NEUROINFORMATICS

An Overview of the Human Brain Project

Edited by

STEPHEN H. KOSLOW
MICHAEL F. HUERTA

National Institute of Mental Health

LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
1997 Mahwah, New Jersey

Lawrence Erlbaum Associates, Inc., Publishers
10 Industrial Avenue
Mahwah, New Jersey 07430

Library of Congress Cataloging-in-Publication Data

Neuroinformatics: an overview of the Human Brain Project / edited by Stephen H. Koslow and Michael F. Huerta.
p. cm. — (Progress in neuroinformatics research ; v. 1)
Includes bibliographical references and index.
ISBN 0-8058-2099-X (cloth : alk. paper)
1. Human Brain Project. 2. Neuroinformatics. 3. Neurosciences. I. Huerta, Michael F. II. Series.
[DNLM: 1. Human Brain Project. 2. Databases, Factual. 3. Neurosciences. WL 300 N4938157 1996]
QP376.N486 1996
612.8'2—dc20
DNLM/DLC
for Library of Congress 96-23836
CIP

Books published by Lawrence Erlbaum Associates are printed on acid-free paper, and their bindings are chosen for strength and durability.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

9

A Structural Information Framework for Brain Mapping

James F. Brinkley
Loyd M. Myers
Jeff S. Prothero
Greg H. Heil
Jay S. Tsuruda
Kenneth R. Maravilla
George A. Ojemann
Cornelius Rosse
University of Washington, Seattle

The long-term goal of this work is to build an information system for managing, visualizing, and integrating brain mapping data. The data are organized around a structural information framework being developed as part of the Digital Anatomist Program at the University of Washington (Brinkley, Prothero, Prothero, & Rosse, 1989). *Structural information* is defined here as information about the physical organization of the body. Because structure is a useful foundation for understanding function, it is believed that such a structural information framework will be a rational basis for organizing other types of biomedical information, including functional brain mapping data.

The prototype system is designed to manage and integrate data about human language, as an example of brain mapping data that are both important and complex. The initial system will manage cortical stimulation data obtained at the time of neurosurgery for left temporal tumors or intractable focal epilepsy. Later versions of the system will deal with other types of data, including electrocortigraphy (ECoG), optical and functional imaging, positron emission tomography (PET) imaging, and intracellular recording, with the long-term goal of developing models of language organization that integrate these diverse and sometimes contradictory sources of information.

Within the structural information framework, the language data will be organized both spatially, in terms of a patient-specific three-dimensional model obtained from high-resolution magnetic resonance imaging (MRI), and symbolically, in terms of well-defined terminology and semantic relationships that will be developed as extensions to the Unified Medical Language System (UMLS) of the National Library of Medicine (NLM; Lindberg, Humphreys, & McCray, 1993).

Once the organizational and visualization methods have been implemented, the information system will be used to: (a) develop probability maps for use in preoperative surgical planning; (b) provide a bridge between surgical stimulation data and other language data obtained from functional MRI, MRI spectroscopy, or PET; and (c) test hypotheses that can help one understand language organization in the brain.

NEED FOR INFORMATICS IN CORTICAL LANGUAGE MAPPING

The initial prototype is based on electrical stimulation mapping data obtained at the time of left temporal neurosurgery for tumors or intractable epilepsy. Stimulation data obtained from more than 300 cases at the University of Washington show that cortical language organization is both more variable and extensive than the traditional Broca–Wernicke model implies (Haglund, Berger, Shamseldin, Lettich, & Ojemann, 1993; Ojemann, Ojemann, Lettich, & Berger, 1989). For this reason, stimulation mapping is performed on most left temporal patients to plan the resection. These clinically motivated data provide a rich source of information for postoperative analysis, and can further the understanding of language organization in the brain. However, even in this circumscribed area, the amount of data became too large and complex to manage and analyze by previous manual techniques.

The primary stimulation data used for surgical planning at the University of Washington are obtained with an object-naming task, although other tasks are used on a subset of the patients. Once the initial craniotomy is performed, a series of numbered tags are placed on the exposed cortical surface. The patient is awakened and shown slides of easily recognizable common objects, such as boats, trees, or planes. For every other slide, the neurosurgeon applies a small (1.5–10 mA) current to each of the cortical areas marked by a numbered tag, for an approximate total of three stimulations for each site. If during at least two stimulations the patient is unable to name the object, although he or she is able to name the object when the current is removed, the site is called an *essential language site,* and is avoided during the resection. These sites are called essential language sites because: (a) resecting tissue close to such areas usually results in postoperative aphasia, (b) avoiding them by 1.5 to 2 cm avoids such language disturbance, and (c) all aphasic syndromes include anomia (Haglund 1993).

Prior to the resection, a photograph is also taken of the exposed cortical surface (Fig. 9.1A). In an analysis of 117 patients (Ojemann 1989), a piece of tracing paper was placed over the photograph, and the stimulation sites, as well as the major cortical landmarks visible on the photograph, were sketched onto the paper. Multiple patients were related by establishing a surface coordinate system based on the lateral and central fissures, dividing the surface into zones relative to these fissures, and counting the number of essential sites in each zone, as a percentage of the number of stimulated sites.

Statistical analysis of these histograms provided suggestive evidence that females have more essential language sites than males, that there are different patterns of language sites in the two sexes, and that there is a correlation between verbal IQ and the distribution of sites.

These empirical correlations lead to speculation that the distribution of language sites may have functional significance, and that, because cortical anatomy is so variable, perhaps cortical anatomy is related to language, and therefore behavioral variability. However, to test these hypotheses, it is necessary to develop more accurate mapping techniques, and to integrate these data with PET, functional MRI, and behavioral data. To do this, the stimulation data must be related to MRI, and database and statistical techniques must be developed for managing, analyzing, and visualizing large amounts of information.

PROGRESS

This work is based on a collaboration among three groups at the University of Washington: (a) neurosurgeons (Ojemann) who provide the data and have contributed much to the understanding of cortical organization of language in the human brain, (b) neuroradiologists (Maravilla) who are developing high-resolution MRI for brain topography and the associated vasculature, and (c) structural informatics (Brinkley 1991) researchers in the Digital Anatomist program (Brinkley, Myers, Heil, Prothero, and Rosse), who are developing the structural in formation framework for managing and integrating these two kinds of data.

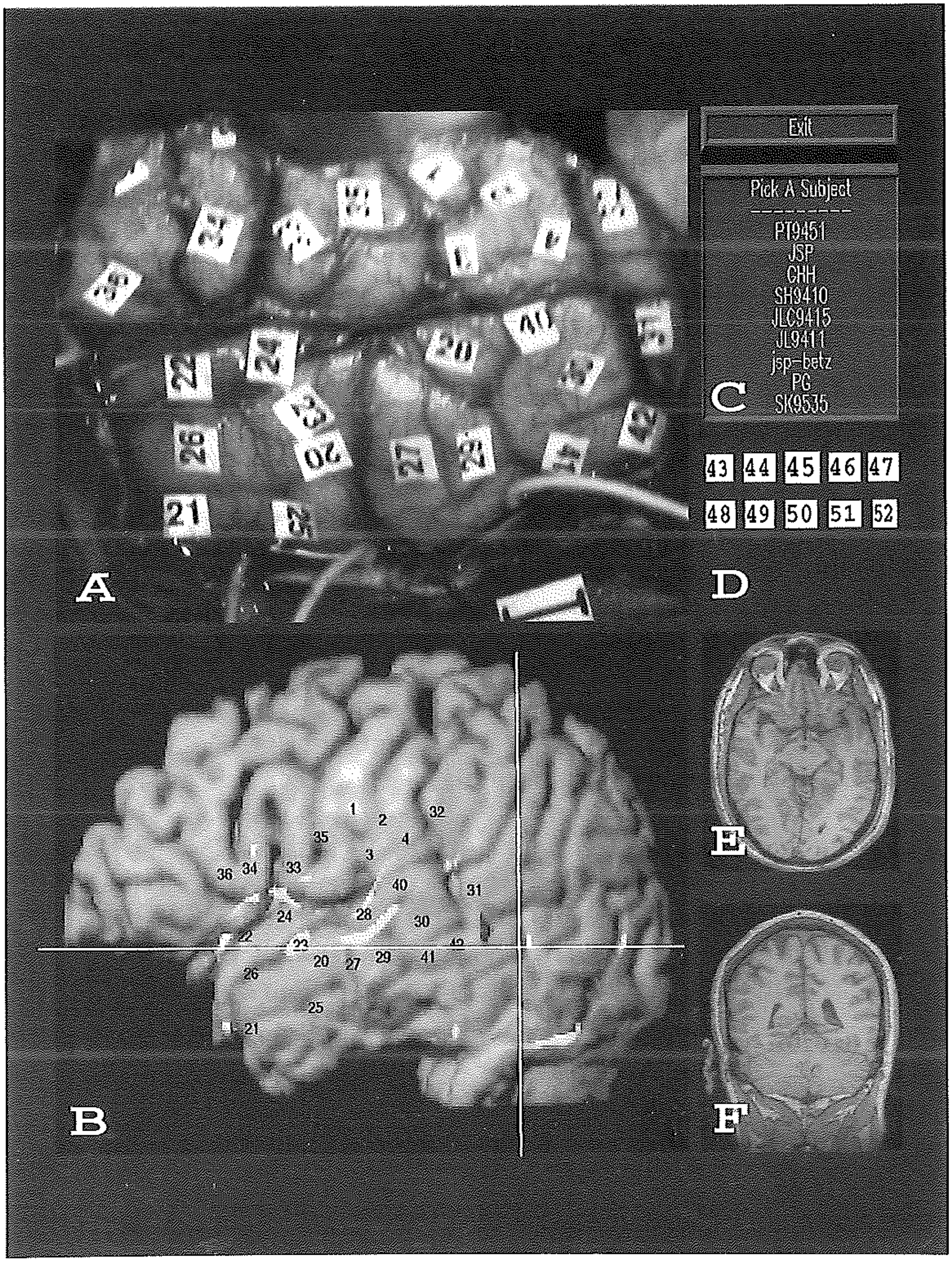

FIG. 9.1. (See plate 16). Interface for the mapping module of the Brain Mapper. (A) Intraoperative photograph. (B) Rendered left temporal surface with arteries (red), veins (blue), and mapped language sites. (C) List of patients obtained from the symbolic database. (D) Palette of numbers to drag onto rendering for mapping. (E) Axial slice through the anatomy series, corresponding to the position of horizontal line in the rendering. (F) Coronal slice through the anatomy series, corresponding to the position of vertical line in the rendering.

During the approximately 1½ years that this project has been funded, the overall architecture of the structural information framework for brain mapping has been designed, and parts of all the component modules have been implemented, although the modules do not yet work together in an integrated fashion. The primary concentration has been on developing the tools for visualization and mapping because there is no point in developing an organizing framework if there are no data to organize. Now that some of the tools are in place, they have begun to be used to acquire patient data, and other modules have been implemented in the framework. The following sections describe the overall design of the structural information framework, followed by the design and current status of the individual modules.

The Structural Information Framework

The structural information framework for brain mapping is part of the Digital Anatomist framework being developed to solve many problems in structural informatics (Brinkley, Eno, & Sundsten, 1993; Brinkley et al., 1989). Figure 9.2 shows the client–server organization of this framework, in which application programs (top two rows) designed to solve specific problems in clinical medicine, research, and education access a set of structural information resources (bottom row) by means of one or more servers (middle row). This overall organization is a conceptual design that drives the development of specific applications. The modules being developed specifically for the cortical language mapping problem include the Brain Mapper, Brain Map Browser, and components of all the server modules. This figure is described from the bottom up.

Structural Information Resources. The four information resources shown along the bottom row of Fig. 9.2 correspond to the classification of structural information into four categories. These categories draw distinctions along two axes: (a) data versus knowledge, and (b) spatial versus symbolic information. *Structural data* are information about an individual (e.g., an individual cortical language mapping patient). *Structural knowledge,* by our definition, is a model that abstracts data about a collection of individuals, such as a probability map of cortical language sites as determined from multiple patients. *Spatial information* deals with the geometric attributes of entities, and has dimension (e.g., the MRIs for a patient). *Symbolic information* deals with nonspatial attributes (e.g., the names of anatomic structures and their semantic relationships). Traditionally, spatial information is represented in

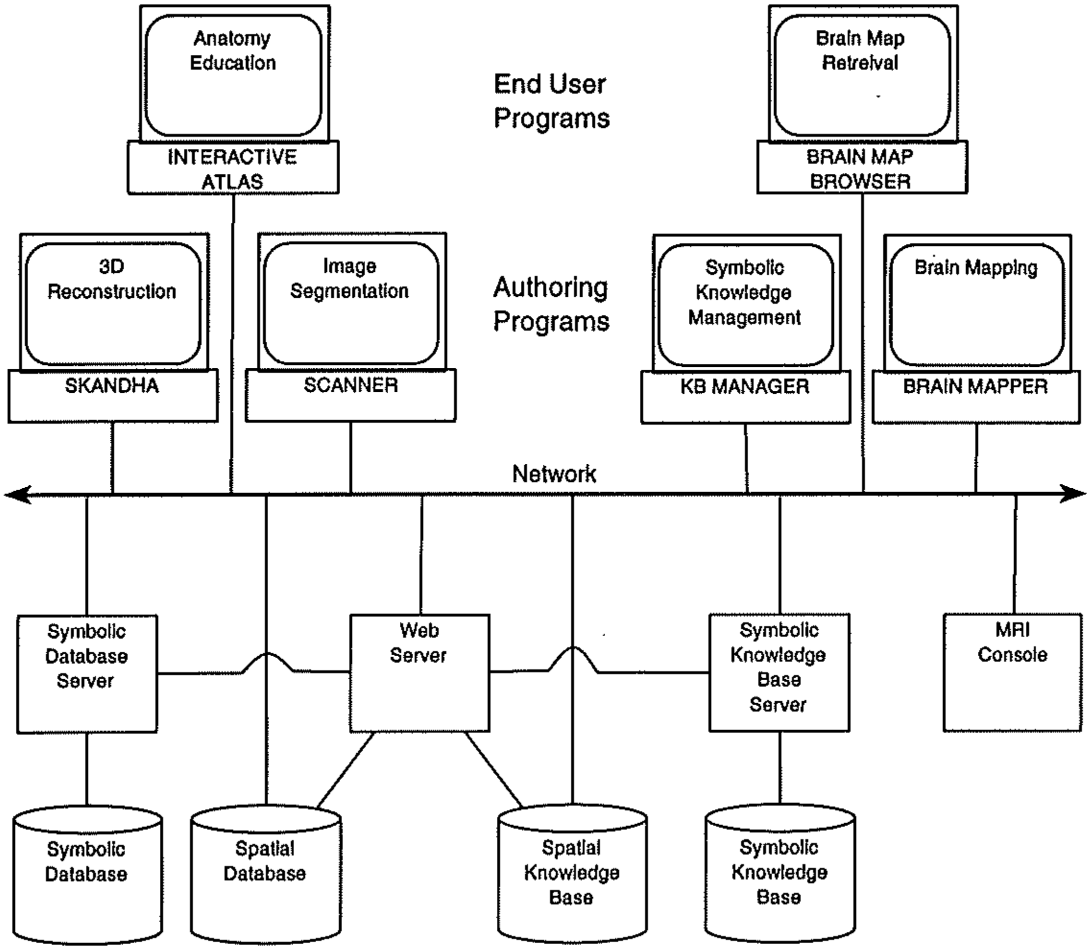

FIG. 9.2. Overview of the structural information framework. Application programs in the top two rows access over the network a set of structural information resources in the bottom row, via servers in the third row. Application programs being developed specifically for this project are the Brain Map Browser and the Brain Mapper.

images, whereas symbolic information is represented in the form of text.

The specific information associated with each of these resources is determined by the application. For cortical language function, the mapping data are stored as files in the spatial database and tracked by means of tables in the symbolic database, whereas terminology and semantic relationships are stored in the symbolic knowledge base. In later work, the spatial knowledge base will contain generalized models obtained from multiple patients.

Structural Information Servers. The servers provide Internet access to the structural information resources, which at some level are all stored as files on Unix servers. For this and many (but not all) current applications, the spatial data and knowledge are directly network file system (NFS) mounted on the local area network, where they are available to application programs running locally. The symbolic information is accessed via database and knowledge base servers, which, in addition to providing remote access to the information (outside the local network), also provide a level of information abstraction so that the client programs need not be encumbered by the details of the database or knowledge base implementation. Recently, we have also begun incorporating World Wide Web (WWW) servers, which either directly access the files in the spatial database and knowledge base, or contact the symbolic servers in response to Web requests.

Figure 9.2 also shows the radiology MRI console because the console is accessed as a server to transfer MRI data from radiology to the Digital Anatomist program.

Application Programs. The top two rows (above the network line) in Fig. 9.2 represent the application programs currently being developed, as part of this or other funded work. The top row represents programs designed for end users. The second row represents programs designed for authors—domain experts who are adding content to the structural information resources.

The original proposal for this project envisioned a single cortical brain mapping application program, running on a high-performance graphics work station, which would combine both brain mapping and brain map retrieval. However, further discussion with the proposed end users (the neurosurgeons) established the need to divide the application into two components: the Brain Mapper and the Brain Map Browser.

The Brain Mapper runs on a high-performance graphics work station, and is used locally to generate MRI visualizations and mappings of language sites onto the rendered cortical surface, storing the results in the databases. The Brain Map Browser, in contrast, is a World Wide Web (WWW)-based application that accesses the mapping data via a common gateway interface (CGI) package called from a standard WWW server. The advantage of this approach is that the mapping data can be made available to anyone on the Internet with a standard WWW browser. Because only a few authors will be using the Brain Mapper, only one or two high-performance work stations will be needed, whereas the WWW based approach can allow an almost unlimited number of users to access the data once they have been mapped.

The other authoring programs shown in the figure include Skandha, for reconstruction and surface display of anatomic objects from serial sections (Prothero & Prothero, 1989), Scanner, for image segmentation using spatial knowledge of anatomy (Brinkley 1993a, Hinshaw, Altman, & Brinkley, 1995), and Knowledge Base (KB) Manager, for management of the symbolic Knowledge Base (Rosse, Ben Said, Eno, & Brinkley, 1995a). The other end user program is the Interactive Atlas, for image-based anatomy education (Bradley, Rosse, & Brinkley, 1993; Brinkley, Eno, Sundsten, Conley, & Rosse, 1995). Components of these other programs are useful for the brain mapping application.

The following sections describe the design and current status of the relevant components in more detail, beginning with the Brain Mapper because the most effort has been placed on this module. As of this writing, all modules have been at least partially implemented, but they do not yet work together in a seamless fashion.

Brain Mapper

The goal of the Brain Mapper module is to generate three-dimensional coordinates of cortical language sites, in terms of a three-dimensional MRI-based spatial model of the patient's own brain. Once the sites have been related to the MRI model, they can be related to other patients or to other modalities by means of Talairach (Talairach & Tournoux, 1988), surface-based (Ojemann et al., 1989), or deformable coordinate systems. They can also be related to functional MRI or MRI spectroscopy studies done on the same patient at a later time.

In the long term, the language coordinates can be obtained using a direct three-dimensional locating device to intraoperatively map language sites onto a three-dimensional MRI rendering of the patient's own brain. However, because currently there is not access to an accurate three-dimensional locator, the initial approach is to generate a rendering of the patient's left temporal cortical surface (Fig. 9.1B) that matches as closely as possible the intraoperative photograph taken at the time of surgery (Fig. 9.1A). This rendering is visually matched to the photograph, and the stimulation sites are mapped from the photograph onto the rendering. The two-dimensional mapped coordinates are then related to the three-dimensional coordinate system of the MRI model by coordinate transforms retained in the rendering.

In the protocol being developed the patient is sent to the radiology department the day before surgery, where a series of MRI studies are obtained. Following transfer of the images over the network, computer analysis in the Digital Anatomist program replaces manual sketches previously done in the postoperative analysis. Details of cortical anatomy are obscured at the time of surgery, therefore the surgeons use cortical blood vessels as an important means to orient themselves. Thus, it is important in the rendering to include the arteries and veins. This is done in the imaging protocol by acquiring three separate MRI volumes: one optimized for surface anatomy, one optimized for veins, and one optimized for arteries. These three volume data sets are then combined to produce the rendering.

The following sections describe the image acquisition protocol, as well as the current postoperative methods for generating the rendering and mapping. This procedure is shown schematically in Fig. 9.3. The boxes in Fig. 9.3 represent data that are or will be recorded in the symbolic database described later.

Image Acquisition and Transfer. For cortical language mapping, all studies are performed on a whole-body 1.5 Tesla MRI scanner (SIGNA, General Electric Medical Systems, Milwaukee, Wisconsin). The patient's head is positioned in a specially designed, close-fitting, birdcage-type RF head coil that provides an increase in signal-to-noise of approximately 40% compared with the standard quadrature head coil (Hayes, Tsuruda, & Mathis, 1993). Following a fast localizer to confirm patient position, three sequences are obtained. The first sequence is a volume image of the entire brain, from which the cortical anatomy is obtained. The other two provide detailed maps of surface arteries and veins, which are used as landmarks for localizing the exposed cortical surface during neurosurgery:

Three-Dimensional SPGR. This sequence is a three-dimensional (volume) gradient recalled echo with a spoiler gradient (SPGR). This study is used to obtain a T1-weighted image of the entire brain, which results in detailed images of the ventricles, cisterns, surface anatomy of the sulci and gyri, together with excellent visualization of gray and white matter.

Two-Dimensional TOF MRI Venogram. This sequence is a two-dimensional time-of-flight (TOF) MRI angiogram that utilizes a gradient recalled echo image optimized for flow-related enhancement of the cortical veins, which appear bright on a dark background. This study is performed in a sequential mode in the axial plane to increase inflow enhancement and to reduce inplane saturation effects on the slowly flowing blood in the cortical veins. An inferior spatial presaturation pulse is applied at the skull base to suppress arterial signal. The entire venous system of the brain is imaged with this technique.

Three-Dimensional MOTSA MRI Arteriogram. This is a three-dimensional (volume) gradient recalled echo pulse sequence that is obtained using multiple overlapping thin slabs (MOTSA) to improve sensitivity for inflow of unsaturated arterial blood and a variable flip angle excitation pulse (RAMP Excitation, GE Medical Systems, Milwaukee, Wisconsin) to reduce arterial saturation artifact. This MRA

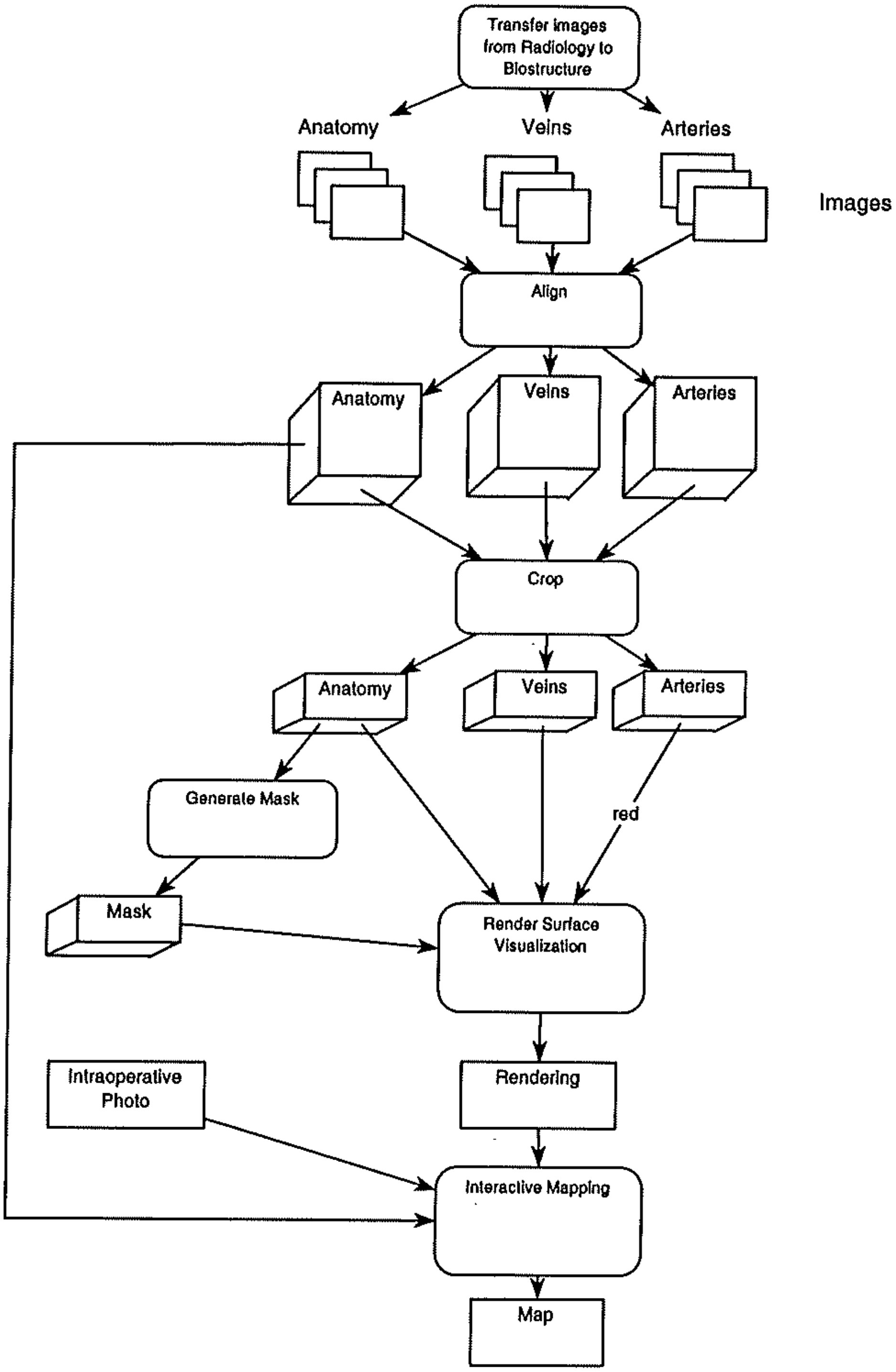

FIG. 9.3. Data flow for visualization and mapping. Images are transferred from radiology, then aligned and cropped. The cortex is segmented in the anatomy volume to generate a mask, in which all voxels outside the mask are transparent. Nontransparent voxels from the three tissue types are combined to produce a rendering, which is then interactively combined with the intraoperative photo to generate a language map. Selected intermediate files are recorded in the symbolic database.

study is performed at high resolution to visualize small arterial structures that appear bright on a dark background. A superior saturation pulse is applied to suppress signal from venous flow. The study encompasses the major arteries at the base of the brain, and extends up to the roof of the lateral ventricles. It is centered through the Sylvian fissure region to visualize small middle cerebral artery branches in this region.

All data sets are performed with an 18-cm field of view, which is referenced to the magnet isocenter (an imaginary point in space within the center of the magnet that represents the intersection of the three orthogonal planes) to allow registration of the data sets for image postprocessing. The patient is immobilized using stiff sponges wedged between the head and the sides of the close-fitting head coil. For the purposes of this study, it is assumed that patient motion between the three MRI studies, which takes approximately 50 minutes, is negligible. This assumption is expected to be valid because these patients are generally cooperative and highly motivated. Once the images have been acquired, they are stored on a GE optical disk in the radiology department, then transferred over the network to computers in the Digital Anatomist program.

The Digital Anatomist computer hardware used for visualization is a Silicon Graphics Indigo^2 Extreme with 96 megabytes of memory, which is connected via Ethernet to smaller Silicon Graphics Incorporated (SGIs) and several NeXT computers. The visualization software that together comprise the Brain Mapping module consists of Advanced Visual Systems' AVS (AVS 1994), a commercial data visualization package, and custom software based on the Slisp programming environment (Brinkley & Prothero, 1996). This custom software is currently used for alignment and mapping, whereas AVS is used for the other processing steps in Fig. 9.3.

Alignment and Cropping. Because each of the three studies is acquired in a different orientation and spatial resolution, the volumes must be aligned and resampled to uniform voxel sizes before further processing. Routines were developed to automatically extract, from the MRI headers, the information necessary to resize and align the volume data sets. The resampled volumes are written to disk as AVS three-dimensional volume fields, which include the necessary information for locating each voxel in the fixed magnet coordinate system. The aligned volumes are then loaded into an AVS network used to interactively crop the volumes to the left temporal area. Cropping is necessary in the current version of AVS because of its inefficient use of memory. However, the coordinate information is retained in the cropped volumes, so each cropped voxel may still be related to the magnet coordinate system.

Segmentation. To accurately map the language sites onto the cortical surface, it is necessary to segment the cortex from the rest of the image volume. The segmented surface can also be used as a mask that eliminates structures external to the cortex that might otherwise obscure the image rendering. The current approach to cortical segmentation is to use an interactive region-growing algorithm ("region grow") implemented in an AVS network (Myers & Brinkley, 1995).

The premise of region grow is that at least some portion of each object in the volume space is fairly homogeneous. Therefore, it attempts to find these initial areas by looking for voxels having a variance below some threshold. Regions are then "grown" recursively from these seed voxels. As regions grow, the algorithm also begins to consider the mean of the growing region, and adjacency constraints of other voxels in the region. More formally, for a given voxel, v, whose 26-adjacent neighborhood, N, contains at least one voxel belonging to a partially grown region R, v will be included in R if $cost < 1$ where:

$$cost = \left(\frac{1}{T_R}\right)\frac{(I_v - \bar{x}_R)^2}{\sigma_R^2}\,w + \frac{\sigma_N^2}{T_N}(1 - w) \tag{1}$$

where

I_v = intensity of the given voxel, v
T_R = threshold for region, R
T_N = threshold for neighborhood, N
$\bar{x}_R$ = mean of voxels in R
σ_R^2 = variance of voxels in R
σ_N^2 = variance of voxels in N
w = a weighting factor

The weighting factor, w, is determined by three user-specified parameters: a voxel counter g_{min}, a second voxel counter g_{window}, and a percentage P_{gray}. Let n_R be the number of voxels in R, then w is as follows:

$$w = \begin{cases} 0 & \text{if } n_R < g_{min} \\ P_{gray}\left(\dfrac{n_R - g_{min}}{g_{window}}\right) & \text{if } g_{min} \le n_R \le g_{min} + g_{window} \\ P_{gray} & \text{if } n_R > g_{min} + g_{window} \end{cases} \tag{2}$$

This causes regions to grow initially based only on local smoothness, but, beyond $n_R > g_{min}$, to consider progressively the voxel's intensity relative to R up to a fixed percentage, P_{gray}.

By adjacency constraints, v will definitely be included in R if the number of adjacent voxels belonging to R exceeds a user-defined threshold (typically 16–18). Conversely, v will definitely be excluded from R if the number of adjacent voxels belonging to R is less than the user-defined minimum (typically 2–4). The minimum adjacency constraint limits R's ability to extend itself in a single line of voxels, whereas the maximum adjacency constraint limits R's ability to omit a single voxel, which may be surrounded geometrically by voxels already contained in R.

The output of region grow is a volume in which each voxel contains an integer label (1–255) indicating to which object it belongs. AVS graphical display tools are then used to choose the label corresponding to the cortex. If no region corresponding to the cortex is visible, the user changes the parameters of region grow and runs the region-growing network again. In the patients imaged to date, it has always been possible to generate a reasonable cortical region in this way, although considerable experience is required to do so. The segmented cortical region is saved as a mask volume.

Rendering. The mask, cortical, venous, and arterial data sets are loaded into an interactive AVS script, from which the user controls the rendering process. The initial mask is morphologically processed to produce three separate region-of-interest shells, which are applied to their respective data sets. This allows the user to eliminate deep vessels that might otherwise appear in the rendering as surface features. For each tissue volume, the user interactively determines a gray-scale threshold to ensure that extraneous information is excluded from the final image.

A two-dimensional image is produced for each tissue type using a gradient-shading ray-tracer contained in AVS. Because gradient calculations are especially susceptible to noise, the gradient is averaged over each 27-voxel neighborhood to produce a more uniform lighting. The user manipulates the direction of the light source, composition of lighting (ambient vs. diffuse vs. specular highlights), surface color, and gloss for the best result. The final rendered image is produced by compositing the resulting 3 two-dimensional images.

In addition to the three RGB values for each pixel in the rendered two-dimensional image, a fourth channel records the z coordinate of each pixel in the image as obtained from the mask. Because each pixel is part of the cortical surface, this information, plus the x–y coordinates implied by the location of the pixel in the image array, is enough to calculate the three-dimensional MRI machine coordinates of each point on the rendered surface. The completed image is saved to disk as a depth rendering. A more detailed description of the visualization process may be found in Myers and Brinkley (1995). Figure 9.1B is an example rendering from one of the surgical patients.

Mapping. The depth rendering, plus the scanned intraoperative photograph, is loaded into an interactive mapping module of the Brain Mapper, shown in Fig. 9.1. The technician visually matches the two images, using all the visual cues that are available, then drags numbered tags from a palette to locations on the rendered image that correspond to the numbered tags in the photograph. This simulates a procedure the surgeons are already familiar with when they place the actual tags on the exposed cortical surface. Lines on the rendered image allow the technicians to relate features on the rendered surface to the original MRI. The completed mapping is saved as a file that associates site numbers with three-dimensional machine coordinates of surface points.

Current Status. Most of the components of the Brain Mapper are currently working, although they must be manually controlled, with communication by means of intermediate files. To date, six patients have been imaged and all of them have been visually matched to the intra-operative photograph. Three normal volunteers have also been imaged.

Brain Map Browser

The Brain Map Browser (top row in Fig. 9.2) is designed to allow retrieval and display of language maps, renderings, and MRI based on various search criteria. Later the Browser will allow multiple patients to be related via Talairach or other coordinate systems, and cortical language maps to be related to other language information available on the Internet.

The Browser will run in a standard WWW browser, which means that all its functionality is derived from a set of custom common gateway interface (CGI) functions (Vetter & Spell, 1994) called by a standard WWW server running within the local area network. The WWW server, by means of the CGI package, has access to the files in the spatial database, as well as the symbolic database and knowledge base servers. The CGI package is modeled after a similar CGI package already built for the Interactive Atlas (Bradley et al., 1995).

At the present time, the Browser consists of a set of hand-generated hypertext markup language (HTML) files (Vetter et al., 1994)—one for each patient or normal volunteer—that point to renderings and photos created by the Brain Mapper and saved as files in the spatial database.

Currently, the files are only WWW-accessible to University of Washington sites because they represent the raw patient data. This method of dissemination has been found to be useful for the collaborating groups at the University of Washington. The decision to release the data to the wider community before the data have been published is a policy rather than technical decision (not a simple decision).

Slisp Software Environment

Before describing the design and current status of the server modules, this section first describes the Slisp software environment (Brinkley and Prothero, 1996), which is the basis for the symbolic knowledge base and symbolic database servers, as well as many of the application modules in Fig. 9.2 (Skandha, Scanner, and the aligner and mapping components of the Brain Mapper).

Slisp is a hybrid programming tool kit designed to combine rapid prototyping with efficient execution. Computationally intensive operations, such as database access or graphics, are coded in C, then encapsulated as primitive Lisp functions. The primitive Lisp functions are "glued" together by interpreted Lisp functions to define a specific hybrid application containing both interpreted and C-coded Lisp. The advantage of this approach is that new ideas may be quickly prototyped in Lisp, then recoded as needed in C once the ideas have been shown to be useful.

Slisp is based on a publicly available Lisp programming system called Xlisp (Betz, 1989), which is a C-coded implementation of a Lisp interpreter and most of the basic Lisp functions. Slisp adds the ability to collect together commonly used C-coded primitives as self-documenting and self-testing modules, which may be selectively compiled into Xlisp. The modules provide Lisp primitives that may be called in the same way as any other primitive Lisp function, and, in particular, may be called from Lisp functions defined by the Lisp DEFUN function, which is used to dynamically create new functions. The combination of C-coded primitive Lisp functions, plus interpreted Lisp DEFUN'D at runtime, constitutes a Slisp hybrid application.

The resulting hybrid application may be run in several different modes: from the command line, embedded in another C program, as a server, and as a client of another Slisp server. Any Slisp program can be run in any of these modes, which means that any of the Slisp modules shown in Fig. 9.2 can be run as a client or a server, and can be embedded in a different C program (usually to create a graphical user interface). A more detailed overall description of the Slisp programming environment is described in (Brinkley & Prothero, in press). The basic Slisp package, including client-server capabilities and documentation for adding new modules, is available by ftp from ftp:\\ftp.biostr.washington.edu\pub\slisp.

Databases and the Symbolic Database Server

The spatial and symbolic databases, and the symbolic database server, are represented by the boxes in the lower left part of Fig. 9.2. Figure 9.4 is a detailed view of these components. The files generated in the

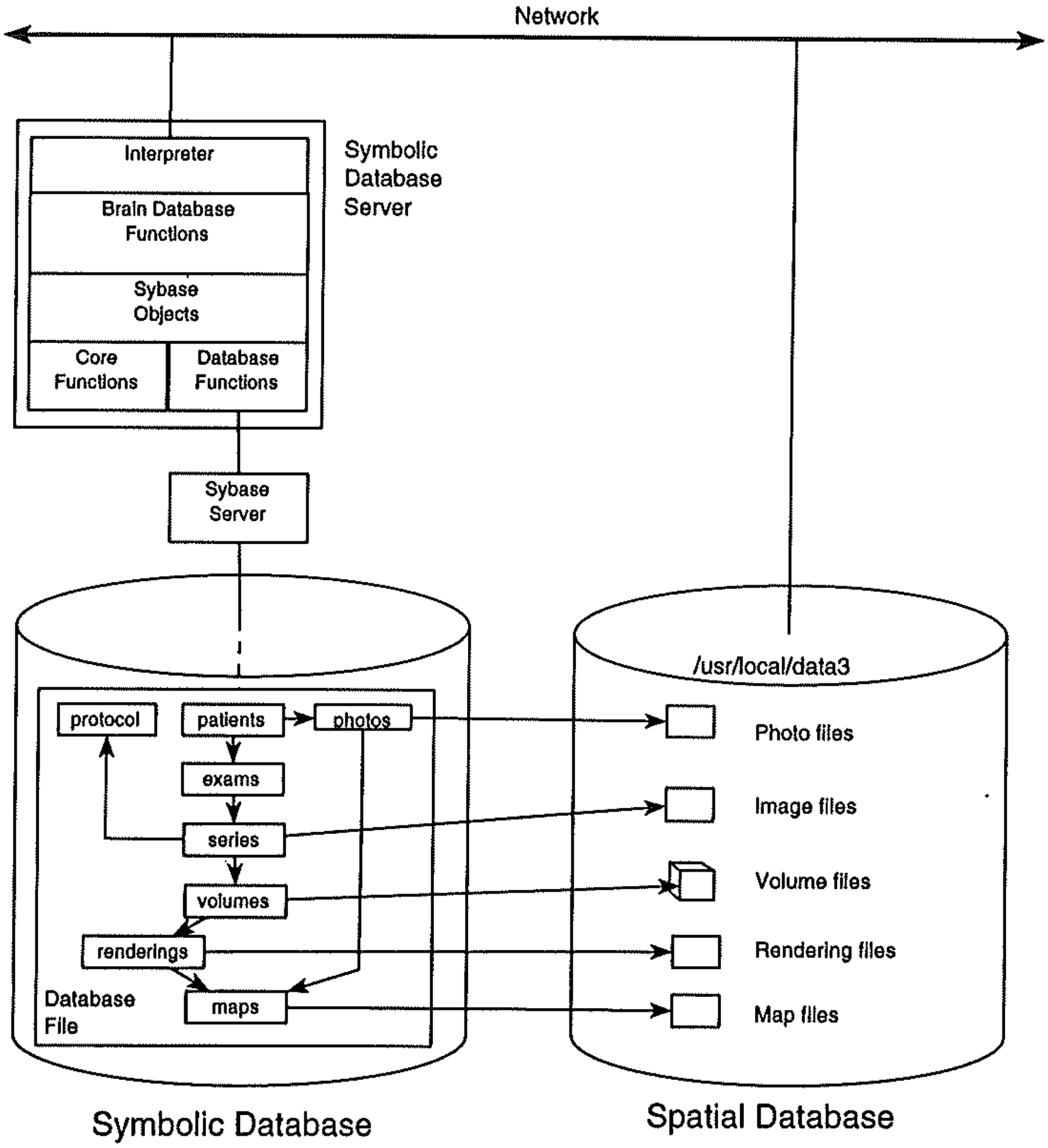

FIG. 9.4. Symbolic and spatial databases. Files generated in the visualization process are stored in the spatial database and referred to by relational tables in the symbolic database. The symbolic database is accessed via the symbolic database server, which is a hybrid Lisp–C program.

imaging protocol shown in Fig. 9.3 are stored in the spatial database and tracked in the symbolic database. For the local environment, these files are mounted on most machines as the Unix file directory /usr/local/data3.

The symbolic database is a relational database implemented in Sybase. Tables in the relational database model the information generated by the imaging protocol (Fig. 9.3), and contain, where relevant, the names of corresponding files in the spatial database. Each patient can have one or more imaging exams. Each exam contains several image series, each of which is acquired using a particular MRI protocol. A series is used to derive an aligned image volume, which in turn is used in combination with other volumes to create a rendering. The rendering is combined with the intraoperative photo to generate a language map.

The relational database is accessed by a commercial Sybase Server, which in turn is accessed by the Slisp-based Symbolic Database Server. The C-coded functions of the Symbolic Database Server include a Lisp interpreter, which parses Lisp commands and either interprets them if they are nonprimitive, or dispatches to C functions if they are primitive (all functions are eventually built up as calls to primitive functions). The basic Lisp commands (CAR, CDR, LIST, etc.) are implemented by a core functions module. The custom database functions module encapsulates a subset of the Sybase database library as a set of Lisp functions. The main such function is (DB-SQL SQL-STRING), which passes a structured query language (SQL) string to the Sybase server and returns a list representing the result of the database query.

The Lisp-level functions are loaded at run time, and provide a higher level interface to the basic DB-SQL function. The Sybase objects module utilizes the Xlisp object system to model the local Sybase database environment as a set of Lisp objects corresponding to the components of the Sybase server environment. For example, the Sybase object has, as one of its instance variables, a list of Sybase servers visible from the local machine. Each server has one or more databases, each of which has a schema consisting of tables defined by columns. Each of these objects has appropriate methods for accessing it, all via the underlying DB-SQL primitive. Methods are called to open a server, to login to it, to create a database, to create a schema from a set of Lisp objects, and to submit queries to the currently connected database.

The Lisp-level module defines specific Lisp functions for accessing the database via the Sybase objects module. For example, once the database has been connected to, a function called (DB-GET-PATIENTS) formulates an SQL query sql-string as "select name from patients" and then sends it to the appropriate database by the method (SEND SYBASE

:USERQUERY SQL-STRING). The USER QUERY method in turn calls DB-SQL with that query, which returns the list of patient id strings.

The Lisp function (DB-GET-EXAMS patient name) can then be called to retrieve the list of image exams for a given patient. This function generates the SQL query "select exam.id from patients, exams where patients.name = patient-name and patients.id = exams.patientid order by exam.id." The client programs need only be aware of the Lisp functions, rather than the more complex SQL queries that are generated by them. In addition, if and when one moves to a different kind of database system, the high-level Lisp commands do not need to change.

Because any Slisp program can be a client to any Slisp server, and because the application programs Skandha, Scanner, and part of the Brain Mapper are Slisp applications, all these programs will have immediate access to the database once the server is configured on the network. Within any of these applications, the Lisp call to retrieve the list of patients is (NET-EVAL '(DB-GET-PATIENTS)), which contacts the Symbolic database server to query the database, then returns, in the application program, the list of patients returned by the server.

For example, the list of patients in Fig. 9.1C will be generated by calling (NET-EVAL '(DB-GET-PATIENTS)) from the mapping module of the Brain Mapper. When one of the patient identifiers is clicked, the database server will return the names of the relevant image and map files in /usr/local/data3, which will then be loaded into the mapper.

All components of the symbolic database are currently functional, although the brain database functions in Fig. 9.4 are only minimally implemented, and the application programs have not yet been connected to the database server. The spatial database in /usr/local/data3 exists as a defined file directory organization, which is maintained manually.

Symbolic Knowledge Base and the Symbolic Knowledge Base Server

The MRI visualization and mapping represents a method for organizing cortical language data in terms of a spatial model. An alternate and complementary method for organizing the data is a symbolic model, which consists of sets of neuroanatomical and functional terms, including synonyms, arranged in hierarchies that capture semantic relationships. If the terms reflect common usage in the literature, they can be employed for relating multiple studies. In addition, if terms can be associated with categories of sulcal variation, hypothesized correlations can be tested among these categories and the distribution of language sites.

The symbolic knowledge base is being developed as an extension to the UMLS Project of the NLM (Lindberg et al., 1993). The most complete neuroscience nomenclature that is currently part of the UMLS is Neuronames, which was developed at the University of Washington (Martin, Dubach, & Bowden, 1990). However, even this nomenclature does not deal with cortical surface anatomy and its variations.

In the anatomy education work, Neuronames was incorporated, as well as gross anatomy extensions being developed as part of an NLM contract (Rosse et al., 1995) into the symbolic knowledge base. The terminology of and relationships among terms are represented as a semantic network that encodes relationships like "the telencephalon is part of the brain," where *telencephalon* and *brain* are terms and *part of* is a link in the semantic network.

Figure 9.5 which is a more detailed view of the knowledge base components in the lower right-hand side of Fig. 9.2, shows how the semantic network is implemented in the symbolic knowledge base. The semantic network is stored in a Sybase relational database file, and accessed via a Sybase server. These files are different than those shown in Fig. 9.4 used to store the symbolic database, but the mechanisms are the same. Terminology is stored in a terms table, and semantic links between terms are stored in a links table.

In addition to the semantic network, the symbolic knowledge base also includes text files, whose names are the names of terms in the semantic network, and which contain additional textual information about the terms not yet encoded in machine readable form, such as *definition, function, blood supply,* and so on.

The symbolic knowledge base server is a Slisp application, like the symbolic database server; in fact, the C components and Sybase objects Lisp component are exactly the same. The only difference is that a file of Lisp-level knowledge base functions is loaded at run time, rather than the file of brain database functions used to define the symbolic database server.

The symbolic knowledge base and server are currently running, and are accessible from any Slisp program—from a custom Macintosh program developed for anatomy education (Brinkley et al., 1993), and from a custom program called *KB Manager,* developed for entering new terminology and relationships into the knowledge base. Access to the knowledge base via the Web has also been demonstrated. The knowledge base is not yet integrated with the Brain Mapper or Brain Map Browser.

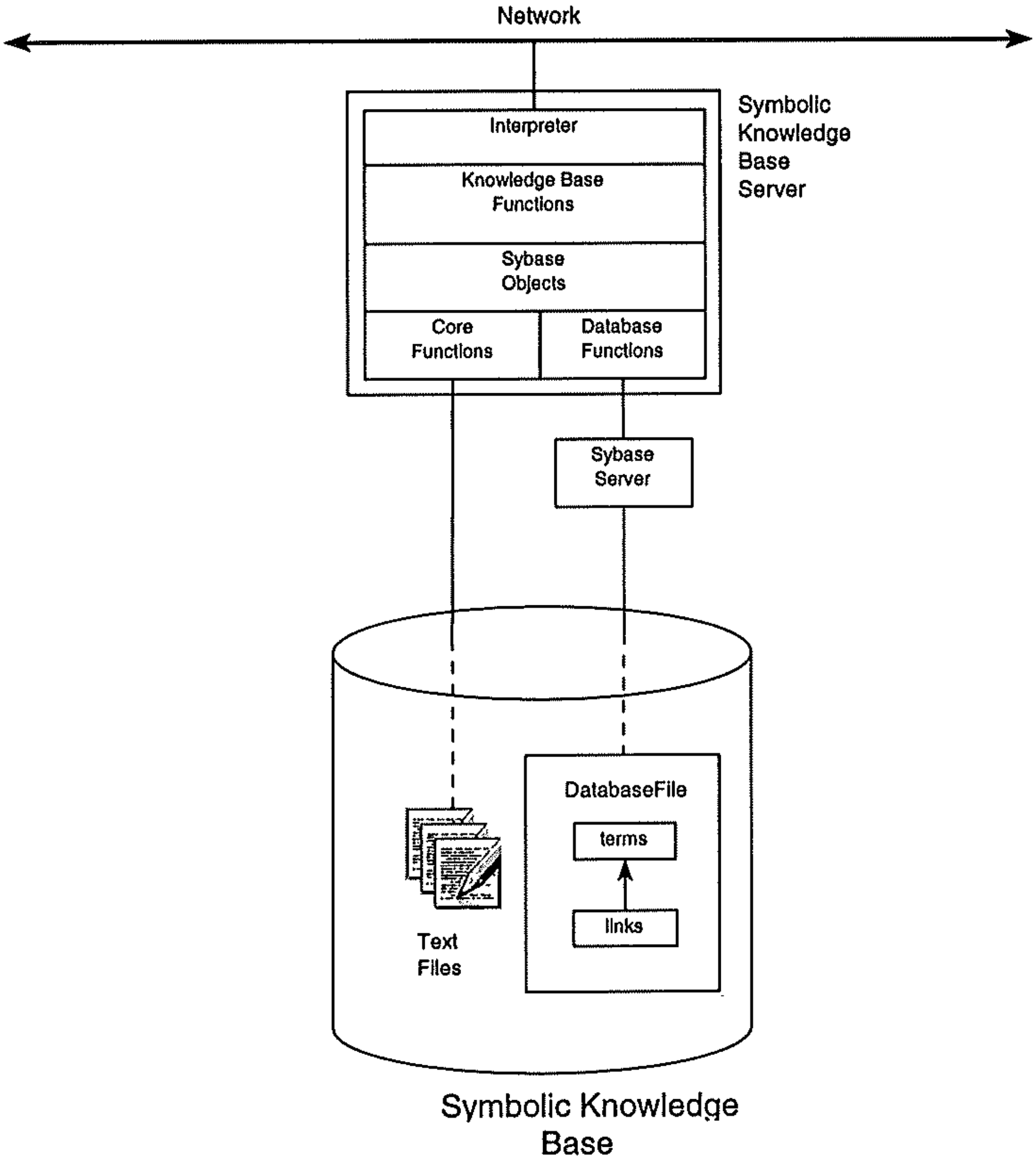

FIG. 9.5. Symbolic knowledge base. The knowledge base consists of terms connected via links in a semantic network, as well as associated text files. The knowledge base is accessed by the symbolic knowledge base server.

The content of the Knowledge Base is currently being augmented with gross anatomic terms as part of a separate contract with NLM (Rosse et al., 1995). The neuroscience component contains only Neuronames. Future work will use the software tools to add appropriate language mapping terminology and relationships.

IMPACT

The previous sections described the design of a structural information framework for brain mapping, and the progress in implementing each of the modules. Although the modules do not yet work seamlessly together, they are sufficiently developed (particularly the Brain Mapper) that we can begin to collect more patient data.

Thus, in addition to further development of the software modules, we are now in the process of validating the mapping approach through the new data collected. If these further data confirm the preliminary data, the system could be used for surgical planning and for testing hypotheses about language organization in the brain. It is at this point that the impact of the system being built will be felt.

Neuroscience Impact

The neuroscience impact will have both a clinical and a basic science component. The clinical impact should eventually lead to better methods for surgical planning. Once a sufficient number of patients have been entered into the database, we will be able to generate probability maps of likely language sites, based on patient characteristics such as age, gender, and verbal IQ. Although these maps will not eliminate the need for stimulation mapping, they may reduce the amount of stimulation needed, or help in the placement of the stimulation sites. In the long run, a virtual reality environment at the time of neurosurgery may be useful, in which the MRI visualization is superimposed on the surgical field, thereby aiding the surgeon in the identification of anatomical features useful for planning the surgery (League, 1995). If this approach is combined with direct three-dimensional locating devices, the stimulation sites will appear immediately on the image.

The basic science impact will be in the area of language understanding, which will come about as researchers develop methods for relating multiple patients, and for relating cortical language sites to language areas found by other modalities. When all these data can be mapped onto common spatial and symbolic models, the language mapping community will be in a position to make sense of the sometimes contradictory data about human language. It will then be easier to develop models for explaining these data—models that will be implemented on the computer and that will directly access the individual data and literature sources explained by them.

Informatics Impact

The information system envisioned for language mapping will require techniques from many areas of informatics, including, among others, multimedia databases, visualization, image analysis, geometric modeling, interface design, client–server systems, and knowledge representation. As researchers attempt to build systems to solve the kinds of problems that arise in the Human Brain Project, they can expect that these attempts will lead to fundamental advances in informatics. In fact, it can be argued that almost all the major advances in informatics have been driven by the need to solve real-world problems, and that if informatics research becomes too far removed from the real world, it becomes increasingly meaningless.

Like the Human Genome Project, which has already engendered advances in areas such as database management and pattern recognition, the Human Brain Project deals with information at the molecular level. However, the Human Brain Project is even more encompassing because, in the long run, it will attempt to organize information at all levels—ranging from the molecular to the anatomic, and from the physiological to the behavioral. The development of generic methods for integrating these highly diverse forms of information will, of necessity, lead to advances in such areas as databases, graphics, artificial intelligence, computer vision, and others.

If the information systems that researchers build aid the development of comprehensive models that begin to explain human intelligence, then this new understanding should be applicable to informatics research on artificial systems that try to emulate human capabilities, such as language or vision, which is the long-term goal of artificial intelligence (AI) research. The new results in AI should, in turn, be applicable to more advanced tools for understanding the human brain, thus leading to a synergistic interaction between intelligent systems based on "wetware" and those based on hardware.

FUTURE DIRECTIONS

As described previously, the accomplishments of the first year and a half of this research include: (a) adaptation of the Digital Anatomist framework to the design of a "structural" information framework for cortical language mapping, (b) partial implementation of all the component modules, and (c) more extensive implementation and testing of the Brain Mapper module. Based on these results, future directions have short-term and long-term plans.

Short-Term Plans

The MRI renderings from the initial set of patients are convincing that the visual matching technique is a useful means for relating MRI to intraoperative language sites, at least until more accurate three-dimensional hardware locating devices are available. Therefore, we are now in the process of collecting more patient data. It is expected that a high proportion of the mappings will be successful, and that these patients will form the initial patient population for the database. Concurrently, we will continue to develop the modules of the information framework shown in Fig. 9.2 so that the modules work together as designed.

Long-Term Plans

Completion of the short-term plans will result in a prototype system for acquiring, managing, and visualizing language data relevant to essential naming sites. This system should be useful for collecting and managing the data that are acquired during neurosurgical language mapping. In addition, the foundation will be in place for extending the basic framework to relate multiple patients, to integrate data from multiple modalities, and to test hypotheses about correlations between sulcal anatomy and language variation.

To relate multiple patients, the ability to record data in terms of both Talairach coordinates and surface-based coordinates (Ojemann et al., 1989) will be added to the language mapper. These features will extend the earlier work (without the extensive manual effort), and relate the language sites to other studies recorded in the Brain Map database developed by Fox, Mikiten, Davis, & Lancaster, (1994). It will also be possible to generate probability maps of likely language sites for a given subpopulation by color coding frequency distributions on these grids, and then superimposing the grids on the patient's MRI rendering.

The language maps will also provide the missing link between surgical mapping studies and other MRI or non-MRI-based functional mapping studies. For example, the success of the visualization procedure on several patients has prompted neuroradiologists (Maravilla and others) to propose the development of protocols for the collection of MRI-spectroscopy data on the cortical language sites—to determine if there are biochemical changes at the language sites that are not present at nonlanguage control sites.

The framework being developed will also allow development of methods for testing hypotheses relating sulcal and language variation. The highly variable nature of the cortical surface tempts one to hypothesize that at least some of the observed language variation may be related to the sulcal variation, and that if sulcal variation could be accounted for, perhaps the observed language variation could be more precisely explained. To do this, methods for mapping sulcal anatomy from one patient to another will need to be developed.

A symbolic approach to this problem is to classify sulcal variants according to a naming scheme, such as that developed by Ono, Kubik, and Abernathy (1990), and then to determine statistically if language variation is related to these sulcal classes. This kind of study will be possible once we have modeled the sulcal variations in the symbolic knowledge base, and created an interface in the Brain Mapper that allows a user to record the class of a particular patient's sulcal anatomy by clicking on the appropriate terms from the knowledge base.

To correlate language variation with spatial descriptors of sulcal variation, it is necessary to segment the cortical surface and then develop methods for matching surface anatomy from one patient to another. The current volume visualization methods do not directly provide cortical surface anatomy in enough detail to allow sulcal matching, although improvements in the region growing method may lead to a better surface.

Because segmentation of the surface is a long-term goal, alternate methods for segmenting the surface are being considered, including isosurface thresholding methods developed by David MacDonald in Alan Evan's group at the Montreal Neurological Institute, and shape-based methods being developed as part of our own image segmentation work (Hinshaw et al., 1995).

For sulcal matching, we hope to utilize some of the deformable modeling work being developed by other institutions involved in the Human Brain Project, including Montreal Neurological Institute (Evans, Collins, Neelin, & Marrett, 1991), University of Michigan (Bookstein, 1989), and Washington University (Christensen, Rabbitt, & Miller, 1994). We may also want to explore the cortical unfolding work at Washington University and the University of Washington (Sherk, 1992) as another potential method for explaining the variation. The distributed information framework should allow us to easily incorporate these other methods into our approach.

The distributed framework being developed should also make it possible for our applications to contact servers developed by other members of the Human Brain Project, as an alternative to incorporating their work in our local environment. Conversely, other groups may want to contact servers developed by us, such as our symbolic database and symbolic knowledge bases.

As an example, one group may develop a good method for warping one brain to another. As an alternative to providing the code to each lab, the group could set up a server that would take as input the two brains to be mapped, and return over the network the result of the match. Other examples might include cortical unfolding servers, segmentation servers, or symbolic knowledge base servers. If such servers are developed, and if standard methods are developed for accessing them via well-defined application program interfaces (APIs), then the Human Brain Project could become a model for distributed information management and processing.

The Human Brain Project, like other information management initiatives, is occurring at an opportune time in the evolution of information technology and communication. Developments such as high-performance desktop work stations, high-speed networks, and the WWW promise that it will indeed be possible to build information systems that can manage the enormous amounts of data being collected about the brain, while preserving the autonomy of individual researchers.

ACKNOWLEDGMENTS

This work was sponsored by Human Brain Project Grant DC/LM02310, and co-funded by the National Institute for Deafness and Other Communication Disorders, as well as the National Library of Medicine. We would like to thank the following individuals for their help in various aspects of this project: Jill Gardner, Cecil Hayes, Mitchell Berger, Etore Lettich, John Sundsten, and Shelley Vogel.

REFERENCES

AVS (1994). AVS User's Guide. Advanced Visual Systems, Waltham, Mass.

Betz, D. (1989). *Xlisp: An object-oriented lisp.* [On-line] Available FTP: ftp.biostr.washington.edu and other Xlisp ftp sites.

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

Bradley, S., Rosse, C., & Brinkley, J. (1995). Web-based acess to an online atlas of anatomy: The digital anatomist common getaway interface. In *Proceedings of the 19th Symposium on Computer Applications in Medical Care.* 512–516.

Brinkley, J. (1991). Structural informatics and its applications in medicine and biology. *Academic Medicine, 66,* 589–591.

Brinkley, J. (1993). A flexible, generic model for anatomic shape: Application to interactive two-dimensional medical image segmentation and matching. *Computers and Biomedical Research, 26,* 121–142.

Brinkley, J., Eno, K., & Sundsten, J. (1993). Knowledge-based client-server approach to structural information retrieval: The digital anatomist browser. *Computer Methods and Programs in Biomedicine, 40,* 131–145.

Brinkley, J., Eno, K., Sundsten, J., Conley, D., & Rosse, C. (1995). A distributed framework for distance learning in anatomy: The digital anatomist interactive atlas. In *Proceedings of the 19th Symposium on Computer Applications in Medical Care,* 972.

Brinkley, J., & Prothero, J. (in press). *Slisp: A flexible software toolkit for hybrid, embedded and distributed applications.* Software—Practice and Experience.

Brinkley, J., Prothero, J., Prothero, J., & Rosse, C. (1989). *A framework for the design of knowledge-based systems in structural biology.* In Proceedings, 15th annual symposium on Computer Applications in Medical Care, 61–65, Baltimore, MD.

Christensen, G., Rabbitt, R., & Miller, M. (1994). 3d brain mapping using a deformable neuroanatomy. *Physics in Medicine and Biology, 39*(3), 609–618.

Evans, A., Collins, D. L., Neelin, P., & Marrett, S. (1991). Warping of a computerized 3-d atlas to match brain image volumes for quantitative neuroanatomical and functional analysis. In *SPIE Vol. 1445 image processing* (pp. 236–246). San Jose, CA: SPIE.

Fox, P., Mikiten, S., Davis, G., & Lancaster, J. (1994). Brainmap: A database of human functional brain mapping. In *Functional neuroimaging* (pp. 95–105). San Diego: Academic Press.

Haglund, M., Berger, M., Shamseldin, M., Lettich, E., & Ojemann, G. (1993). Cortical localization of temporal lobe language sites in patients with gliomas. *Neurosurgery, 34*(4), 567–576.

Hayes, C., Tsuruda, J., & Mathis, C. (1993). Temporal lobes: Surface mr coil phased-array imaging. *Radiology, 189*(3), 918–920.

Hinshaw, K., Altman, R., & Brinkley, J. (1995). Shape-based models for interactive segmentation of medical images. In *SPIE Medical Imaging 1995* (pp. 771–780).

League, D. (1995). Interactive, image-guided, stereotactic neurosurgery systems. *AORN Journal, 61*(2), 360–370.

Lindberg, D., Humphreys, B., & McCray, A. (1993). The unified medical language system. *Methods of Information in Medicine, 32*(4), 281–291.

Martin, R., Dubach, J., & Bowden, D. (1990). Neuronames: Human/macaque neuroanatomical nomenclature. In *Proceedings, 14th annual symposium on Computer Applications in Medical Care* (pp. 1018–1019). American Medical Informatics Association.

Myers, L., & Brinkley, J. (1995). Visualization of brain surface features using registered partially segmented mri scans. In *SPIE medical imaging: Image display* (pp. 43–52). San Diego, CA: SPIE.

Ojemann, G., Ojemann, J., Lettich, E., & Berger, M. (1989). Cortical language localization in left, dominant hemisphere. *Journal of Neurosurgery, 71,* 316–326.

Ono, M., Kubik, C., & Abernathy, C. (1990). *Atlas of the cerebral sulci.* New York: Thieme Medical Publishers.

Prothero, J., & Prothero, J. (1989). A software package in c for interactive 3d reconstruction and display of anatomical objects from serial section data. In *NCGA proceedings,* pp. 187–192.

Rosse, C., Ben Said, M., Eno, K., & Brinkley, J. (1995). Enhancements of anatomical information in UMLS knowledge sources. In *19th symposium on computer applications in medical care* (pp. 873–877).

Sherk, H. (1992). Flattening the cerebral cortex by computer. *Journal of Neuroscience Methods, 41*(3), 255–267.

Talairach, J., & Tournoux, P. (1988). *Co-planar stereotaxic atlas of the human brain.* New York: Thieme Medical Publishers.

Vetter, R., Spell, C., & C.W. (1994). Mosaic and the world-wide web. *Computer, 27*(10), 49–57.

PLATE 16

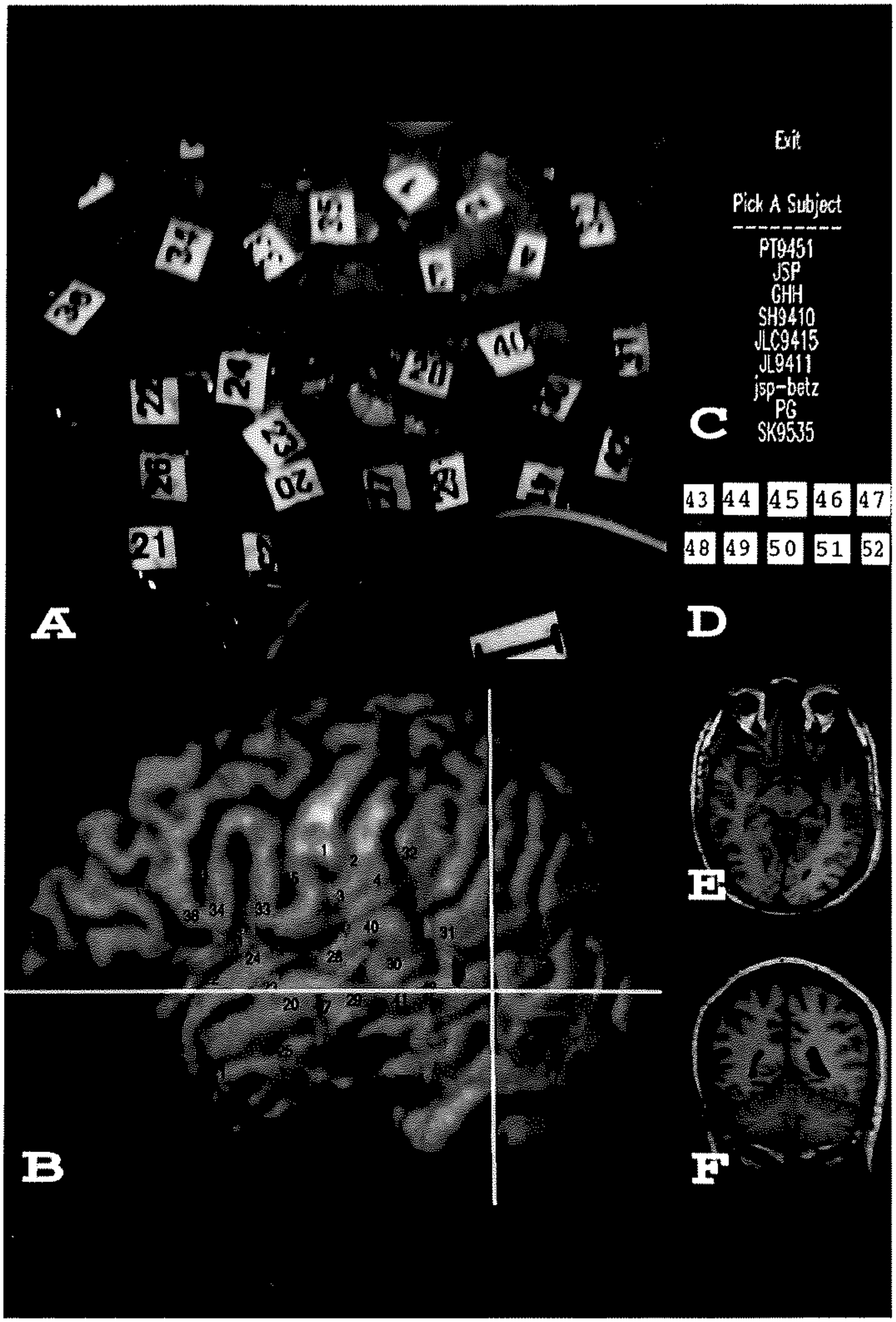

FIG. 9.1.