Neuroinformatics
© Copyright 2007 by Humana Press Inc.
All rights of any nature whatsoever are reserved.
1539-2791/07/3–10/$30.00 (Online) 1559-0089
DOI: 10.1385/NI:5:1:3

News Item

Unobtrusive Integration of Data Management With fMRI Analysis

Andrew V. Poliakov,,[1] Xenia Hertzenberg,[1] Eider B. Moore,[1] David P. Corina,[2] George A. Ojemann,[3] and James F. Brinkley[1]*

[1]Structural Informatics Group, Department of Biological Structure, University of Washington, Seattle, Washington; [2]Linguistics and Psychology Center for Mind and Brain, UC Davis, Davis, California; [3]Department of Neurological Surgery, University of Washington, Seattle, Washington

Abstract

This note describes a software utility, called X-batch which addresses two pressing issues typically faced by functional magnetic resonance imaging (fMRI) neuroimaging laboratories (1) analysis automation and (2) data management. The first issue is addressed by providing a simple batch mode processing tool for the popular SPM software package (http://www.fil.ion.ucl.ac.uk/spm/; Welcome Department of Imaging Neuroscience, London, UK). The second is addressed by transparently recording metadata describing all aspects of the batch job (e.g., subject demographics, analysis parameters, locations and names of created files, date and time of analysis, and so on). These metadata are recorded as instances of an extended version of the Protégé-based Experiment Lab Book ontology created by the Dartmouth fMRI Data Center. The resulting instantiated ontology provides a detailed record of all fMRI analyses performed, and as such can be part of larger systems for neuroimaging data management, sharing, and visualization. The X-batch system is in use in our own fMRI research, and is available for download at http://X-batch.sourceforge.net/.

Index Entries: fMRI; SPM; ontology; neuroimaging; data management; analysis automation.

(Neuroinformatics DOI: 10.1385/NI:5:1:3)

Introduction

Processing of functional neuroimaging data remains a computationally demanding and labor-intensive task. Neuroimaging laboratories dedicate substantial, computational, and human resources to perform sophisticated analysis required by such techniques as functional magnetic resonance imaging (fMRI), event-related potentials, diffusion tensor imaging, and many others. In the case of fMRI, analysis of an individual subject's dataset requires several steps of spatial/temporal preprocessing, followed by

*Author to whom all correspondence and reprint requests should be addressed.
E-mail: poliakov@u.washington.edu

statistical analysis of the resulting time series of image volumes (Frackowiak et al., 1997). Typical fMRI studies require analysis of dozens of subjects using one or more analysis protocols. These requirements, together with the need to eliminate human error during the analysis, have resulted in many batch scripts and utilities (for example, http://www.fil.ion.ucl.ac.uk/spm/ext/ provides a comprehensive list of such utilities for SPM). More advanced tools link together, analyses are done by different neuroimaging software packages, and provide advanced user interface, monitoring, and logging features, load balancing and supercomputing capabilities, and so on. Examples include the Laboratory of Neuro Imaging (LONI) pipeline (Rex et al., 2003), and functional imaging software widgets (FisWidgets; Fissell et al., 2003), RUMBA (Bly et al., 2004). The LONI pipeline provides a visually intuitive interface to data analysis whereas also allowing for diverse programs to interact seamlessly. FisWidgets is a graphical computing environment that provides a desktop style framework into which approx 100 subcomponents from a number of widely used fMRI analysis software packages (e.g., AFNI, AIR) are incorporated.

However, most of these utilities as well as many other analysis automation and batch processing tools, lack record-keeping or "self-documenting" features. Such features are becoming increasingly important as neuroimaging scientists face the problems of data management and sharing. Functional MRI analysis generates a large volume of data, together with associated metadata that describe

1. demographic information such as age, gender, and hand preference;
2. MRI scanner and coil parameters such as magnetic field strength, scanner software, and revision;
3. experimental protocol details;
4. analysis parameters;
5. statistical results; and
6. logistical information.

Maintaining, managing, and organizing such data, although very important for successful processing of large fMRI studies, can be burdensome. Many laboratories use basic tools to manage this information, including lab books or general purpose software tools such as spreadsheets. However, these tools do not scale up to the very large datasets generated by neuroimaging, nor do they record data in a manner that permits sharing.

These limitations were recognized by the developers of the Dartmouth fMRI Data Center (Van Horn et al., 2001) Experimental Lab Book (ELB) who developed and made available a software tool that facilitates record keeping and data management for fMRI studies. This tool organizes fMRI metadata using an ontological framework that not only acts as a container for the study data and metadata, but also preserves the relationships among data objects. However, the data for the ELB must currently be entered by hand, thereby decreasing the likelihood that a researcher will go to the extra trouble of using it.

In the current report, we describe a software tool that combines batch processing with data management. The tool, called X-batch is a plugin for the popular SPM software package. As the program is being used for batch processing of fMRI data, it automatically and transparently records the information in the ELB ontology, thereby unobtrusively capturing analysis metadata, whereas performing a time-saving task for the researcher. In the remainder of this report we describe the tool, its use in an example image-processing task, and the subsequent use of the generated ELB for data management and visualization.

Materials and Methods

Design

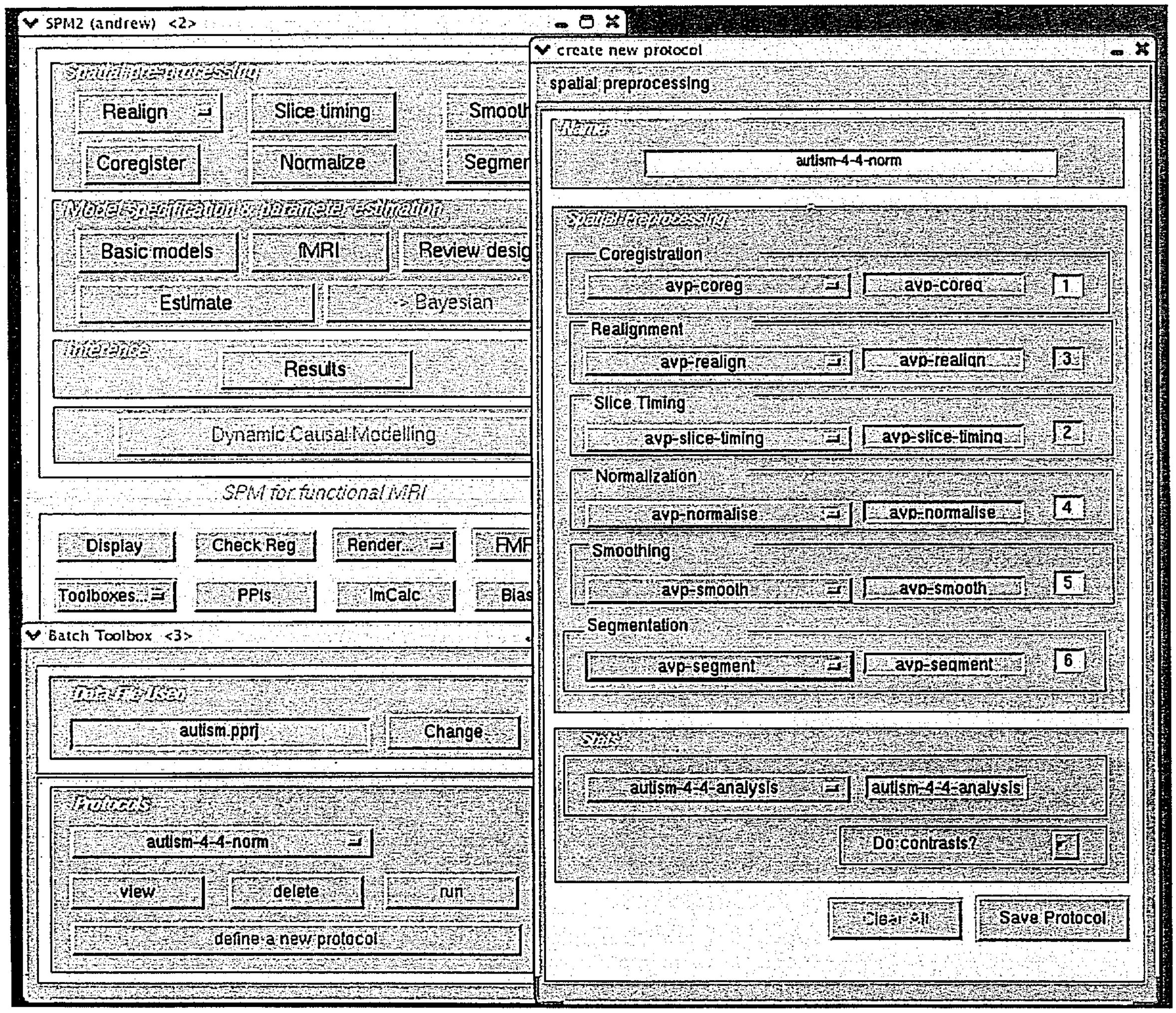
X-batch was developed as a toolbox for the popular SPM software package (SPM2; www.fil.ion.ucl.ac.uk/spm). Like SPM, X-batch is written in Matlab and is therefore platform-independent. The Dartmouth fMRI Data Center ELB, on the other hand, is based on the popular ontology building tool Protégé (Gennari et al., 2003), which represents knowledge as a semantic network of *classes* (e.g., Subject, Experiment, and so on), *relationships* (Subject *participates-in* Experiment), and *data* as instances of the classes (Subject *has-instance* John Doe). Protégé is written in Java, and exports a Java application programming interface (API) that provides access to the classes, relationships, and instances of the ontology. X-batch accesses this API using the Java interface provided by Matlab, thus, eliminating the need for the SPM user to know that Protégé is running in the background. Like X-batch itself, both SPM and Protégé are free open source software packages.

The X-batch toolbox presents the user with an easy to follow graphical user interface (GUI) that conforms to the familiar SPM2 interface (Fig. 1). The GUI includes custom components that are used to create/manage/delete instances of such ELB classes as Subject and Protocol. It is used, for example, to define preprocessing and statistical analysis parameters or to show the progress of data processing during the batch job using SPM2 graphics. The latter can be disabled, making batch processing faster, particularly, when it is being used remotely (using remote X windows, VNC, or other tools).

The ELB ontology generated by X-batch is saved in a Protégé-specific file format that can be added-to in multiple runs of X-batch, can be viewed by the Protégé ontology editor, and can be converted to other formats using the Protégé API. The original ELB ontology includes entries for subject demographic parameters, scanner and coil, experimental design, and many others. We extended the ontology by adding a statistical processing node (*see* Fig. 2) to capture all parameters of fMRI analysis performed by SPM, and the results of this analysis saved in various files (Fileset node). The statistical processing node captures the sequence of SPM processing steps (Protocol node) as well as parameters and options used at every step, including spatial preprocessing (coregistration, normalization, realignment, slice timing correction, smoothing, segmentation) and statistical analysis parameters (analysis design, image volumes, and regressors used). This ontology class was designed specifically to capture fMRI analysis as implemented by SPM2. Functional MRI analysis can also be performed using other software packages (AFNI, FSL, and so on) as well as other versions of the SPM software package, and therefore a more comprehensive fMRI ontology would conceivably require implementing classes for each version of each software package, possibly with a layer of abstract classes to capture common features of all such classes.

X-Batch Use

X-batch was designed to automate spatial and temporal preprocessing and statistical analysis of fMRI data for individual subjects, i.e., the first level of analysis. This kind of analysis is typically the most labor-intensive part of fMRI analysis. Second level analysis is not supported at this time, but multiple sessions per subject are supported. When invoked from the SPM2 toolbox menu, X-batch presents the user with the Protocol GUI (Fig. 1). Using this interface the researcher defines and stores batch processing sequences–analysis protocols—without using any programming or scripting language, and later runs these protocols as batch processes. Previously defined—protocols are saved as instances in the ELB ontology, and can therefore be viewed and edited in either Protégé (Fig. 2) or X-batch. X-batch supports



Data File Used: autism.pprj

Protocols: autism-4-4-norm

Name: autism-4-4-norm

Coregistration	avp-coreg	avp-coreg	1
Realignment	avp-realign	avp-realign	3
Slice Timing	avp-slice-timing	avp-slice-timing	2
Normalization	avp-normalise	avp-normalise	4
Smoothing	avp-smooth	avp-smooth	5
Segmentation	avp-segment	avp-segment	6

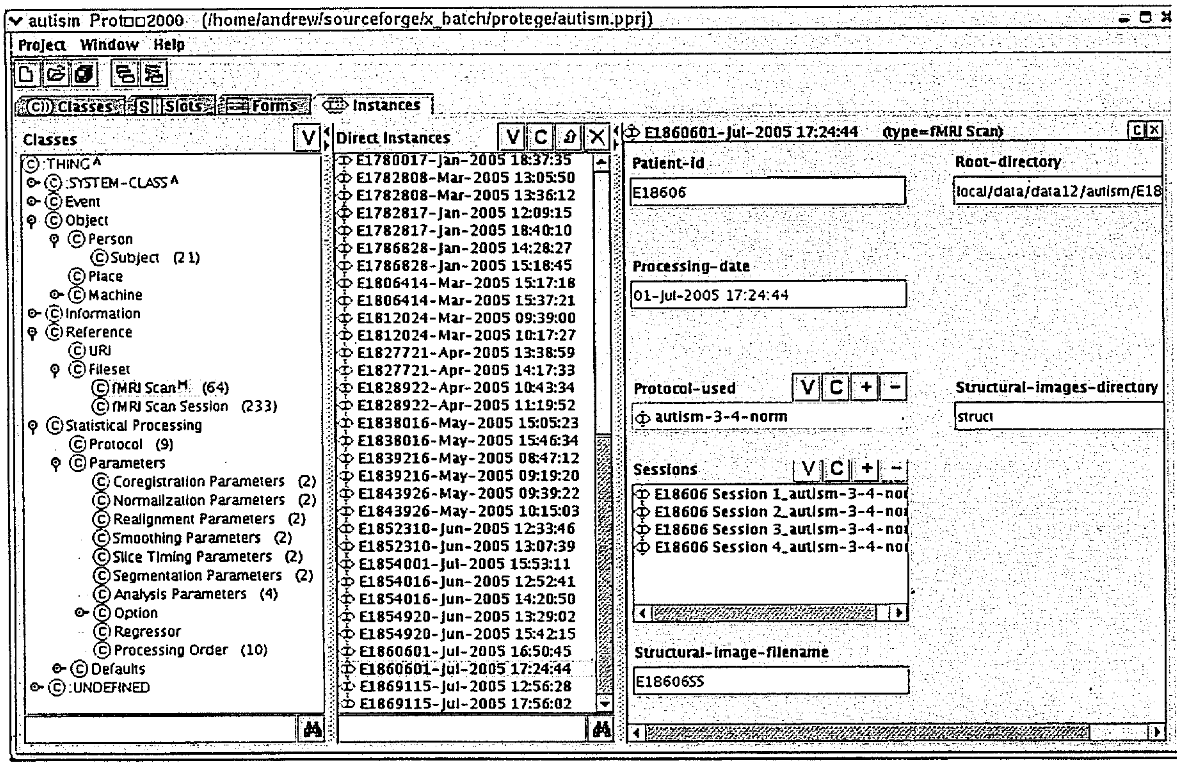
Stats: autism-4-4-analysis | autism-4-4-analysis

Do contrasts? ☑

Fig. 1. Screenshot of X-batch GUI for creating a protocol. The X-batch toolkit can be invoked using the toolboxes pulldown menu of SPM2 (middle left panel). The X-batch top-level GUI allows a user to select the ELB file and analysis protocols. The right panel shows the creation of a protocol called autism-4-4-norm. Protocols consist of processing stages, each stage defined by its parameter set, and ordered by the user-entered numbers at the right. The researcher can reuse previously created parameter sets, define new ones, or use a combination. Parameter sets are defined using the SPM2 user interface, which researchers are already well familiar with. Once a protocol is created, it can be used to process any number of datasets from beginning to end. When the user selects a protocol and presses the Run button, a Subject GUI is invoked that allows the user to manage subjects and run the analysis protocol for any number of subjects.

multiple instances of ELB ontologies that can be used, for example, to manage data for different experiments.

When the protocol is run X-batch organizes the processed images into an appropriate directory hierarchy, with the analysis results for each protocol saved in a separate directory, so that results are not overwritten by running a different protocol (note that SPM2 itself will overwrite the results in the working



autisin Prot□□2000 (/home/andrew/sourceforge/x_batch/protege/autism.pprj)

Classes

```
:THING
:SYSTEM-CLASS
Event
Object
  Person
    Subject (21)
  Place
  Machine
Information
Reference
  URI
  Fileset
    fMRI Scan (64)
    fMRI Scan Session (233)
Statistical Processing
  Protocol (9)
  Parameters
    Coregistration Parameters (2)
    Normalization Parameters (2)
    Realignment Parameters (2)
    Smoothing Parameters (2)
    Slice Timing Parameters (2)
    Segmentation Parameters (2)
    Analysis Parameters (4)
    Option
    Regressor
    Processing Order (10)
  Defaults
:UNDEFINED
```

Direct Instances

```
E1780017-Jan-2005 18:37:35
E1782808-Mar-2005 13:05:50
E1782808-Mar-2005 13:36:12
E1782817-Jan-2005 12:09:15
E1782817-Jan-2005 18:40:10
E1786828-Jan-2005 14:28:27
E1786828-Jan-2005 15:18:45
E1806414-Mar-2005 15:17:18
E1806414-Mar-2005 15:37:21
E1812024-Mar-2005 09:39:00
E1812024-Mar-2005 10:17:27
E1827721-Apr-2005 13:38:59
E1827721-Apr-2005 14:17:33
E1828922-Apr-2005 10:43:34
E1828922-Apr-2005 11:19:52
E1838016-May-2005 15:05:23
E1838016-May-2005 15:46:34
E1839216-May-2005 08:47:12
E1839216-May-2005 09:19:20
E1843926-May-2005 09:39:22
E1843926-May-2005 10:15:03
E1852310-Jun-2005 12:33:46
E1852310-Jun-2005 13:07:39
E1854001-Jul-2005 15:53:11
E1854016-Jun-2005 12:52:41
E1854016-Jun-2005 14:20:50
E1854920-Jun-2005 13:29:02
E1854920-Jun-2005 15:42:15
E1860601-Jul-2005 16:50:45
E1860601-Jul-2005 17:24:44
E1869115-Jul-2005 12:56:28
E1869115-Jul-2005 17:56:02
```

E1860601-Jul-2005 17:24:44 (type=fMRI Scan)

Patient-id: E18606

Root-directory: local/data/data12/autism/E18

Processing-date: 01-Jul-2005 17:24:44

Protocol-used: autism-3-4-norm

Structural-images-directory: struct

Sessions:
E18606 Session 1_autism-3-4-nor
E18606 Session 2_autism-3-4-nor
E18606 Session 3_autism-3-4-nor
E18606 Session 4_autism-3-4-nor

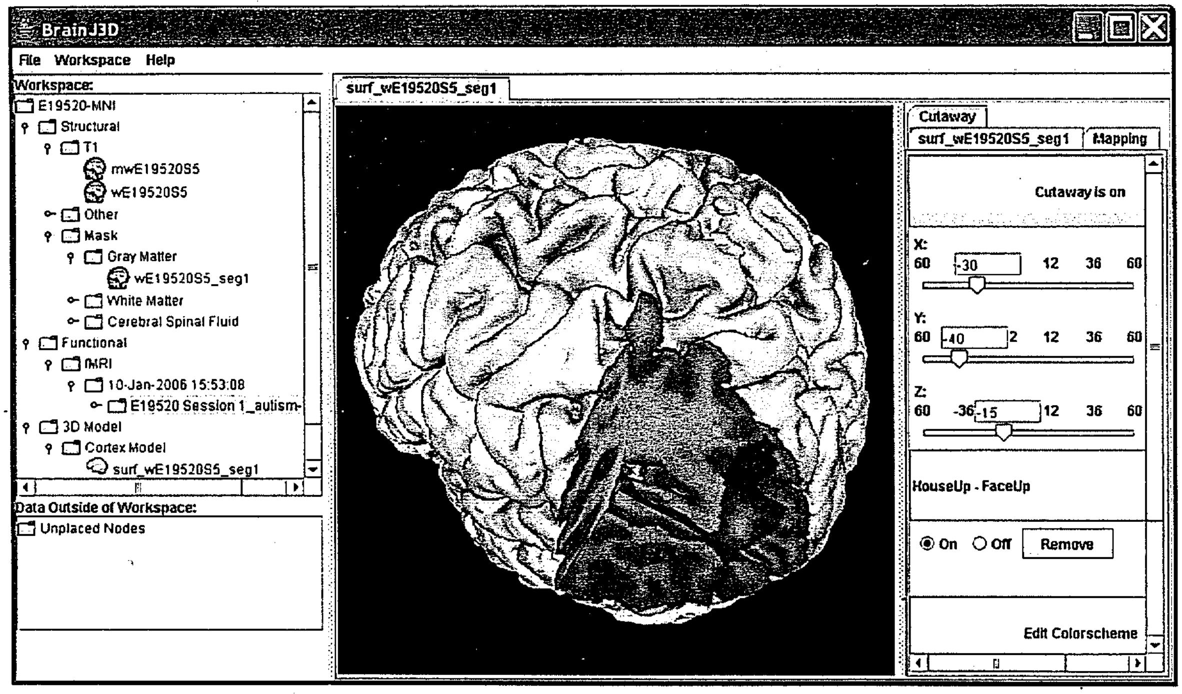
Structural-image-filename: E18606SS

Fig. 2. ELB ontology created while processing fMRI data with X-batch. Once created in X-batch the fMRI analysis ontology can be viewed, studied, and modified using the Protégé ontology editor. The ontology generated by X-batch is an extension of the fMRI analysis ontology of the ELB software developed at fMRI Data Center. The ontology was designed to give a fairly comprehensive description of a typical fMRI experiment, and has entries for patient demographics data, scanner and coil type, experiment description, and many others. For X-batch the Statistical Processing node (highlighted) was added to capture the spatial preprocessing and statistical analysis parameters entered in the X-batch GUI (Fig. 1). While X-batch performs the analysis sequence, pertinent information is captured in the ontology. This information includes subject ID, fMRI scan information, including names and location of all functional and structural MRI scans, analysis protocol used, all analysis parameters, processing date and time, and information about the analysis results, including names and location of various files created during the analysis.

directory, if present). This makes it easy to use multiple protocols to process the same dataset. For example, we routinely use at least two protocols—one for processing the data in patient-specific space, and another for processing the data in standard space that includes a normalization step. The locations and filenames of the generated files are saved in the ontology and can be utilized for data management and further processing.

Utilizing the Ontology

Once the ontology has been created Protégé tools can be utilized to study and analyze the ontology, and to format it in several ways, including HTML for viewing on the web. In addition, we have used the Protégé API to export the ontology in custom formats that permit its integration into larger information systems. For example, we have used the Protégé API to generate an XML file defining a patient-centric view



Cutaway is on

X: -30

Y: -40

Z: -15

● On ○ Off

Fig. 3. Using ELB ontology to automatically create and populate workspaces for data visualization tool. One example of the use of the ELB fMRI ontology is as input to a visualization tool. In this case, the figure shows a snapshot of our 3D-visualization tool Mindseer with the dataset of one subject loaded. All pertinent data are presented in a workspace (on the left) that is derived from the ELB ontology shown in Fig. 2. Mindseer provides appropriate methods for viewing and handling the data of each category in the workspace, including structural and functional image volumes, 3D models, 2D images, and 3D points and labels (last two not shown here). For subjects that were processed using X-batch, we automatically create two workspaces–one for the data in standard MNI space and another for the data in subject-specific space–by postprocessing the information stored in the ELB ontology.

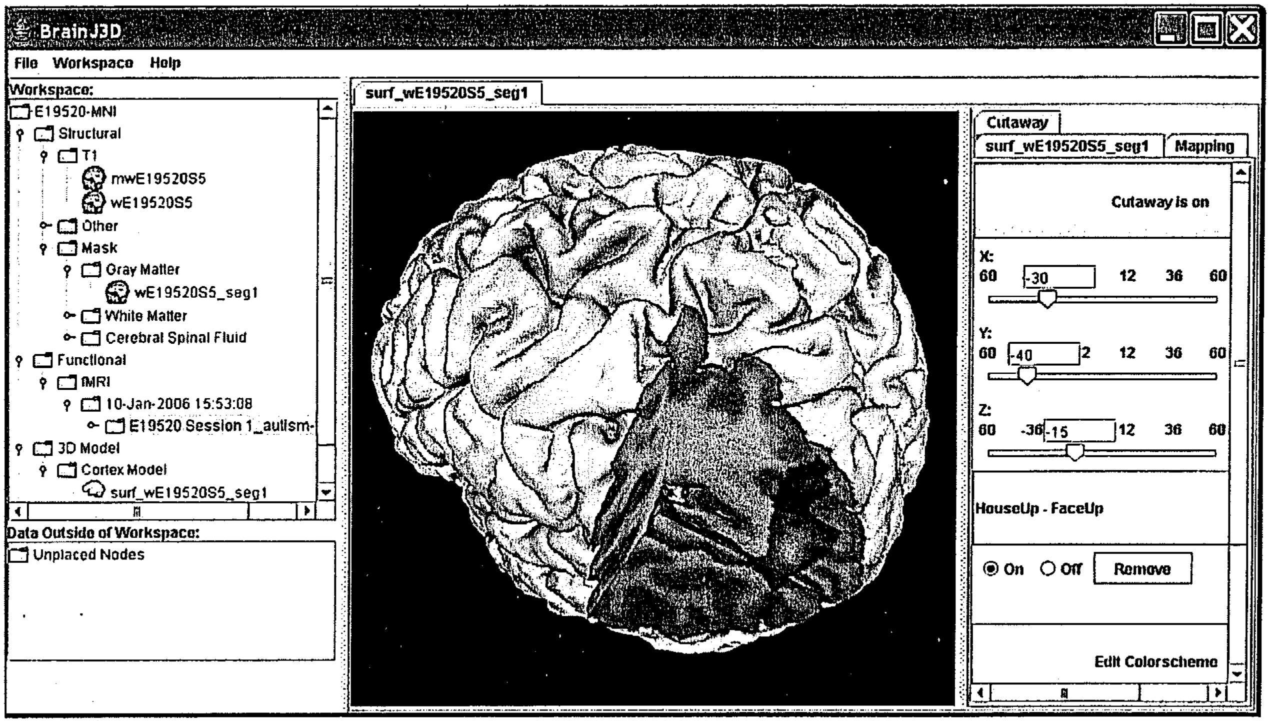
of the ontology. This file can be furthermore processed to generate the input required by our Java three-dimension (3D)-based visualization software called MindSeer (Moore et al., 2004; Fig. 3). It has also been used to generate an fMRI data source in a distributed data management system, we are developing that combines and visualizes multimodality data from multiple sources (Bales et al., 2005).

System Status

X-batch has been in use in our lab for more than 3 yr for our Human Brain Project studies comparing fMRI language activation with cortical stimulation (Corina et al., 2005). In addition, it has been used to process patient datasets in studies of autism and dyslexia at the University of Washington. It has been available on SourceForge http://X-batch.sourceforge.net since August 2004, and based on SourceForge statistics has been downloaded more than 180 times. We would like to express our gratitude to all users who provided feedback and helped us in identifying and fixing the problems as well as improving X-batch software.

Discussion

The primary contribution of X-batch is that it combines solutions to two problems in a single



X: -30

Y: -40

Z: -15

ig. 3. Using ELB ontology to automatically create and populate workspaces for data visualization tool. One xample of the use of the ELB fMRI ontology is as input to a visualization tool. In this case, the figure shows snapshot of our 3D-visualization tool Mindseer with the dataset of one subject loaded. All pertinent data re presented in a workspace (on the left) that is derived from the ELB ontology shown in Fig. 2. Mindseer application: automation of routine image-processing tasks and data management. There are many solutions to the automation problem, ranging from batch scripts to advanced analysis tools such as LONI pipeline, FisWidgets. However, to our knowledge, few, if any of these solutions save the results of these analyses in a database. At the same time, there are many solutions to the neuroimaging-database problem, ranging from local lab-management systems like our own (Jakobovits et al., 1996) to central repositories such as the International Consortium for Brain Mapping (ICBM) database at UCLA (Toga, 2002a,b) the surface management system database at Washington University (Dickson et al., 2001), the Biomedical Informatics Research Network project (Grethe et al., 2005), and the fMRI Data Center at Dartmouth (Van Horn et al., 2001). However, most or all of these systems require separate entry and submission of image data and metadata, thereby adding an extra burden to the individual researcher that often prevents him or her from submitting data to the database.

The Dartmouth ELB is an attempt to reduce some of this burden by allowing the researcher to manage local lab data as a preliminary step to automated upload to the central site. It thus has the potential to satisfy the need of researchers to manage their local data whereas making it easy to upload data to the central site. However, the ELB still requires manual entry of the data in the local lab book, a burden that in our experience is more than many researchers are willing to put up with.

By transparently embedding the ELB within a batch automation tool we reduce or even eliminate the burden of entering data, whereas providing a batch automation facility that is highly desired by the researchers as it saves time and reduces errors. Our own experience suggests that this kind of transparent data management is much more likely to be used than separate data-management tools. Once the local data are recorded in such a system then they become much easier for a researcher who so desires to either submit the data to a central repository or to make the data available as a source in a federated system. Thus, as transparent data management is embedded in tools like X-batch or many of the existing image automation tools there should be a high increase in the amount of image data that are available for sharing (Brinkley and Rosse, 2002; Koslow, 2002).

Information Sharing Statement

X-batch is an open source software distributed under General Public License (GPL) and is available on SourceForge at http://X-batch.sourceforge.net.

Acknowledgments

This work was funded by Human Brain Project grant DC02310. AVP was supported in part by Advance Autism Research and Treatment Centers Program National Institutes of Mental Health (NIMH; grant U54MH066399).

References

Bales, N., Brinkley, J. F., et al. (2005) A Framework for XML-based Integration of Data, Visualization and Analysis in a Biomedical Domain. Proceedings, XSym 2005, pp. 207–221, Trondheim, Norway.

Bly, B. M., Rebbechi, D., Hanson, S. J., Grasso, G. (2004) The RUMBA software: tools for neuroimaging data analysis. Neuroinformatics 2(1), 71–100.

Brinkley, J. F. and Rosse, C. (2002) Imaging Informatics and the Human Brain Project: the Role of Structure, Review Yearbook of Medical Informatics, 111–128. Pub Med ID: 12425235.

Corina D. P., Gibson E. K., Martin R., Poliakov A. V., Brinkley J. F., Ojemann G. A. (2005) Dissociation of action and object naming: evidence from cortical stimulation mapping. Hum. Brain Mapp. 24(1), 1–10.

Dickson, J., Drury, H., and Van Essen, D.C. (2001) The surface management system (SuMS) database: A surface-based database to aid cortical surface reconstruction, visualization and analysis. Phil. Trans. Royal Soc. Ser. B. 356, 1277–1292.

Fissell, K., Tseytlin, E., Cunningham, D., Carter, C. S., Schneider, W., and Cohen J. D. (2003) Fiswidgets: A graphical computing environment for neuroimaging analysis. Neuroinformatics 1, 111–125.

Frackowiak, R. S. J., Friston, K. J., Frith, C. D., Dolan, R. J., and Mazziotta, J.C. (eds.) (1997) Human Brain Function. Academic Press, USA.

Gennari, J. H., Musen, M. A., Fergerson, R. W., et al. (2003) The evolution of Protégé: an environment for knowledge-based systems development. Int. J. Hum. Comput. Stud. 58(1), 89–123.

Grethe, J. S., Baru, C., Gupta, A., et al. (2005) Biomedical informatics research network: building a national collaboratory to hasten the derivation of new understanding and treatment of disease. Stud. Health Technol. Inf. 112, 100–109.

Hinshaw, K. P., Poliakov, A. V., Martin, R. F., Moore, E. B., Shapiro, L. G., and Brinkley, J. F. (2002) Shape-based cortical surface segmentation for visualization brain mapping. NeuroImage 16, 295–316.

Jakobovits, R. M., Modayur, B, Brinkley, J. F. (1996) A Web-based repository manager for brain mapping data. Proc. AMIA Annu. Fall Symp. pp. 309–313.

Koslow, S. H. (2002) Opinion: Sharing primary data: a threat or asset to discovery? Nat. Rev. Neurosci. 4, 311–313.

Moore, E. B., Poliakov, A. V., and Brinkley, J. F. (2004) Brain Visualization in Java3D. Proceedings, MedInfo 2004 San Francisco, CA, p. 1761.

Poliakov, A. V., Hinshaw, K. P., Rosse, C., and Brinkley, J. F. (1999) Integration and Visualization of Multimodality Brain Data for Language Mapping, Proceedings AMIA Annu. Fall Symp pp. 349–353.

Rex, D. E., Ma, J. Q., and Toga, A. W. (2003) The LONI Pipeline Processing Environment. NeuroImage 19, 1033–1048.

Toga A. W. (2002a) Imaging Databases and Neuroscience. Neuroscientist 8, 423–436.

Toga, A. W. (2002b) Neuroimage databases: the good, the bad and the ugly. Nat. Rev. Neurosci. 4, 302–309.

Van Essen, D. C. and Drury, H. A. (1997) Structural and functional analyses of human cerebral cortex using a surface-based atlas. J. Neurosci. 17, 7079–7102.

Van Horn, J. D., Grethe, J. S., Kostelec P., et al. (2001) The Functional Magnetic Resonance Imaging Data Center (fMRIDC): the challenges and rewards of large-scale databasing of neuroimaging studies. Philos. Trans. R Soc. Lond. B Biol. Sci. 356(1412), 1323–1339.

Wong, B. A. and Brinkley, J. F. (1998) Dynamic 3-D scene navigation in web-based anatomy atlases, Proceedings AMIA Annu. Fall Symp. pp. 1100–1104.