Our approach to Repositories of Empirical Data emphasizes the notion of a protocol. In your own laboratory, you can have a bunch of data and place the electronic recordings of what happened at a particular time on disks or tapes and find them in some drawer as needed. But, if you want other people to look at your data, you need to provide a *protocol:* information on the hypotheses being tested, the experimental methods used, etc. It is this protocol that will allow people to search for and find your experimental data even if they did not conduct the experiment. We have developed NeuroCore™ as our basic design for such databases. If your laboratory has special data structures, you can extend this core in a way that makes it simple for other users to understand the structure of your data. One analogy is with the Macintosh desktop, which is designed to meet certain standards in such a way that if you encounter a new application, you can figure out how to use key elements of the application even without reading the manuals. The idea of NeuroCore™ is to provide a general *data schema* (i.e., a basic structure for the tables of data in the database) which other people can extend readily to provide a tailored data structure that is still easy to understand. We have also invested some energy into the MOP prototype Model for On-line Publishing (Chapter 3.3), which increases the utility of online journals, etc. by offering new ways to link them to repositories of empirical data and personal databases.

Summary Databases A *Summary Database* is the place where you go for high-level data, such as assertions, summaries, hypotheses, tables, and figures that encapsulate the "state of knowledge" in a particular domain. A Summary Database is like a review article but is structured as entries in a database rather than as one narrative. If you want to know what is true in a field, you may start with a Summary Database and either accept a summary that it presents to you and work with it to test models or design experiments, or you may follow the links or otherwise search the database federation for data that support or attempt to refute the particular summary. In Summary Databases, assertions can be linked not only to primary literature but also to models or empirical data. One of the issues to be faced below is that, in many fields, there is no consensus as to just which hypotheses have been firmly established. Once you leave the safe world of airline reservations and look at databases for the state of research in any domain of science, you go from a situation where you can just say true or false to the situation where there is controversy, with evidence offered for and against a particular position. Different reviewers may thus assign different "confidence levels" to different primary data, and these will affect the confidence level of assertions in the Summary Database. One contribution of USCBP is the development of Annotation Technology (Chapter 5.4) for building a database of annotations on documents and databases scattered throughout the Web. This may be a personal database for private use or may be a database of annotations to be shared—whether between the members of a collaboratory or with a larger public. In particular, a Summary Database can be seen as a form of annotation database, with each summary serving as an annotation on all the clumps (selected items) that it summarizes. Once annotations are gathered within a database, rather than being embedded in the text of widely scattered documents, it becomes easy to efficiently search the annotations to bring related information together from these many documents. The key idea of Annotation Technology is to provide an extended URL for any "clump" (i.e., any material selected from a document for its interest) which tags the start and endpoint of the clump as well as the URL of the document that contains it. The extended URL methodology then makes it simple to jump to documents, whose relevance can then be determined.

Model Repositories Finally, very important to our concern to catalyze the integration of theory and experiment, is the idea of a *Model Repository*, which is a database that not only provides access to computational models but also links each model to the Empirical and Summary Databases to provide evidence for hypotheses in the model or data to test predictions from simulation runs made with the model. When we design an experiment or make a model of brain function, we have various assertions that summarize what we know for example, the key data from particular laboratories, a table that summarizes key connections, a view of which cells tend to be active during this type of behavior, etc. We have viewed the protocol as a way of understanding what an experiment is all about. When we design a model, we will often give an interface which mimics the protocol so that operations on the model capture the manipulations the experimenter might have made on the nervous system. This will allow the experimenter to make corresponding manipulations through the computer interface to see if the model replicates the results. This makes it easy for somebody not expert in detailed modeling to nonetheless evaluate a model by seeing how it runs in a variety of situations.

In particular, we will emphasize USCBP's model repository, Brain Models on the Web (BMW; see Chapter 6.2). BMW will serve as a framework for electronic publication of computational models of neural systems, as a database that links model assumptions and predictions to databases of empirical data, and as an environment for the development and testing of new models of greater validity. Current work focuses on four types of structures to be stored in the database:

1. *Models:* High-level views of a model linked to the more detailed elements that follow.
2. *Modules:* These are hierarchically structured components of a model.
3. *Simulations:* For each "useful" run of a model, we need to record the parameters and input values used and annotate key points concerning the results.
4. *Interfaces:* To aid non-experts using a model, interfaces must be available to provide a natural way to emulate a number of basic classes of experiments.

With the above typology of databases, we can already see many opportunities for database federation: Entries in a Summary Database may be supported by links to articles in an Article Repository as well as directly to data in a Repository of Empirical Data; articles may come with explicit links from summaries in the articles (figures, tables, assertions in the text) to more detailed supporting of data in the Repositories of Empirical Data; and hypotheses may be supported by models as well as data, thus assertions in a Summary Database may also be linked to predictions in BMW.

Typology 2: Access to Data

In our view, the database federation will include both lightweight personal databases corresponding to personal and collaboratory databases, as well as integrated public databases that serve a whole community. The issue is to foster both development of these individual databases and federation between them which gives each user the most powerful access to relevant data.

Our next typology is based on considerations of *security*. Every item in a database can be tagged for access by specific individuals or groups and *refereeing* items can be tagged for whether they have been posted by "just anybody" or by a member of some qualified accredited group, or whether an editorial board has looked at and passed an item and said, "Yes, that meets our standards." This provides the benefits of immediate access to results that are not guaranteed to be of high quality and delayed access to results that have been refereed. This is a useful model for using the Web to disseminate results. Thus, not only will databases differ in their type and in the particular scientific data on which they focus, they will also differ in their levels of access and refereeing. We see this as containing at least four levels:

1. *Personal laboratory databases:* These contain all the data needed by an individual or a particular laboratory: both data generated within the laboratory (some of which are too preliminary for publication) and data imported from other sources which are needed for the conduct of experimentation or modeling in that laboratory.

2. *Collaboratory databases:* Such databases will be shared by a group of collaborators working on a common problem. This will include all or part of the data in the personal laboratory databases of the collaborators, but because these collaborators may be scattered in different parts of the country or different countries of the world, these various subsets of the shared data must be linked through the Internet.

3. *Public "refereed" databases:* Whereas the above two kinds of databases are the personal property of an individual or a small group which accepts responsibility for the quality of the data they themselves use, there will also be public databases whose relation to the private data is similar to the relation of a published article to preliminary drafts and notes. Just as journals are now published by scientific societies and publishers, so do we expect that public scientific databases will be maintained by scientific societies and commercial publishers. A governing body for each database will thus take responsibility for some form of refereeing as well as ensuring the archival integrity of the database. Given the large size of datasets, we do not envision that in general such a dataset will be reviewed in detail. Rather, we envision two tracks of publication—for articles and datasets—in which there may be a many-to-many relationship between articles and datasets. The articles will be refereed in the usual fashion. A dataset will be endorsed to the extent that it can be linked to articles that have been refereed and support the data; however, there will also be a role for "posters" that have not been refereed but are supported by membership in an established scientific community.

4. In addition, of course, there will be the *World Wide Web*, in which material can be freely published by individuals irrespective of their expertise or integrity. It will be a case of "*caveat emptor*" (buyer beware), as not all scientists are reliable, while lay persons will often come up with interesting perspectives on scientific questions. Clearly, then, there will be many databases of many kinds in the federation that serves neuroscience, in particular, and science more generally.

One of the primary concerns that people have in contemplating the formation of a database federation such as that we envisage for neuroscience is the issue of what is to be done with old data. In the case of private databases, the data can simply "wither away" when the owner of the database no longer maintains the computer on which the data have been stored and provides no alternative means of access to the relevant databases. On the other hand, once a public database has been established, and once a proper form of references has been set up so that people will come to rely on the data that are referred to, then the data "cannot" be deleted. Yet, as time goes by, the way in which such archived data are treated can indeed reflect their changing status in light of new information. Published data can be *annotated* with personal annotations,

refereed annotations, and links to subsequent supporting, competing, and completing material. Data that have proved of less and less current relevance, or whose subsequently questionable status makes them less likely to be referred to, can be demoted to low-cost, slow-access, tertiary storage, thus reducing the cost while the increase in retrieval time becomes of only marginal concern. This is an example of the importance of database research addressing the issue of how to support a user community that needs timely access to increasingly massive datasets (*cf*. Chapter 5.5, Management of Space in Hierarchical Storage Systems).

More generally, within the context of scientific databases, a crucial feature of the USCBP strategy is the linkage of empirical data to models and hypotheses so that the currently dominant ones can help provide and maintain coherent views of increasingly massive datasets. Datasets can then be demoted either because they have become completely subsumed by models that make it far easier to calculate values than to look them up or because the success of the models to fit a wide body of data has made the anomalous data seem suspect—whether because they have been superseded by data gathered with newer experimental techniques or because they no longer seem relevant as challenges useful for the restructuring of theory.

1.1.2 Modeling and Simulation

The term "neural networks" has been used to describe both the networks of biological neurons that constitute the nervous systems of animals and a technology of adaptive parallel computation in which the computing elements are "artificial neurons" loosely modeled after simple properties of biological neurons (Arbib, 1995). Modeling work for USCBP addresses the former use, focusing on computational techniques to model biological neural networks but also including attempts to understand the brain and its function in terms of structural and functional "networks" whose units are at scales both coarser and finer than that of the neuron. While much work on artificial neural networks focuses on networks of simple discrete-time neurons whose connections obey various learning rules, most work in brain theory now uses continuous-time models that represent either the variation in average firing rate of each neuron or the time course of membrane potentials. The models also address detailed anatomy and physiology as well as behavioral data to feed back to biological experiments.

Levels of Detail in Neural Modeling

Hodgkin and Huxley (1952) demonstrated how much can be learned from analysis of membrane properties and ion channels about the propagation of electrical activity along the axon; Rall (see Rall, 1995, for an overview) led the way in showing that the study of a variety of connected "compartments" of membrane in dendrite, soma, and axon can help us understand the detailed properties of individual neurons. Nonetheless, in many cases, the complexity of compartmental analysis makes it more insightful to use a more lumped representation of the individual neuron if we are to analyze large networks. To this end, detailed models of single neurons can be used to fine-tune the more economical models of neurons which serve as the units in models of large networks.

The simplest "realistic" model of the neuron is the *leaky integrator* model, in which the internal state of the neuron is described by a single variable, the *membrane potential* m(t) at the spike initiation zone. The time evolution of m(t) is given by the differential equation:

$$\tau dm(t)/dt = -m(t) + \sum_i w_i X_i(t) + h$$

with resting level, h; time constant, τ, $X_i(t)$, the firing rate at the i^{th} input; and w_i, the corresponding synaptic weight. A simple model of a spiking cell, the integrate and fire model, was introduced by Lapicque (1907) and that coupled the above model of membrane potential to a threshold; a spike would be generated each time the neuron reached threshold. Hill (1936) used two coupled leaky integrators, one of them representing membrane potential and the other representing the fluctuating threshold. What I shall call the leaky integrator model *per se* does not compute spikes on an individual basis, firing when the membrane potential reaches threshold, but rather defines the firing rate as a continuously varying measure of the cell's activity. The *firing rate* is approximated by a simple, sigmoid function of the membrane potential, $M(t) = \sigma(m(t))$.

It should be noted that, even at this simple level of modeling, there are alternative models (e.g., using shunting inhibition or introducing appropriate delay terms on certain connections); there is no modeling approach that is automatically appropriate. Rather, we seek to find the simplest model adequate to address the complexity of a given range of problems. In general, biological neurons are far more subtle than can be captured in the leaky integrator model, which thus takes the form of a useful first-order approximation. An appreciation of neural complexity is necessary for the computational neuroscientist wishing to address the increasingly detailed database of experimental neuroscience, but it should also prove important for the technologist looking ahead to the incorporation of new capabilities into the next generation of artificial neural networks. (For an introduction to subtleties of function of biological neurons, the reader may wish to consult the articles "Axonal Modeling" (Koch and Bernander, 1995), "Dendritic Processing" (Segev, 1995), "Ion Channels: Keys to

Neuronal Specialization" (Bargas and Galarraga, 1995), and "Neuromodulation in Nervous Systems" (Dickinson, 1995).)

We may thus distinguish multiple levels of modeling, which include at least the following:

1. System models simulate many regions, with many neurons per region; neuron models such as the leaky integrator model permit economical modeling of many thousands of neurons and are supported by simulation systems such as the Neural Simulation Language (NSL; Chapter 2.2).
2. Compartmental models permit the modeling of far fewer neurons, unless unusually massive computing resources are available, and are supported by simulation systems such as GENESIS (Bower and Beeman, 1998) or NEURON (Hines and Carnevale, 1997).
3. Even more detailed models may concentrate on, for example, the diffusion of calcium in a single dendritic spine or the detailed interactions of neurotransmitters and receptors underlying synaptic plasticity, long-term potentiation (LTP), etc., as will be seen in the EONS Library (Chapter 2.3).

A Range of Models

Among the foci for USCBP modeling have been

1. *Basal ganglia:* The role of the basal ganglia in saccade control and arm control, as well as sequential behavior, and the effects on these behaviors of Parkinson's disease have been examined.

2. *Cerebellum:* Both empirical and modeling studies of classical conditioning, as well as modeling studies of the role of cerebellum in motor skills, have been conducted.

3. *Hippocampus:* Neurochemical and neurophysiological investigations of LTP have been related to fine-scale modeling of the synapse; we have also conducted systems-level modeling of the role of rat hippocampus in navigation, exploring its interaction with the parietal cortex.

4. *Parietal-premotor interactions:* We have worked with empirical data from other laboratories on the monkey, and designed and analyzed PET experiments on the human, to explore interactions between parietal cortex and premotor cortex in the control of reaching and grasping in the monkey, and, via our Synthetic PET methodology, have linked the analysis of the monkey visuomotor system to observations on human behavior.

5. *Motivational systems:* Swanson has conducted extensive anatomical studies to show the fine division of the hypothalamus into different motor pattern generators and to show their linkage to many other parts of the brain. Our work on modeling mechanisms of navigation also includes a motivational component related to this work.

The essential results for a number of these models will be summarized in Chapter 2.1.

Hierarchies, Models, and Modules

A great deal of knowledge of available neural data goes into the construction of a comprehensive model. In Chapter 2.1 we will present a model of interaction of multiple brain regions involved in the control of saccadic eye movements. Here, we simply want to preview some of the methodological issues involved.

For each brain region, a survey of the neurophysiological data calls attention to a few basic cell types with firing characteristics strongly correlated with some aspect of saccade control. For example, some cells fire most strongly near the onset of the target stimulus, others seem to be active during a delay period, and others are more active near the time of the saccade itself. The modeler using USCBP's NSL Neural Simulation Language then creates one array of cells for each such cell type. The data tell the modeler what the activity of the cells should be in a variety of situations, but in many cases experimenters do not know in any quantitative detail the way in which the cell responds to its synaptic inputs, nor do they know the action of the synapses in great detail.

In short, the available empirical data are not rich enough to define a model that would actually compute. Thus, the modeler has to make a number of hypotheses about some of the unknown connections, weights, time constants, and so on to get the model to run. The modeler may even have to postulate cell types that experimenters have not yet looked for and show by computer simulation that the resulting network will indeed perform in the observed way when known experiments are simulated, in which case: (1) it must match the external behavior; and (2) internally, for those populations that were based on cell populations with measured physiological responses, it must match those responses at some level of detail. What raises the ante is that (1) the modeler's hypotheses suggest new experiments on neural dynamics and connectivity, and (2) the model can be used to simulate experiments that have never been conducted with real nervous systems. The models considered in Chapter 2.1 are fairly complex, yet a few years from now we will consider these models simple, for the new models will both examine the interactions of a larger number of brain regions and analyze cells within each region in increasing detail. There is no way we would be able to keep cognitive track of these models if we had to look at everything at once. Our approach is to represent complex models in an object-oriented way, using a hierarchy of interconnected modules (Chapter 2.2 presents the particular formal approach to modules employed in

NSL). A module might be an interconnected set of brain regions; each region in turn might itself be a module composed of yet smaller modules that represent arrays of neurons sharing some common anatomical or physiological property (Fig. 1). In any case, a module is either decomposable, in which case this "parent module" is decomposed into submodules known as its children modules, or the module is a "leaf module" which is not decomposed further but is directly implemented in the chosen programming language such as Java or C++. In many NSL models, the neuron array provides the leaf modules for a model. In other models, decomposition can proceed further. There are basically two ways to proceed for a complex model. One is to focus on some particular subsystem, some module, and carry out studies of that. The other is to step back and look at higher levels of organization in which the details of particular modules are hidden. We can get both a hierarchical view of the model, where we can step back and analyze the whole model in terms of its overall relationship, or zoom in on subsystems and study them in detail.

NSL Neural Simulation Language

The NSL Neural Simulation Language developed at USC is especially designed for systems analysis of interacting circuits and brain regions. Chapter 2.1 focuses especially on NSLJ, written in Java. The main advantages with Java, of course, are (1) portability: you write it once and "it runs everywhere"; (2) maintainability: you only have to maintain one version of the software; (3) it runs on the client side of the Web; and (4) Java has parallel processing capabilities and a plan for future work is to develop a parallel version of our software.

Schematic Capture

NSL offers module composition to create hierarchical models. It provides layers of leaky integrator neurons connected by masks of weights as the base module for large scale simulations, but finer neuron models may be substituted. Currently, it emulates parallel execution mode. Essentially it has a fairly simple *scheduler* that will take each module in turn to execute the modules sequentially, but because the modules are all double

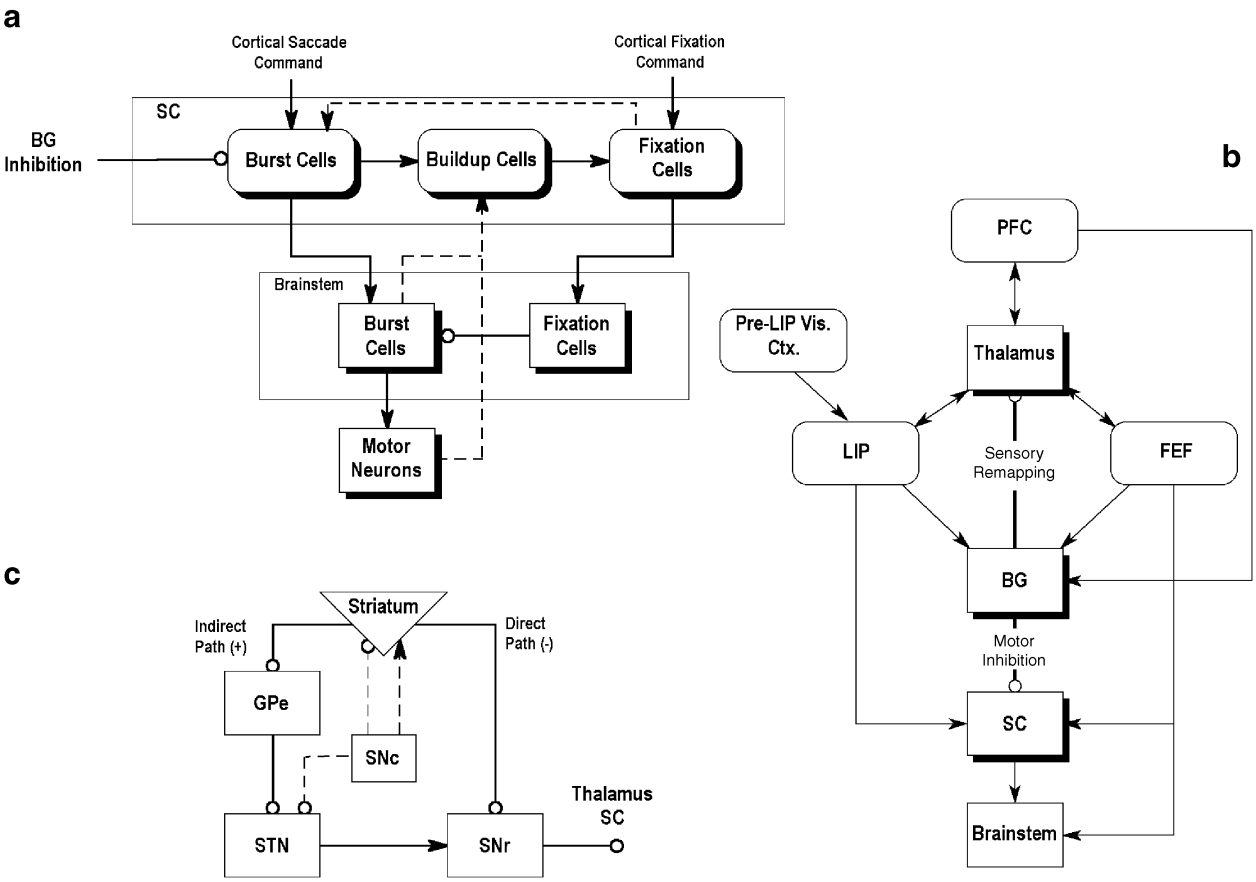

Figure 1 **(a)** A basic model of reflex control of saccades involves two main modules, one for superior colliculus (SC) and one for brainstem. Each of these is decomposed into submodules, with each submodule defining an array of physiologically defined neurons. **(b)** The model of (a) is embedded into a far larger model which embraces various regions of cerebral cortex (represented by the modules Pre-LIP Vis. Ctx., LIP, PFC, and FEF), thalamus, and basal ganglia (BG). While the model may indeed be analyzed at this top level of modular decomposition, we need to further decompose BG, as shown in **(c)**, if we are to tease apart the role of dopamine in differentially modulating (the two arrows shown arising from SNc) the direct and indirect pathways within the basal ganglia.

buffered, it appears as though they are all firing simultaneously.

Given a rich library of modules, users will be able to fashion a rich variety of new models from existing modules, connecting them together and running a simulation without having to write code beyond tweaking a few parameters. A useful new aid to this is the development of a graphical user interface called the Schematic Capture System (SCS), which lets the user do much of the programming at the level of diagrams rather than having to type in every aspect of the model as line after line of code. The SCS lets modelers just draw boxes and label them. When one draws a box, one has to specify what its inputs are, what its outputs are, and what the data types are for each of them. The system will either fill in the information automatically or leave blanks for the modeler to fill in. A drawing tool lets one position copies of the boxes and click to form connections. Again, the SCS will automatically create NSL code for connecting those modules. In the same vein, one can specify, for example, that "basal ganglia" is a unitary module, BG, at the start of model design. Later on, one can click on the BG icon to create a new window in which one can decompose it graphically—with NSL code being generated automatically—until finally reaching the level where one either calls on preprogrammed modules for neural arrays or neurons or writes out the NSLJ code for the leaf modules oneself.

This approach to modular, graphical programming will be made easier by access to libraries containing modules that model portions of cerebral cortex, cerebellum, hippocampus, and so on. These can be plugged together using SCS to build novel models. The SCS is, in a sense, a "whiteboard" that makes it easy to connect different modules out of the library to make a new model and then run it.

Current work at USCBP will provide ways to interface diagrams generated by the SCS with various other databases to link assumptions made in constructing the model to empirical data. Correspondingly, other work on Brain Models on the Web (BMW, Chapter 6.2) will link simulation results to the data which test, whether supporting or calling into question, predictions made with the model.

The SCS style of programming has (at least) two advantages:

1. It makes it easy to program. It is a tool that lets the user place on the screen icons which represent modules already available or yet to be coded and then allows the user to make further copies of these modules and connect them to provide a high level view of a neural model. Any particular module may then be refined or modified to be replaced by a new module within the context of an overall system design.
2. When one views an existing model, the schematics make the relationship between modules much easier to understand. Using the SCS, an experimentalist who does not know how to program would still be able to sketch out at least a high-level view of the model, thus making it easier for the experimentalist and the modeler to interact with each other. A related virtue of the SCS approach is that it encourages collaboration between modelers and experimentalists who can examine an SCS representation of the model and analyze the various connections so displayed and the assumptions on which they rest.

We return to the key notion of the *experimental protocol*, which defines a class of experiments by specifying a set of experimental manipulations and observations. Another tool to aid comparison of experiment and model is the use of simulation interfaces which represent an experimental protocol in a very accessible way, thus making it easy for the non-modeler to carry out experiments on a given model. For example, the interface (Fig. 2) designed for the double saccade experiment described in Chapter 2.1 allows the user to simply click on points of a rectangle representing the visual field to determine the location of the fixation point as well as of targets 1 and 2. Similarly, sliding various bars on the display allows the user to specify the time periods of activation of the fixation and target points. Once this is done, the user has simply to press a "start" button to initiate the simulation and to see various panels representing the activity of different arrays of neurons. Various tools are available to change the chosen set of displays and the graphing conventions used for them. Tools are also available for the recording of particular activity patterns and their printing.

Brain Models on the Web

A major goal of our work is to model the brain in a way that is tightly integrated with experimentation. We are interested in both function and learning. In a sense the whole brain is involved in every task, but holism is not very helpful when one wants to do science. Our modeling strategy, then, for a particular range of behaviors is to start with a data survey to determine a list of brain regions that are involved. Modeling may then concentrate initially on just a few regions to explore what range of behavior is involved, while other models may emphasize other regions. The driving idea is that if all details are modeled initially then it will be almost impossible to understand the effect of any one detail, but if models are built incrementally—both by adding regions and by adding details to the model of a particular region—one will better understand the implications of each part of the model and, it is hoped, the features so represented in the actual brain. It is in this spirit that we have developed NSLJ and SCS to ease the construction and "versioning" of models. These design considerations also motivate our design for Brain Models on the Web (BMW), a database of models with links to Summary

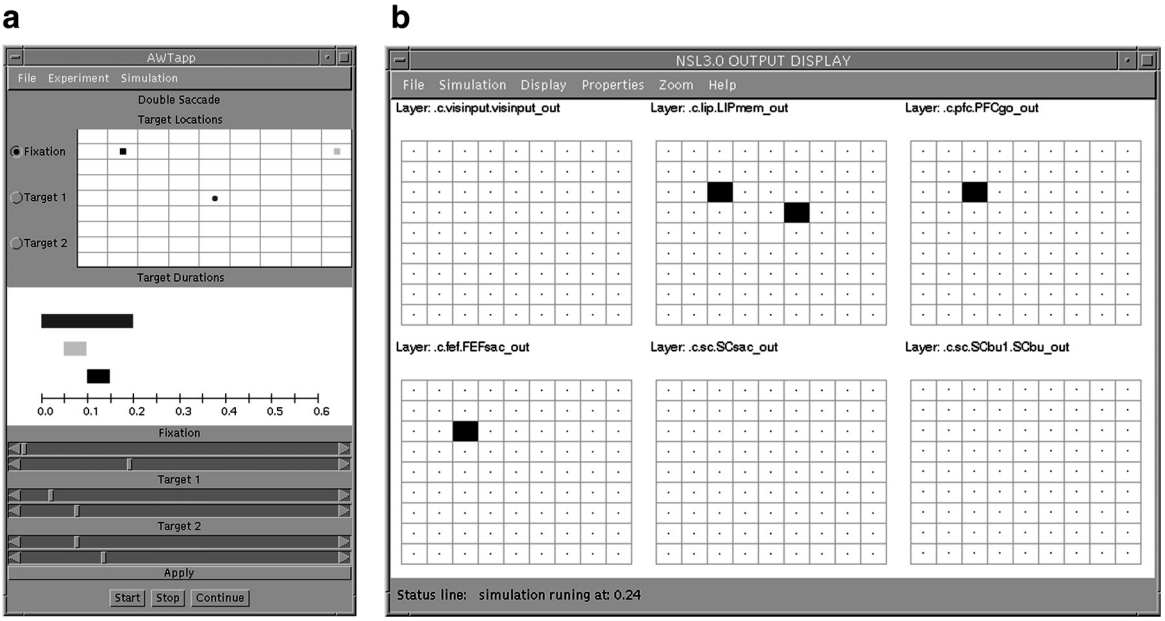

Figure 2 A simulation interface for the double saccade protocol in which a monkey fixates a fixation point during which time two targets are briefly flashed. After the fixation point is removed, the monkey saccades to the remembered position of the two targets in turn. **(a)** The position of the three targets for the simulated experiment is fixed by clicking on the display in the upper panel, and the duration of each stimulus is determined by the sliders in the lower panel. **(b)** This display presents the changing activity during the simulation in six of the arrays of the model shown in Fig. 1c. The menu at the top of the display lets one control the display and change what aspects of the simulated activity are displayed and the type of graphics used to display them.

Databases and Repositories on Empirical Data to support hypotheses and test predictions (Chapter 6.2). The results of analyzing a specific model in relation to the empirical data will in many cases establish a wide range of validity for the model, making confident predictions that can then be checked against empirical data or can be used to design new experiments. In other cases, comparison of predictions with empirical data will enable us to isolate defects in a given model which will lead us to develop new models. It is thus a crucial feature of BMW that it supports both modular structure and the versioning tools which allow one not only to build new models by combining or altering modules from existing models but also to document the efficacy thus gained in explaining a broader set of data, or using fewer assumptions, or gaining greater computational efficiency.

EONS: A Multi-Level Modeling System and Its Applications

The GENESIS and NEURON modeling systems have already been mentioned briefly. Each is designed most explicitly to address the issue of detailed modeling of neurons when the form-function relation of those neurons are to be explained by charting the pattern of currents and membrane potentials over diverse compartments of the structured neuron. At USC, we have addressed an even finer level of analysis, looking at how neural compartments can be further decomposed even down to the level of individual channels placed in spatial relationship across the cell membrane, with diffusion of calcium and other substances in the synaptic cleft defined by these membranes. The idea is, again, to adopt an object-oriented approach, with these "Elementary Objects of the Nervous System" (EONS) being placed together by a composition methodology like that offered by NSL. In fact, in some EONS models (Chapter 2.3), the top module is very small indeed, being a synapse which is then represented by a connection of objects for membranes and the synaptic cleft, and each of these can be further refined in turn.

A major concern in the development of EONS (and it is certainly a consideration for all groups seriously concerned about linking simulation to the data of neuroscience) has been to formalize this process of interaction between modeler and experimentalist. One side of the story, described in later sections of this chapter and volume, is to structure the experimental databases such that a modeler can easily find relevant data by constructing a search based on protocols. The other side of the story is to develop a model that will stimulate the experimenter to test various hypotheses. Whether involving the large-scale study of neural mechanisms of cognitive behavior or the fine scale of spatio-temporal patterns of synaptic transmission, one of the major paths to understanding is by studying the underlying mechanism by way of decomposing an existing model to include lower level features. Another point is matching model parameters with an external protocol so that the experimentalist can look at the protocol and transfer the

parameters and then manipulate the model in novel ways. If a model fails to match the experimentalist's needs, then one needs ways for experimentalists to contribute to the design of new models. Doing so benefits from tools to facilitate sharing and exchange of available models. In this spirit, EONS (following the modular approach of NSL) enables models to be made up from self-contained objects that are described with the neurobiological terms that experimenters use and can form a library of neural objects. A synapse to a biologist is a synapse. It does not matter whether its model is just a number as in most artificial networks, or is an alpha function, or includes the presynaptic release mechanism and the kinetics of the receptors. With this system, we can construct varied models and then ask the question of what would happen if one manipulates them at the molecular level, by emulating the application of certain agonists or antagonists to determine what would happen at a synapse or in network dynamics.

From our modeling point of view, various experimental databases provide different experimental data for constraining and testing the model. On the other hand, the modeler will provide ways for experimentalists to test their hypotheses. In relating to the issues of database management and data mining, the models will also be part of the database search; therefore we can do intelligent searches and provide links for the search. A future goal is to develop a taxonomy of protocols to enable the database system to provide an intelligent search to find and query relevant data more easily.

At present, the EONS library of objects and methods includes numerical methods for the study of molecular kinetics, including diffusion, boundary conditions, and meshing, and provides objects describing axon terminals, the synaptic cleft, the postsynaptic spine, and their further subdivision down to the level of ion channels and receptor channels. One set of simulations has looked in detail at a two-dimensional slice across the synaptic cleft, representing the way in which vesicles release neurotransmitters into the cleft and how calcium diffusion influences the way in which neurotransmitters affect the receptors in the postsynaptic membrane. It has been shown that not only can the position of the vesicle relative to the receptors be important, but the very geometry (as revealed by EM) of the membranes can have a dramatic effect on synaptic efficacy.

Multi-Level Simulation: Complexity vs. Efficacy

We close with the interesting fact that a careful simulation of several seconds of activity in a single synapse at this level of resolution requires 24 hours of computation by a moderately powerful workstation of 1998 vintage. Jim Bower (personal communication) reports that, in 2000, the world's fastest supercomputer can only handle six of his GENESIS simulations of the Purkinje cells of the cerebellum. Recall that there may be of the order of 10,000 synapses on a "typical" neuron, millions of neurons in a single region, and hundreds of regions in a brain. Clearly, any simulation methodology which simply required one to simulate every synapse or every neuron in such detail would be doomed to failure. No short-term increase in computer power will allow us to reduce the simulation of a system with 10^{15} synapses from 10^{15} days (even ignoring all the overhead of connectivity and non-synaptic membranes) down to a single second.

A major challenge for our work in multilevel simulation is thus to understand how to use detailed simulation at one level to validate a (possibly context-dependent) approximation that can be used in far more efficient large-scale simulations at the next level. For example, a NSL model might employ a neuron module that is far simpler than a corresponding compartmental model developed in NEURON but that has been validated by careful studies to yield an economical but effective approximation to it. Or a GENESIS modeler might want to check that a model of a compartment provides a satisfactory approximation to a far more detailed EONS model. All this raises two important challenges for the neural simulation community. One is to increase the range of tools currently available for comparing model to model, as well as model to data (with the parameter search methods that this implies). The other is to develop "wrapping" technology, so that modules developed using one simulator can indeed be used to replace objects (whether to simplify them or attend to crucial new details) in an existing model developed using another simulator. For example, if we had a large network model in NSL using leaky integrator neurons, we might like to plug in a more subtle neuron model of the individual neurons. It would then be more efficient to wrap a GENESIS or NEURON model of each neuron to serve as a module in a new version of the overall NSL model than to reprogram these complex neuron models in Java to fit them into the NSLJ environment directly. This topic is one of the USCBP goals for outreach to the broader neuroinformatics community, creating a set of standards for modularity, versioning, and data linkage so that wrapping technology will enable BMW to document and provide tools for linking models built using multiple neural simulators, not just the NSL system.

Brain Imaging and Synthetic PET

Since the neuroinformatics of human brain imaging is so well covered by many research groups with and without Human Brain Project funding, this has not been a major focus of USCBP research. However, we have been concerned with the question: "How can the data from animal neurophysiology be integrated with data from human imaging studies?" Our answer is *Synthetic PET Imaging* (Chapter 2.4), a technique for using computational models derived from primate neurophysiological data to predict and analyze the results of human

PET studies. This technique makes use of the hypothesis that regional cerebral blood flow (rCBF) is correlated with the integrated synaptic activity in a localized brain region. We first design NSL models of a key set of brain regions in the monkey, specifying the simulation of visual input and motor output in relation to the neural networks of the model. Synthetic PET measures are then computed for a set of simulated experiments on visually guided behavior in monkeys and then compared to the results of a similar human PET study. The human PET results may be used to further constrain the computational model. Moreover, the method is general and can potentially accommodate other hypotheses on single-cell correlates of imaged activity; it can thus be applied to other imaging techniques, such as functional MRI, as they emerge. Thus, although the present study uses Synthetic PET, we emphasize that this is but one case of the broader potential for systems neuroscience of synthetic brain imaging (SBI) in general.

1.1.3 Databases for Neuroscience Time Series

Neuroscience provides many examples of time series data (Chapter 3.1). Fig. 3 shows the well-known Pavlovian paradigm of "classical conditioning" as used in the Thompson laboratory at USC, with a blinking rabbit rather than a salivating dog. Puffing some air at the eye (the *unconditioned stimulus*) yields a blink (the *unconditioned response*; actually a closure of the nictitating membrane, the "third eyelid") a little later. Thompson precedes the airpuff with a tone as the *conditioned stimulus*, and eventually the animal learns this relationship and will eyeblink at each tone in anticipation of the airpuff, thus avoiding the noxious stimulus. The issue for Thompson's laboratory for many years now has been to go beyond Pavlov's behavioral studies to track down the neural mechanisms underlying that phenomenon, looking for cells in the brain responding in relation to these different effects and changing as conditioning proceeds. In fact, such changes are crucially observed in portions of the cerebellar cortex and the interpositus nucleus which lies beneath it.

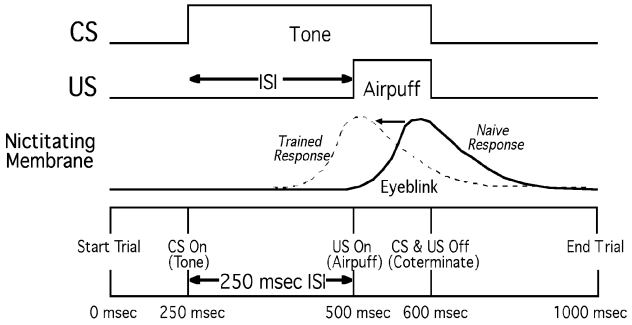

Figure 3 The protocol for using a tone to condition the eyeblink response of the rabbit.

Consider the time series data for such a study of classical conditioning shown in Fig. 4a. The top panel presents separate traces from separate trials of the same

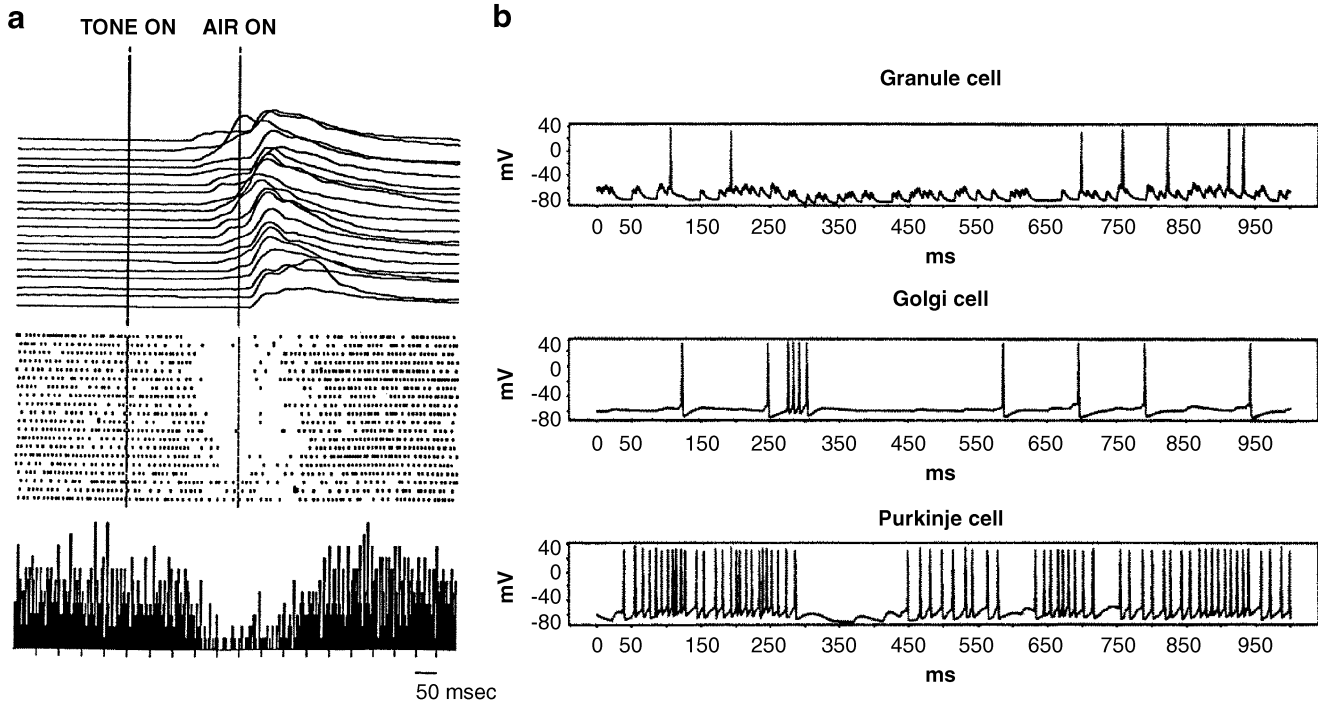

Figure 4 **(a)** Data from a Purkinje cell in temporal relation to the eyeblink behavior. **(b)** Result of a model of cellular interactions. These displays indicate the importance of linking empirical data and synthetic data (simulation results) to a common protocol for the comparison of data and model.

experiment, showing the movement of the eyelid. Each line in the middle panel is a "raster display," a series of dots corresponding to the firing of an action potential along the axon of a single neuron. At the bottom is the histogram produced by adding the firings of the neuron over the trials of the second panel which emphasizes the pause in firing that precedes the movement of the eyelid. This example brings up the database issues: How do we store time series? How do we register data with time stamps to facilitate interesting processing of sets of data? We need to store data with a *protocol* making explicit what hypotheses were being tested and what experimental methods were used. This must be supplemented with explicit data on what conditions were required to elicit each data set. To address these issues, we have developed NeuroCore, a general structure for the design of neuroscience databases (Chapter 3.2). Fig. 4b provides results of simulation with a detailed model, stressing the need for the use of a common protocol to structure comparison of model and data. We want to be able to take real neural data and compare them with the results of elaborate simulations of various brain regions. In this case, we use results from a model developed at USC by Gabor Bartha. He developed a network model, with biophysical properties built into the neurons, and then predicted patterns of firing of cells in cerebellar cortex and interpositus in the untrained and trained animal. Fig. 4 compares simulations predicting a shutdown in Purkinje cell activity with real data showing just this effect. We relate real data to computational predictions.

Fig. 5a shows the NeuroCore database architecture we have developed at USCBP for database management. The left-hand side of the figure shows the software required to keep track of queries in the standard query language (SQL; see Chapter 1.2) and to structure, enter, and retrieve data. The Informix architecture allows one to plug in a set of "Datablades." Consider how a Swiss Army knife has ordinary blades for cutting and then a set of additional devices for removing stones from horses' hooves and other important operations. In the same way, if one has a database functionality to add to the basic relational structure (the standard SQL processes), one can design or purchase a Datablade to structure and process data appropriately. The Web Datablade makes it easy to use a friendly Web interface to post queries and get the results. There are various Datablades available for two- and three-dimensional pictures and images. The previously available time series Datablade for financial applications was not suitable for neuroscience applications, so we developed a new neuroscience time series Datablade (Appendix 2) to handle the sort of spike data and behavioral data shown in Fig. 4(a).

NeuroCore provides a "core schema," a novel extendible object-relational database schema implemented in Informix. The schema (structure of data tables, etc.) for each NeuroCore database is an extension of our core database schema adapted to meet the needs of some group of neuroscience laboratories. Fig. 5b shows how the core database schema can be extended to accommodate Thompson's data. As shown in Fig. 6, the Core Experimental Framework, which we can link to the neuroanatomical and neurochemical concepts, provides an extendible specification of items needed in most experimental records, such as research subject, experimental manipulation, structure of the research data, and the statistics performed on the data. We see a slot for research data and a standard extension for handling time series data. This is then extended for the needs of this particular laboratory to provide fields for eyeblink (nictitating membrane response) data as well as unit data from the cells, whether from one unit or many units at a time. These are the sort of data we saw in Fig. 4a. Chapter 3.1 has more to say on this example and also discusses in some detail a protocol for intracellular recordings from hippocampal slices as an example of the flexibility of NeuroCore in developing Repositories of Empirical Data for neuroscience. NeuroCore comes with a Java applet called the Schema Browser, which allows one to learn the structure of a particular laboratory's database by showing, for each familiar core table, the extensions particular to that laboratory. Thus, the database structure becomes easy to understand.

Fig. 5a also indicates the use of Netscape or other browsers to go beyond whatever standard interfaces are given by the Web driver to develop an *on-line notebook* interface which make it easy both for the experimenter to enter comments and ideas during an experiment and for anybody to analyze the data (Chapter 3.3). The aim is to replace the situation where uninterpreted data are stored on disks or reels of tape with experimenter's comments in a separate handwritten notebook by a format that allows the experimenter to enter easily everything that would have been entered in the written notebook, and moreover, to have it time stamped and locked to the electronic data which themselves are coupled with that protocol information so the nature of the experiment, the fine data, and the comments are all electronically linked together. For the database and on-line notebook to foster inter- as well as intra-laboratory collaborations and communication, there need to be security protocols that allow researchers to "publish" and/or share their data in a secure fashion. Currently a simple security scheme has been implemented that allows us to track usage of the database through the on-line notebook. Built-in Informix security features allow a researcher to store his own data securely in the database; however, we are currently implementing a more complete security scheme to allow researchers to share their data in a secure fashion as well. Another way of extending the core, shown in Fig. 6, is by federating the given database with other databases, providing interfaces with, for example, BMW (our Model Repository) and other databases of neural data and literature (Article Repositories) and Summary Databases.

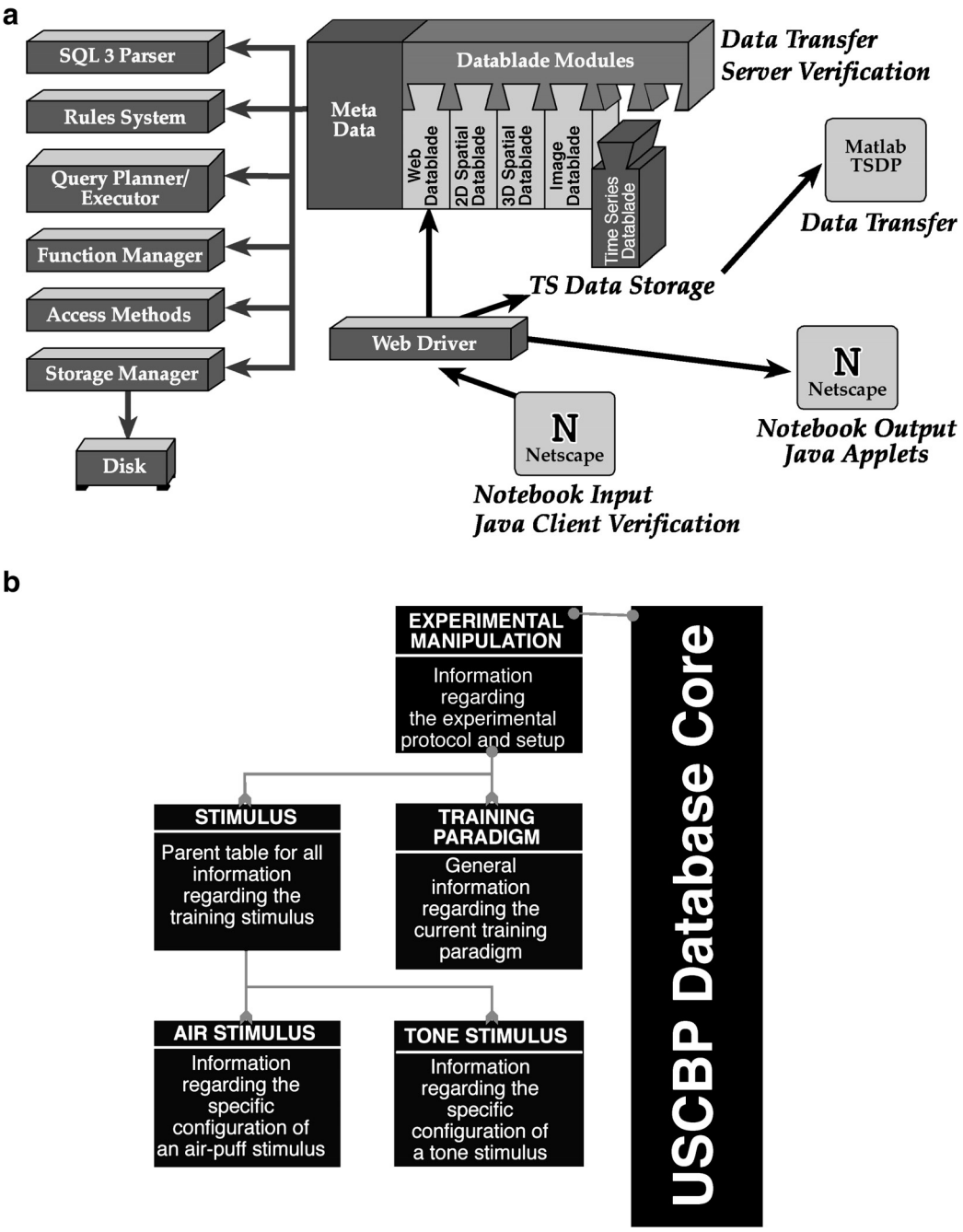

Figure 5 **(a)** The NeuroCore system, the general structure for the design of neuroscience databases developed by the USC Brain Project, as implemented in Informix, with linkage to the Web. **(b)** How to embed the Classical Conditioning Protocol in NeuroCore, the extensible structure of NeuroCore.

At USC, the protocols used by various laboratories are different because the research is fairly different. But, as we build a database protocol, we can converge with laboratories doing similar work at other institutions. For example, people doing classical conditioning on the cerebellum might have a shared extension which will handle about 80% of the variance. In that community, a laboratory that has already developed a successful extension of NeuroCore would, as freeware or for a fee, offer its database schema to other people working in that area; the extensions required for any other laboratory working with a similar research paradigm would be minor, making it easier for colleagues to share and compare their data. However, researchers do not have to agree on all the appropriate extensions. Our goal is federation without conformity. Note that we do *not* take responsibility for providing protocols for all types of experiments. That would exceed our knowledge and resources. Rather, our

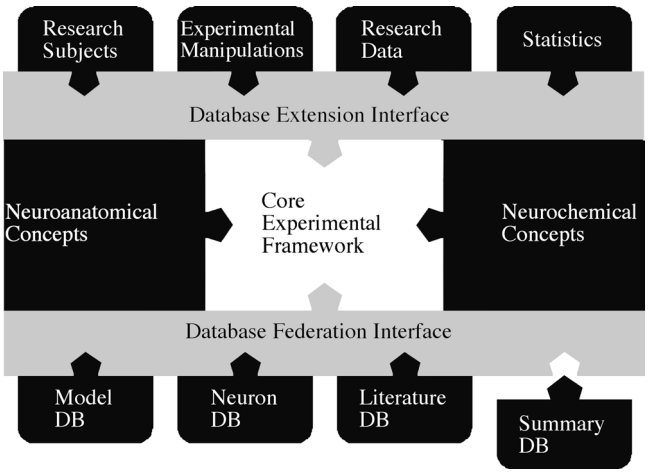

Figure 6 The USCBP NeuroCore database architecture.

task is to document NeuroCore and the tools for its extension, and clearly explain enough key examples to allow other researchers to program their own protocols and use the NeuroInformatics Workbench.

1.1.4 Visualization and Atlas-Based Databases

How are data from diverse experiments on the brains of a given species to be integrated? Our answer is to register the data—for example, the locations of cells recorded neurophysiologically, the tract tracings of an anatomical experiment, or the receptor densities revealed on a slice of brain in a neurochemical study—against a standard brain atlas for the given species, such as that for the rat brain developed at USC by Larry Swanson. Just as people have different faces, so do rats and other animals have different brains; therefore, there is a registration problem: given a location in an individual brain, what is the "best bet" as to the corresponding location in the "standard" brain?

The Swanson atlas (Swanson, 1998) contains 73 plates representing cross-sections of one half of the rat brain. These are not uniformly spaced, but were rather chosen to exhibit many crucial features of the rat's neuroanatomy. Each plate contains a photomicrograph of a stained brain section on the left and Swanson's representation of that section on the right, in which he draws boundaries separating different brain regions and labels the regions. We use the term "level" to refer to a two-dimensional representation of a slice of the rat brain obtained by pairing one of Swanson's drawings with its mirror image. Many of the curves dividing one nucleus from another correspond obviously to boundaries in the cell densities visible on the micrograph. Others cannot be seen from that particular micrograph and can only be revealed by a variety of staining techniques or by the incorporation of physiological and other data. It thus requires great skill on the part of the anatomist to draw those "non-obvious" divisions, and in fact even expert neuroanatomists may disagree. Thus, while there is much agreement between the Swanson atlas and the other leading atlas of the rat brain, the Paxinos-Watson atlas (Paxinos and Watson, 1998), there are also disagreements. Thus we have the future challenge of not only registering data against a particular choice of atlas but also facing the issue of how to update such datasets as future anatomical research resolves certain disagreements and leads to more reliable demarcation of boundaries.

Swanson has used his atlas as the basis for a personal database of PHAL (Phaseolus Vulgaris Leucoagglutinin) tract-tracing sections related to the projections of different regions of the hypothalamus. A tracer is injected into some region of the brain of interest, and this tracer is picked up by axons leading either into the given region or out of the given region. Successive sections through the brain may then reveal the stain which allows one to follow these fibers. In Swanson's laboratory, these observations of successive slices of different brains are meticulously drawn onto the different levels of the Swanson atlas, forming layers which can be shown in registration with the template for that level of the brain. Initially, all this work was done using Adobe Illustrator on a Macintosh, and the results were thus only available to someone who had access to all these files as a download onto their Macintosh. For us, the challenge was to replace this personal utility by a net-accessible database, in which the templates for different brain regions and the overlays from different experiments become elements in the Web-accessible database.

The solution to this problem is called NeuARt, a viewer for atlas-based neural data (the NeuroAnatomical Registration Viewer) which, though initially developed to register data against the Swanson atlas, is in fact a technology applicable to any atlas of the brain. For example, James Stone, formerly a member of USCBP and now at University of California, Davis, is adapting NeuARt to display data on the monkey brain gathered by Edward Jones. But, here, let us concentrate on the use of NeuARt with the Swanson atlas. The system allows one to view through a Web browser any level of the Swanson atlas together with any overlays retrieved from the database (Fig. 7). A Display Manager allows one to see these different results, and a Viewer Manager allows one to customize the Display Manager to one's needs. The Query Manager provides forms which make it easy to request anatomical information from our Informix database, the results of these queries are described textually by a Results Manager, and the user can maintain a set of results of interest. The Level Manager allows one to choose which level of the brain to examine, and the Active Set Manager then shows which results of the query have data relevant for that set. These can then be displayed by clicking on the appropriate elements.

NeuARt alone, however, does not solve the problem of transforming the results of an experiment into data

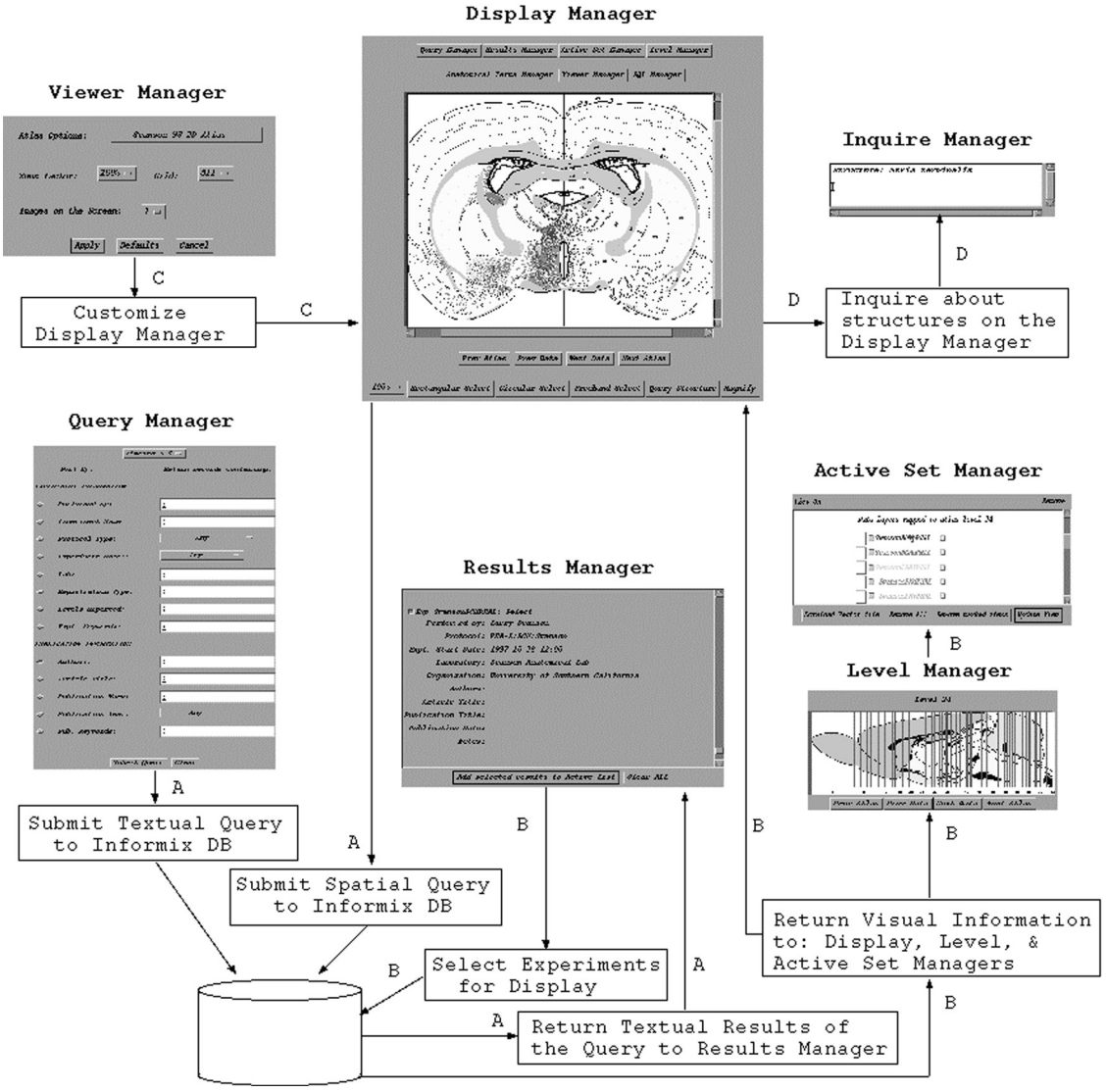

Figure 7 An overview of the NeuARt system. The Level Manager allows one to choose which level (i.e., drawing of a cross-section of the rat brain) of the Swanson atlas to examine. The Display Manager allows one to view through a Web browser any level of the Swanson atlas together with any overlays retrieved from the database. The Viewer Manager allows one to customize the Display Manager to one's needs. The Query Manager provides forms for requesting anatomical information from the database. The results of these queries are described textually by a Results Manager, and the user can maintain a set of results of interest. The Active Set Manager then shows which results of the query have data relevant for that set. These can then be displayed by clicking on the appropriate elements.

that can be overlaid against the atlas. Not only do different brains within a species differ, but also (even if we were using clones with identical brains) each brain will undergo different patterns of shrinkage as it is prepared for sectioning, and any actual slice made by the neuroanatomist will vary from those already used in the atlas. Thus, registering data against a level already in the atlas is not an optimal approach. We have thus produced a three-dimensional reconstruction of the rat brain by outlining the boundaries of each region in all 73 levels of the Swanson atlas and then using the Microstation CAD system to join up the outlines of a given region to form a three-dimensional representation as a surface bounding the region (Chapter 4.4). This surface can be rendered for viewing at different angles but, even more importantly for our present concern, the various surfaces can be sliced at arbitrary angles. Thus, given a particular slice of a particular brain containing data of interest—whether stains marking our fibers of passage, or stains representing density of chemical receptors, or marks indicating the position of cells encountered in a neurophysiological experiment—we match the slice not to the closest profile in the atlas, but rather to a whole variety of slices obtained from the three-dimensional atlas. We have used the warping algorithm developed by Fred Bookstein (1989), which provides a number called the "Procrustes distance" that indicates how far the landmarks on the original slice had to be moved to bring

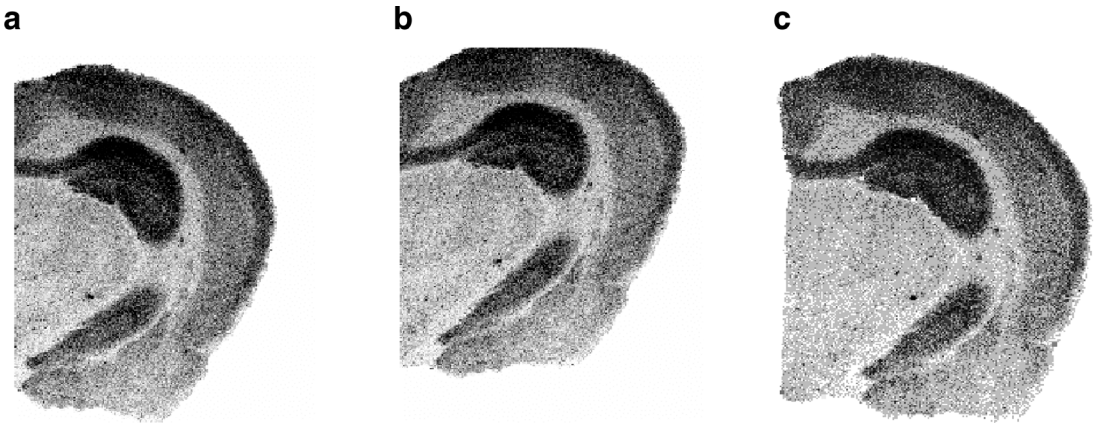

Figure 8 Improved registration results for matching against a three- rather than a two-dimensional atlas. **(a)** Experimental image. **(b)** Warped to closest profile in the atlas; Procrustes distance is 0.2135. **(c)** Warped to closest profile from resectioning; Procrustes distance is 0.0958, a numerical and visible improvement.

them into registration with landmarks on the slice from the atlas. We thus register the data slice to the atlas level which has the minimum Procrustes distance to yield our estimate of how best to embed this specific experimental data into our three-dimensional atlas of the brain. Fig. 8 demonstrates the improvement obtained by registration against the three-dimensional atlas.

We close this section by noting some of the other challenges for atlas-based databases. One is the issue of cytoarchitectonics, showing for each region of, for example, cerebral cortex, the distribution of cell bodies as seen in different layers through the cortex, a pattern that varies from position to position in the cortex. We have already spoken of registering the position of specific cells as identified during neurophysiological experiments. We can also link fine-grain neuroanatomical data to specific points in the brain to show what the characteristics of cells are as seen in that particular sub-area. For example, studies in neuromorphology may demonstrate the typical branching pattern of the dendrites and axons of a cell of given type in a given region and also show the distribution of synaptic spines along the various branches of the dendrites. An additional challenge to work on registering of brain sections is posed by the study of brains that have been damaged. As is well known by anyone who has watched television commercials, it is "easy" to morph any object into any other object; in particular, it is easy to register a brain section which has been lesioned against an intact brain section from the atlas. Thus, we must extend our registration technology to not only map to whole sections but to map to partial sections indicating not only what is the best slice for registration but also which sub-portion of that slice best matches the tissue of the experimental data that survived the lesion.

Given experiments on, for example, binding two different receptors, we may use registration to aid comparison of localization. The improved registration obtained with the three-dimensional atlas allows one to see subtle changes that were missed with less careful registration for analysis. Such results are important for the development of our atlas-based database of neurochemical data (Chapter 4.5). With good registration, it is easy to subtract one image from the other to isolate differences in, for example, two different ligand bindings on the same type of receptor (e.g., a map of AMPA-CNQX).

1.1.5 Data Management and Summary Databases

The USC Brain Project is developing a set of exemplary databases as a core for a larger federation of databases and will also develop tools such as NeuroCore for formatting neuroscience databases so that other people can build databases that are easy to federate with those at USC. However, no matter how much information we have in our own set of databases, there is going to be much relevant material "out there on the Web." Other groups will have interesting databases that are in a different format from NeuroCore. For example, people in neuroscience increasingly study genetic correlates of structure and function, with knock-out mice providing one exciting example. Thus, neuroscientists will want access to genome databases which are going to use different data structures, and this will pose a challenge for database federation. We also study techniques to manage and mine data from elsewhere and import them when we need them to augment our own databases.

Federation

No monolithic database will serve the needs of all neuroscientists; rather, neuroscience will rely upon a federation of databases (i.e., a set of databases linked in such a way as to allow queries to be answered by data gathered from any relevant database). Although private

data could be stored in public databases with tags that limit access, many users will be concerned about security or will prefer the added control of keeping private data in a private database on a workstation whose contents are not accessible to general users of the Internet. Such lightweight personal databases might use relatively inexpensive, relatively widely available database managers, whereas the integrated public databases would use more powerful engines, such as the Informix DBMS used at present by the USC Brain Project. As papers are published, the data related to these publications should then be made available in Repositories of Empirical Data housed in integrated public databases which are accessible to a broad community of users via the Internet. Individual laboratories with the lightweight personal databases in which they develop new data or simulations will be linked to a public database in which the relevant results would eventually be published, with further potential access via the Internet to all the integrated public databases that serve the neuroscience community. We emphasize two different forms of access: one is to publish models or data, probably in a specific few integrated public databases; the other is to look for relevant data—whether for atlas data, other empirical data, articles in Article Repositories, or models—by broad searches across the Internet.

Federation is the interconnection of databases, with some variation in structure, so that they may be loosely coupled to support information sharing. This may involve more or less centralized control of information as there can be a spectrum of architectures for federated databases. The aim is to support information sharing between heterogeneous databases. The old pre-federation solution was full centralization, having an integrated database that subsumes the individual databases, replacing their individual schemas by one unified schema; however, the natural inclination of different user communities is towards heterogeneity. The problem, necessitating semantic models, is that terminology may differ from one database to another, and so the issue of matching the fields of one database to the fields of another database will be non-trivial and may depend more on negotiated agreements than on any automatic process. Our hope is that NeuroCore will develop into one interlingua so that many workers in neuroscience can communicate via this database structure whatever they want to export or import with other databases.

Federation provides the middle ground between the two extremes of integration with full centralization on the one hand and full autonomy on the other. One key aim is to support *discovery* in the sense of finding relevant data. This becomes very difficult in a too loosely coupled system, as there is no centralized knowledge. Because there will be databases that do not conform with our basic NeuroCore structure, research on database federation will be required to provide tools whereby some intermediate structure can be created to make "foreign" data more readily accessible, maintaining information about the import and export schemas of the various databases and providing some "dynamic knowledge" of the available types of data as the pattern of sharing evolves over time.

This may be done "manually" by directly pointing (e.g., from a feature of a model to relevant laboratory data) or from a cell recording to atlas coordinates. This is useful in many cases, but often we would like to replace specific pointers by a generic description that can yield updated retrievals as the available data set changes. We want to avoid manual updating and "truth maintenance" to the extent possible.

Summary Databases: The Essential Notion Is the Clump

Journals are now available on-line, and a number of these journals provide the facility to link to "backup data sets." What we add is that Summary Databases may provide access to many different Article Repositories and Repositories of Empirical Data, and that "backup data" will not be isolated as appendices to specific articles but will be structured within Repositories of Empirical Data where they may more easily be collated with related data. A Summary Database might serve a large community, cover a general theme or a specialized theme, or be a personal database. In each case, the user needs tools to build the database and mechanisms to determine which users have access to a given class of data. We also need tools for merging (portions of) compatibly structured databases. For example, the author of a review article may simultaneously have (1) the article accepted for insertion in an article repository and (2) the personal Summary Database developed in compiling an article (with its assertions anchored by links to Article Repository clumps, Repository of Empirical Data data, and BMW models) merged into a public Summary Database serving the same community as the Article Repository. Taking an electronic file and adding it to an Article Repository is a well-understood process; much work remains to determine how to merge Summary Databases efficiently.

Whether an experimentalist is summarizing the fruit of multiple experiments, a reviewer is summarizing material in a variety of articles, or a modeler is providing the general implications of a set of modeling studies, the basic item in a Summary Database will be an *assertion* that can be supported, or contraverted, by the citation of specific data sets from a Repository of Empirical Data, specific extracts from an Article Repository, or specific simulation runs from a Model Repository. We use the term "clump" for the basic pieces of information which the summary thus refers to.

In other words, links to articles and databases will most usefully point to specific *clumps* of related material, rather than to the article or database as a whole. At

present, a reference is usually to an entire article, or in some cases a specific page, table, figure, or equation. The notion of a *clump* generalizes this; a clump can be any set of sentences, parts of figures, entries in a table, etc. that provides the minimal description of a particular idea in an article or database.

In general, when we follow a link to an article repository, we would prefer to be sent to a highlighted clump in the article of interest, rather than to the first page of the article and then have to scroll through the article to find material of apparent relevance to the pointer. We thus need to provide a unique coordinate system that can identify portions of figures, videos, computer demos, etc., as well as portions of text. A clump can then be specified by giving its extended URL, the URL of the overall article together with the set of coordinate tuples that specify the constituents of the given clump. Currently, we have completed the task of extended URL definition for portions of a hypertext document and have provided the means to click on an index entry and be transferred to the relevant portion of the document with the desired clump of text shown highlighted. This work is part of USCBP's annotation technology. Selection of a clump involves generalized highlighting similar to normal click-and-drag highlighting but generalized to allow highlighting of several non-contiguous items within a given clump. Future work will provide appropriate extensions for figures, videos, computer demos, etc.

A clump may reside in the Article Repository, included in the set of indexed clumps which provide part of the extended hypertext of the article. More generally, it will reside in a Summary Database. The clump may be copied into the Summary Database if the owner of the original material grants permission. Alternatively, following the link from the Summary Database to the Article Repository may require a password and fee for access. As in a review article, the Summary Database may then contain a paraphrase or brief description to indicate the key point of the clump rather than its full content.

Model components may have explicit links to assertions, clumps, and laboratory data, as well as comparisons with elements of other models. When a model is consulted after its initial development, the assertions on which it is based can be used to anchor processes designed to discover new data which support these assertions or call them into question, thus allowing the user to judge the continuing validity of a model or to design paths whereby it may be updated.

Dynamic Classificational Ontologies

In philosophy, "ontology" studies "being as such," including the general properties of things. Quine (1953), however, saw ontology as concerning the question, "To the existence of what kind of thing does belief in a given theory commit us?" This question takes us halfway towards the definition of ontology used by database practitioners, a collection of concepts and their relationships used to describe a given application area and/or a database providing data about the given area. The key problem is that even if two databases record data describing similar aspects of the external world, the actual base concepts in each database might be quite different. "Personnel" might be an explicit concept in one database, but an implicit subset of "People" in another database. This raises a key issue for database federation, finding ways to translate between the different ontologies of different databases which contain related data necessary to fully answer a query. To address this, we developed the technique of dynamic classificational ontology (Chapter 5.2).

Basically, a *dynamic classificational ontology* is just a collection of interrelated terms, but we are really after concepts to which those terms refer and interrelationships to describe whatever information units we are trying to discuss. We start with a base ontology that describes these information units in a selective way. Perhaps it is given by one database that represents a certain set of research articles we are summarizing, experiments, protocols, and so on. However, as we add new articles or link to new databases, we need to generate a *concept thesaurus* which contains derived associations between concepts and the base ontology. To aid the discovery process of extracting relevant data in response to a query we extend the notion of thesaurus from "synonyms" (two ways of saying essentially the same thing) to "associated terms" which occur together with sufficient frequency that a search for one may fruitfully be enriched by a search for the other. For example, the terms "basal ganglia" and "dopamine" are commonly used together in research articles, so a search for articles on dopamine within a certain context can be improved by automatically searching for articles that use the term basal ganglia in that same context, even if the term dopamine does not appear. Our dynamic classificational ontology is one tool for updating the concept thesaurus and thus the derived ontology as use of the database federation proceeds. It is dynamic because the data are changing, so the ontology and the concept thesaurus will evolve in time, as well. Essentially, we employ a data-mining algorithm which counts concept co-occurrences and then takes advantage of common associations revealed in this way to aid further discovery of material relevant to our queries.

The Future of Publishing

We expect articles to continue to be basic units of scientific communication. More and more journals are now being placed on-line, and in many cases the publishers are allowing authors to augment the relatively short document that would correspond to a conventional

hardcopy published article with various electronic appendices representing, for example, datasets that augment the figures and tables of the article or simulations that generated predictions within the article. This availability of more information on data and models is very welcome, but it is our view that the full utility of such augmented datasets cannot be reached unless the data are systematically embedded in databases structured to aid the integration of data from diverse laboratories and to provide support for data mining. It is thus our hypothesis that future publishing will link articles in electronic Article Repositories to relevant data in Repositories of Empirical Data and relevant models in Model Repositories such as BMW. To this end, we have set up an initial prototype for MOP (Model for On-line Publishing) which indicates some of the ways in which articles may be linked to datasets (Chapter 5.3). The display through the browser of a "journal page" is supplemented by an information area which can display any requested information that augments the basic article. A menu provides general facilities for locating supplementary information, and, when one chooses to explore a particular object, a submenu becomes available which lists the operations that are possible. For example, one may replace a figure, which is just a passive bit pattern, by a pop-up applet which allows one to treat the figure as a computational object. In one example shown in Chapter 5.3, each point on a scatter graph corresponds to a particular cell that has been examined in the given study. Clearly, presenting the data for all these cells within an article would overwhelm the balance of the article; however, the pop-up applet allows one to click on any single point in the display and have the choice of viewing general information (including anatomical data), raw data, or statistical data on the selected cell.

Our Model for On-line Publishing thus illustrates ways in which articles can be enriched by manifold links to the federated databases of neuroscience. In part, the success of our effort depends on the development of a data-sharing ethos. At present, many neuroscientists will not even share preprints, let alone data. They want to share papers only after they have been accepted for journal publication and view data as their personal property. This is frustrating because somebody might publish details about only one or two cells plus an average based on 100 cells when the details on those cells might provide crucial information for systematic understanding or contain outliers whose properties suggest new hypotheses. Improved data sharing requires us to surmount both a technological hurdle to make it easier for people to format, store, and retrieve data and a sociological hurdle in making people want to share data and getting pleasure because others analyzed their data to obtain new insights. A related issue is providing levels of security. Certainly there are some data one will never share, because they involve unpolished preliminary studies or early attempts with a new technique. Then there are the data to be kept confidential while writing up the primary papers. However, we believe that if one publishes a paper with tables and conclusions based on 100 cells, say, then one should let other scientists see the data on those 100 cells and have the opportunity to combine these data with other sets as the basis for new analyses which may yield far-reaching insights. There is nothing that the USC Brain Project can do directly to change the ethos, but we can at least provide tools which will make it increasingly easy to share data once an article is published.

Annotation Technology

Conventionally, we annotate hard-copy documents by marking pages and scribbling comments in the margins. Later, in writing up an article or proposal, we may search through a huge stack of papers to find relevant material and re-read the annotations to get ideas for the new document. Today, various word processors offer the capability to make in-line annotations of particular documents, but this is really no different from a footnote and, of course, requires having one's own copy of the electronic file for the document on which to make annotations. Thus, this approach is too limiting. You may often be reading a document and want to make a comment similar to one you have made on an earlier page. While you may have the tenacity to seek the relevant page if it is within the same document, it may exceed your patience to seek for the appropriate page in a pile of different documents. USCBP has developed Annotator (Chapter 5.4), a new Web-based annotation technology to solve this problem, and many more besides. We start by placing annotations in a database that is separate from the document. This allows a particular annotation to refer to multiple places in one document or even to multiple documents in different databases. Thus, one may solve the above problem by searching the annotation database for similar annotations to check whether or not they should be lumped with the proposed new annotation.

Annotator has the advantage that a single annotation can refer to different portions (as noted earlier, we call them "clumps") of a given document or even to relevant clumps in widely scattered documents, even at different Websites. It also solves the problem of annotating documents without having to own an electronic copy, so long as one has access to it over the Web. We use an extended URL to describe the location of each clump, thus making it possible for the user to retrieve the clump and view it together with the annotation even though no actual changes have been made to the annotated document, residing as it does on a different Website. Our annotation technology modifies the browser so that you can view both the annotations and the document referred to by the annotation, with the clump highlighted.

Once annotations are gathered within a database, rather than being embedded in the text of widely scattered documents, it becomes easy to search the annotations efficiently to bring related information together from these many documents. The extended URL methodology then makes it easy to view the clumps, whose relevance can then be determined.

The database of annotations so formed may be a personal database for private use or may be a database of annotations to be shared, whether between the members of a collaboratory or with a larger public. The important thing is that this technology is not just of interest to neuroscientists but is useful to anyone who uses the Internet. Everybody wants to use annotations; however, within the context of USCBP in particular and scientific databases in general, of particular importance is that a Summary Database is a particular example of an Annotation Database, where now each summary may be viewed formally as an annotation on all the clumps that it summarizes.

The NeuroScholar Project

Chapter 6.3 offers both a general framework for Summary Databases, known as Knowledge Mechanics, and an extensible implementation of this design within the domain of neuroscience (called NeuroScholar) that specifically deals with neuroanatomical connections from tract-tracing experiments. The key notion is that it is not sufficient to read, classify, and summarize a paper and then only record the summary. It is essential to record the reasoning that provides the basis for the chosen summary. The aim is to ensure that interpretations of data should be easy to follow. In particular, the way that data are selected as "reliable" or rejected as "unreliable" should be made explicit.

The process of building a representation of users' knowledge in NeuroScholar is accomplished by devising a suitable ontology or data model for it. Just as an individual's interpretations will change over time, individual users' knowledge models will need to be updated, adjusted, and maybe even completely reinvented. In order to keep track of all changes made to the database, detailed access logs will be maintained. Users will be encouraged to timestamp "versions" of their knowledge representation based on their particular outlook at a particular time.

Object primitives contain the classification of a concept and are classified according to "domain type" and "knowledge type." Domain types are based on a classification of the subject under consideration; in NeuroScholar, this is defined by the experimental method being used to obtain the results that support whatever classification is represented by the object in question. For example, this means that we differentiate between objects defined from "tract-tracing studies" and "electrophysiological studies." Fig. 9 illustrates the most important composite object in NeuroScholar: a so-called *neuronPopulation(interpretation, core)* object. This defines a generalized population of neurons. A type of object of importance in this scheme is the *brainVolume(interpretation, core)* object, which is made up of various subobjects which contain relevant data describing the characteristics of those components. These *objects* are simply regions of brain tissue that are defined in terms of a specific publication's parcellation scheme, providing the geometrical substrate for the data in the system. According to this treatment, brain atlases are considered to be "just another publication" with a parcellation scheme expressed as *brainVolume(interpretation, core)* objects to which we link other publications' *brainVolume* objects with various relations. Each subobject of any given *neuronPopulation(interpretation, core)* object may be linked to a given *brainVolume(interpretation, core)* object in order to express its location.

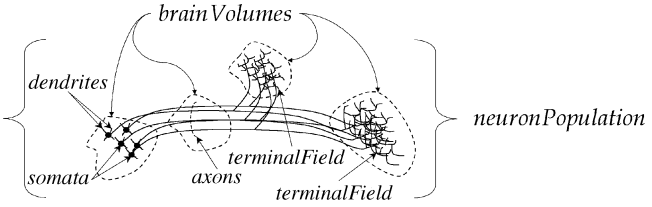

Figure 9 The general *neuronPopulation*(*interpretation, core*) object used in NeuroScholar.

Users' knowledge representations could be compiled into a global summary that represents their understanding of a specific phenomenon or system. This summary could provide the input to secondary analyses either for visualization (as was the case for the neuroanatomical connection databases that were the forerunners of NeuroScholar; Young, 1992) or to investigate organizational properties of the data. Techniques such as nonmetric multidimensional analysis (NMDS), nonparametric cluster analysis (MCLUS), and optimal set analysis (OSA) have been successfully applied to similar problems in the past. Analyses such as these may help expert users interpret their data and make experimental predictions.

Statistical approaches may also be used to generate accurate summaries by searching for optimal solutions that satisfy the many well-defined constraints that exist in the data. For example, consider a situation where the act of comparing data that originate from two different experiments is unreliable, but comparisons between data points from within the same experiment are reliable. In cases such as these, it would be possible to compile a set of metadata comprising all the relevant constraints which could then be analyzed globally to produce global constraints for the whole system. Methods like these were used to calculate finely graded connection weights for the rat visual system (Burns *et al.*, 1996). This was accomplished on the assumption that the density of

labeling in tract-tracing studies is correlated with the anatomical strength of the connection, but different tracer chemicals have different sensitivities. Thus, comparisons of labeling density within the same experiment (or tentatively between experiments that used the same technique) reflected differences in connection strength that could not be inferred from comparisons between experiments. This general approach was also used to convert between parcellation schemes in macaque monkey (Stephan *et al*., 2000).

The Neurohomology Database

The term "homology" is a central one in comparative biology, referring to characteristics of different species that are inherited from a common ancestor. Defining homologies between brain structures requires a process of inference from distinct clusters of attributes. Thus, we anchor our approach to neural homologies (Chapter 6.4) in the concept of *degree of homology*. To define a neural structure, neuroscientists use numerous attributes including gross morphology, relative location, cytoarchitecture, types of cell responses to different means of stimulation, and function. In similar fashion, we employ eight criteria for determining the degree of homology of two brain structures: the morphology of cells within each brain structure and the relative position, cytoarchitecture, chemoarchitecture (neurotransmitters that are found within a brain structure), myeloarchitecture, afferent and efferent connections, and function of each of a pair of brain structures from two species.

If two brain structures have common cell types, chemo- and cytoarchitectonics, and connectivity patterns, then one should expect that those two brain structures have the same function or related functions. This is the case for the primary visual area (area 17). In each major mammalian species, area 17 can be delimited on the basis of myeloarchitecture (heavy myelination) and cytoarchitecture (the presence of a granular layer IV), the presence of a single and systematic visuotopic map, a well-defined pattern of subcortical afferents, small receptive fields, and the presence of many orientation-selective neurons with simple receptive fields.

Chapter 6.4 presents not only a discussion of the homology criteria that can be established between pairs of brain structures across species but also introduces the Neurohomology Summary Database. This database contains three interconnected entities: *Brain Structures, Connections*, and *Homologies*. Fig. 10 shows the modules and relationships that are contained in the database. The various parts of the Neurohomology Database—Brain Structures, Connectivity Issues, and Homologies—can be accessed independently. We have designed the Web interface in independent parts to answer to queries from

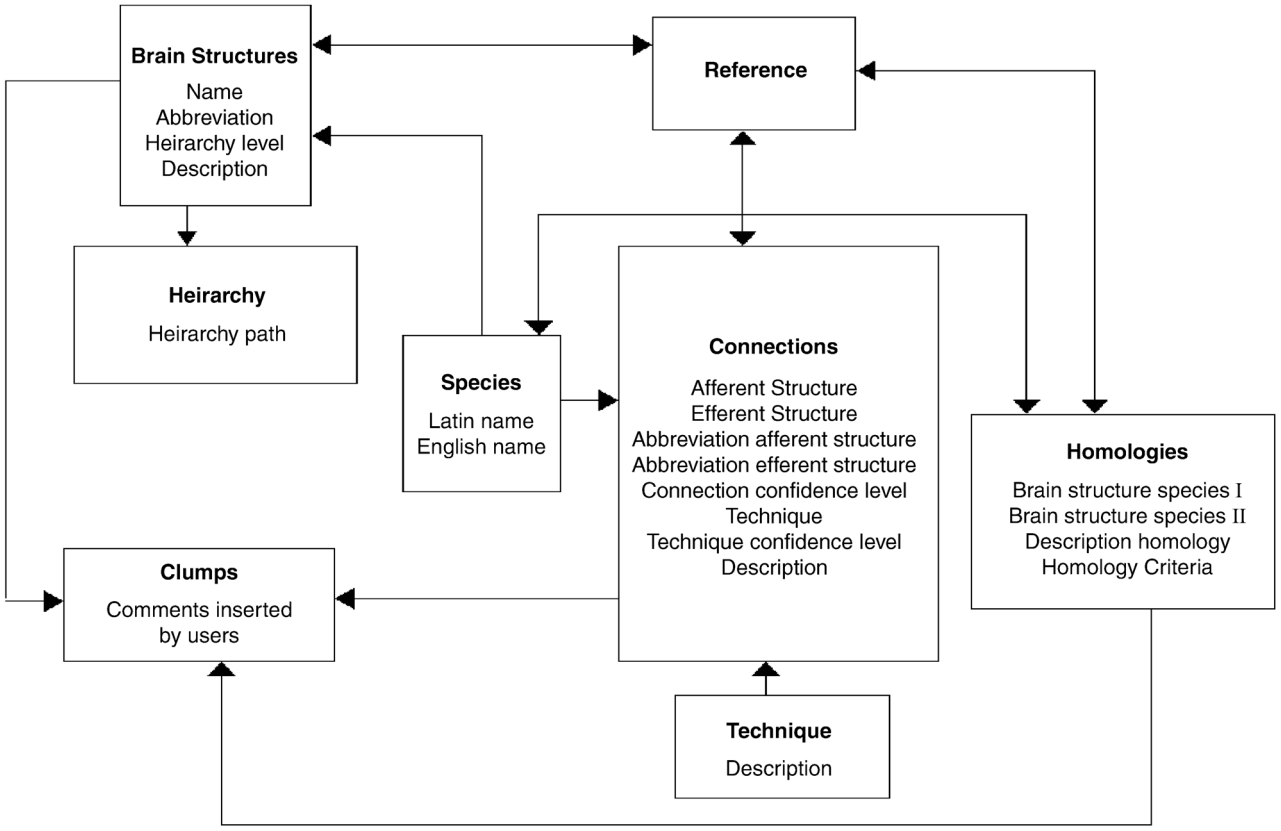

Figure 10 Schematic of the Neurohomology Database structure. A unidirectional connection denotes a 1 to n relationship, while a bidirectional connection denotes an n to n relationship.

a larger category of users. In this way, if users want to find if there is any homology between structures X and Y, from two different species, they can also find the definitions of structures X and Y, according to different sources, as well as the afferents and efferents of these two structures.

Brain Models on the Web

We need to be able to compare different models of related neural systems. At present, a paper describing a model will include graphs of a few simulation results, with an informal claim that the results are more or less consonant with available data. A goal of USCBP research in neuroinformatics is to provide tools that make it easier for researchers to import the code for each other's model and make explicit comparisons. As a result, the community might agree as to what is currently the best model of how the superior colliculus talks to the brainstem and then need the tools to readily modify other models so as to incorporate this model into larger schemes, just as the Fig. 1a model becomes a submodel of the Fig. 1b model. In other cases, comparison of two different models against a set of experimental data may show that one model provides insight into some data, while the other model seems more useful for other data.

The framework for this effort is provided by Brain Models on the Web (BMW; Chapter 6.2), a prototype model repository in which the hypotheses used in defining a model and the predictions made using the model can be linked to Summary Databases (Chapter 6.1) to aid the construction and testing of models. BMW is a step towards providing tools to aid modelers in making comparisons between models and data, learning from them, and constructing a new model that may be seen as an improved version of its precursors and also to provide tools for documenting these improvements—linking models to sets of empirical data—when the new model is placed in BMW. The aim is to have BMW provide modelers with the vehicle to polish their work collaboratively and to involve experimentalists in the collaboration through links to Summary Databases and Repositories of Empirical Data.

In summary: models should be modular so researchers can "mix and match" models of specific subsystems, BMW must provide tools so that comparisons can be made between models and empirical data, and the modeling environment must provide tools for model versioning and for keeping track not only of the new versions as compared to the old but also of the empirical data that justified that migration from one version to another. BMW is based on the idea that you publish models not as take it or leave it entities but with the modular structure made explicit with the links to empirical databases so that anyone with Web access can contribute to model development (subject to usual provisos about access and competence). At present, our task in BMW is simplified because models in the initial tranche "installment" (see *OED*, second edition) are all written in NSL. We also plan to address the issue of using modules that have been developed in different simulation environments. The idea is to develop a "wrapper technology" which enables modules to be "wrapped" in software that provides a uniform communication protocol to allow them to be used together as parts of a larger model. The only catch is that this may reintroduce some of the problems of platform dependence that we have been avoiding by developing a new version of NSL in Java.

Another aspect of BMW is to provide access not only for modelers who are doing the expert work on testing and developing the models but also for non-experts in modeling who are experimentalists who want to test if a model is good enough to justify investing time and resources in testing its predictions. This leads to the idea that we take experimental protocols and build user-interfaces that make it easy to conduct the simulation equivalent of the experiment described by the protocol. We have already developed a number of such interfaces, including that shown in Fig. 2. An interesting research challenge is to come up with a formalized language for protocol definition which makes it possibly to semi-automate the design of corresponding extensions of NeuroCore to store relevant empirical data, as well as the design of interfaces for generating appropriate input and display panels for simulation of experiments based on the protocol.

We provide three levels of membership for models documented in BMW:

1. *Level 1:* All modules of a model are stored in the database with protocol-related interfaces and links to Summary Databases, etc. to support hypotheses, test predictions, and compare models.
2. *Level 2:* Each model receives an exposition in HTML with links to code and to protocol-based interfaces for simulations related to standard experiments.
3. *Level 3:* Exposition and code of model available by ftp.

At present, BMW allows users to view, via the World Wide Web, tutorials, code, and sample simulations of models; in many cases, the model can be imported to a local site for detailed testing. Future work will not only expand the database but also develop tools allowing users around the world to comment on models, develop new versions, and contribute new models to BMW.

BMW will remedy the fact that journal publication of models often give insufficient information to recreate a model in its entirety and offer the results of a limited set of simulation runs. The further testing that BMW makes possible may serve to strengthen our confidence in a model or provide the basis for an analysis of strengths and weaknesses to be used in developing new models with greater validity.

The full development of the USC Brain Project will provide an environment which makes it easy for the user to pass from empirical data to related models and back again. Future BMW-based standards activity will provide modelers using a whole variety of simulation languages (not just NSL) with tools to develop interfaces that make it easy for non-programmers to run basic "experiments" with the models, to add to the database comments on the comparison of simulation results with available empirical data, and to install models, create versions of both models and parameter sets, and to freeze models in various "interesting" states for later analysis under varying conditions. A crucial aspect in all this is to catalyze a truly cumulative style of modeling in neuroscience by facilitating the *re-usability* of modules within current neural models, with the pattern of re-use fully documented and tightly constrained by the linkage with a federation of databases of empirical neuroscientific data.

1.1.6 The NeuroInformatics Workbench

The key contribution of USCBP is that not only has it made great strides in database design, visualization, and modeling for the neurosciences, but it has also designed and implemented an integrated architecture, the NeuroInformatics Workbench, which will enable neuroscientists to use all these methodologies in an integrated neuroinformatics environment. The NeuroInformatics Workbench provides a suite of tools to aid the neuroscientist in constructing and using databases and then linking models and data. At present, the Workbench contains four main components: NeuroCore, for building neuroscience databases; NeuARt, for registering data against brain atlases; NSLJ and related tools, for neural simulation; and Annotator, for building databases of annotations on material distributed across the Web. This section provides a perspective on these varied contributions.

NeuroCore System for Neuroscience Database Construction

Our design of databases to capture time series data, such as records of cell firing or records of behavioral variables, emphasizes the inclusion of *protocol* information about what the experimentalist did to gather these data. NeuroCore is a system for constructing and using neuroscience databases, with the schema data (schema in the database sense of the organizing structure for database tables, etc.) for each "NeuroCore database" being an extension of the "NeuroCore Schema" which we have crafted as a core database schema readily adaptable to meet the needs of a wide variety of neuroscience databases (Chapter 3.2). NeuroCore also provides tools for data entry (JADE), retrieval of data (dbBrowser), inspection of different NeuroCore extensions (Schema Browser), and for data analysis (Datamunch) (Chapter 3.3). For example, if somebody has built a database with a schema that extends the NeuroCore database schema, the Schema Browser allows anyone who accesses that database to easily view the structure of these extensions. NeuroCore has been extended to serve the specific needs of three laboratories: one records data from the cerebellar system of behaving rabbits in a classical conditioning paradigm, another records data from hippocampal slices (Chapter 3.1), while a third stores neurochemical data (Chapter 4.5). The aim is to make these exemplary, so that other neuroscientists will be encouraged to adapt the schema to build federatable databases (which may be private to one laboratory or pool data from a whole community, subject to appropriate access restrictions).

NeuARt NeuroAnatomical Registration Viewer

"NeuARt" was originally short for "Neuroanatomy of the Rat", but it is much more generic than that, a general viewer for atlas-based neural data designed to anchor our work on managing spatial data. It is thus now the acronym for the NeuroAnatomical Registration Viewer. NeuARt is a viewer for atlas-based neural data, with a spatial index manager (Chapter 4.3). In designing NeuARt, the USCBP team emphasized a design closely linked to the Swanson atlas of the rat brain (Chapter 4.2), but the group of Ted Jones at UC Davis is, with the help of a programmer who was formerly a member of USCBP, now extending NeuARt for use with data on the monkey brain. We have also shown how to construct a three-dimensional brain atlas from an atlas of two-dimensional brain sections and to register data against the three-dimensional atlas (Chapter 4.4). This effort undergirds our development of an atlas-based NeuroCore database for neurochemistry (Chapter 4.5). We have also made progress on related Summary Databases, one for neural connections (Chapter 6.3) and one for inter-species neural homologies (Chapter 6.4).

NSL Neural Simulation Language

NSLJ is a modular, Java-based version of USC's NSL Neural Simulation Language, extended by an environment for neural simulation (Chapter 2.2; Weitzenfeld *et al.*, 2000). NSLJ is especially suited for large-scale simulation of neural systems, where we are looking at many brain regions with many, many cells involved. Java implementations enable clients to import a model easily and run it without worrying about what platform they have. Earlier versions of NSL have been used to develop models of basal ganglia, cerebellum, hippocampus, and other neural systems, as well as the interaction of these systems in learning and behavior, and a number of these have now been ported to NSLJ. Our work on EONS, a library of objects for modeling subsynaptic

processes and other neural systems (Chapter 2.3), is at an earlier stage of development, but complements NSLJ by modeling the fine details of how cellular interactions change during LTP and LTD, etc. and has already been used in a number of other studies.

We have also developed a prototype Summary Database linked to the model repository Brain Models on the Web (BMW; see Chapters 6.1 and 6.2). The key idea is that BMW not only stores models but also provides the means to link models to data which support hypotheses and test predictions. All models must be firmly grounded in a careful analysis of a large body of relevant data, wherever it has been developed. Such data will come not only from our own Repositories of Empirical Data but also from the Repositories of Empirical Data of other laboratories where access has been granted and, most strongly, from the published literature at large, which is available in a variety of Article Repositories, an increasing amount being electronic, but much of it still available only in print libraries. The further development of NSLJ and EONS, and their integration with other simulators into a unified multi-modeling environment integrated with BMW and ancillary databases, is an ongoing focus of USCBP research. We will also focus on formal protocol analysis as a basis for providing search engines tuned to use protocols to match and compare sets of empirical and synthetic (model-generated) data. A future challenge is to build a modeling environment where we can go all the way from large-scale system analysis of many interacting regions down to the fine analysis of individual neurons.

Annotator

Annotation technology generalizes the notion of annotation from something placed within a document to a database entry which can refer to annotated data in many other databases. A summary, for example, can be seen as an annotation not on just one document but on all the documents it summarizes. Annotation technology is of general interest for the digital library area, and in fact for anybody who uses the Web, and thus will have much broader applications than simply in neuroscience. USCBP's Annotator allows the creation of paperless annotation databases in which the annotations are kept separate from the annotated document. This has the advantage that a single annotation can even refer to portions ("clumps") of documents at different Websites. It also solves the problem of annotating documents without having to own an electronic copy, so long as one has access to it over the Web. We use an extended URL to describe the location of different clumps, thus making it possible for the user to retrieve the clump and view it together with the annotation even though no actual changes have been made to the annotated document, residing as it does on a different Website.

All the components of the NeuroInformatics Workbench have been introduced earlier in this chapter and are fully described in later chapters of this book. All are available on our Website, some in the form of prototypes which provide proof of concept but are not yet ready for widespread use, while others (such as NSL) have already been released for public use. The reader can consult the "Available Resources" sections of each chapter for a status report as this book goes to press; the current status may always be found by turning to the USCBP Website, http://www-hbp.usc.edu.

Acknowledgments

The work of the USC Brain Project was supported in part by the Human Brain Project (with funding from NIMH, NASA, and NIDA) under the P20 Program Project Grant HBP: 5–P20–52194 for work on "Neural Plasticity: Data and Computational Structures" (M. A. Arbib, Director).

References

Arbib, M. A., Ed. (1995). *The Handbook of Brain Theory and Neural Networks*. A Bradford Book/The MIT Press, Cambridge, MA.

Bargas, J., and Galarraga, E. (1995). Ion channels: keys to neuronal specialization, in *The Handbook of Brain Theory and Neural Networks* (Arbib, M. A., Ed.). A Bradford Book/The MIT Press, pp. 496–501.

Bookstein, F. L. (1989). Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*. **11(6)**, 567–585.

Bower, J. M., and Beeman, D. (1998). *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation Systems*. 2nd ed.. TELOS/Springer-Verlag, Berlin/New York.

Burns, G. A. P. C., and Young, M. P. (2000). Analysis of the connectional organisation of neural systems associated with the hippocampus in rats. *Philos. Trans. R. Soc. London B: Biol. Sci*. **255**, 55–70.

Burns, G. A. P. C., O'Neill, M. A. and Young, M. P. (1996). Calculating finely-graded ordinal weights for neural connections from neuroanatomical data from different anatomical studies, in *Computational Neuroscience Trends in Research* (J. Bower., Ed.). Boston, MA.

Dickinson, P. (1995). Neuromodulation in invertebrate nervous systems, in *The Handbook of Brain Theory and Neural Networks* (Arbib, M. A. Ed.). A Bradford Book/The MIT Press, Cambridge, MA, pp. 631–634.

Hill, A. V. (1936). Excitation and accommodation in nerve. *Proc. R. Soc. London B*. **119**, 305–355.

Hines, M. L., and Carnevale, N. T. (1997). The NEURON simulation environment, *Neural Computation*. **9**, 1179–1209.

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol. London*. **117**, 500–544.

Koch, C., and Bernander, Ö. (1995). Axonal modeling, in *The Handbook of Brain Theory and Neural Networks* (Arbib, M. A. Ed.). A Bradford Book/The MIT Press, pp. 129–134.

Lapicque, L. (1907). Recherches quantitatifs sur l'excitation electrique des nerfs traitée comme une polarisation. *J. Physiol. Paris*. **9**, 620–635.

Payne, J. R., Quinn, S. J., olske, M., Gabriel, M. and Nelson, M. E. (1995). An information system for neuronal pattern analysis, *Soc. Neurosci. Abstr*. **21**, 376.4

Paxinos, G., and Watson, C. (1998). *The Rat Brain in Stereotaxic Coordinates*. 2nd ed.. Academic Press, San Diego, CA.

Quine, W. V. O. (1953). *From a Logical Point of View*. Harvard University Press, Cambridge, MA.

Rall, W. (1995). Perspective on neuron model complexity, in *The Handbook of Brain Theory and Neural Networks* (Arbib, M. A. Ed.). A Bradford Book/The MIT Press, Cambridge, MA, pp. 728–732.

Stephan, K. E., K. Zilles, and Kötter, R. (2000). Coordinate-independent mapping of structural and functional data by Objective Relational Transformation (ORT). *Phil. Trans. R. Soc. London B*. **335**, 37–54.

Swanson, L. W. (1998). *Brain Maps: Structure of the Rat Brain*. Elsevier Science, Amsterdam.

Weitzenfeld, A., Alexander, A., and Arbib, M. A. (2000). *The NSL Neural Simulation Language*. The MIT Press, Cambridge, MA.

Young, M. P. (1992). Objective analysis of the topological organization of the primate cortical visual system. *Nature* **358**, 152–155.