Non-curated distributed databases for experimental data and models in neuroscience

R.C. Cannon[†‡§], F.W. Howell[‡], N.H. Goddard[‡] and E. De Schutter[†]

† Theoretical Neurobiology, Born-Bunge Foundation, University of Antwerp, B2610 Antwerp, Belgium

‡ Institute for Adaptive and Neural Computation, Division of Informatics, University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL, Scotland

Abstract. Neuroscience is generating vast amounts of highly diverse data which is of potential interest to researchers beyond the labs in which it is collected. In particular, quantitative neuroanatomical data is relevant to a wide variety of areas including studies of development, aging, pathology and in biophysically oriented computational modeling. Moreover, the relatively discrete and well-defined nature of the data make it an ideal application for developing systems designed to facilitate data archiving, sharing and reuse. At present, the only widely used forms of dissemination are figures and tables in published papers which suffer from inaccessibility and the loss of machine-readability. They may also present only an averaged or otherwise selected subset of the available data. Numerous database projects are in progress to address these shortcomings. They employ a variety of architectures and philosophies, each with its own merits and disadvantages. One axis on which they may be distinguished is the degree of top-down control, or curation, involved in data entry. Here we consider one extreme of this scale in which there is no curation, minimal standardization, and a wide degree of freedom in the form of records used to document data. Such a scheme has advantages in the ease of database creation and in the equitable assignment of perceived intellectual property by keeping the control of data in the hands of the experts who collected it. It does, however, require a more sophisticated infrastructure than conventional databases since the software must be capable of organizing diverse and differently documented data sets in an effective way. Several components of a software system to provide this infrastructure are now in place. Examples are presented showing how these tools can be used to archive and publish neuronal morphology data, and how they can give an integrated view of data stored at many different sites.

Submitted to: *Network: Comput. Neural Syst.*

§ To whom correspondence should be addressed (robert.cannon@ed.ac.uk)

1. Introduction

There are two broad classes of database activity currently underway in neuroscience: the development of large centralized repositories; and the creation of smaller tightly controlled resources by researchers acting as expert curators. Among the former is the FMRI Data Center[1] which follows an approach that has been very successful in genomics and bioinformatics with projects such as GenBank (*www.ncbi.nlm.nih.gov/Genbank/*) and the Protein Data Bank (*www.rcsb.org/pdb/*). The data are stored and managed at the repository, which typically also provides a range of searching and data-processing facilities. Data entry is by direct submission from the scientists who collect the data. In contrast, heavily curated resources such as Senselab[2] or CoCoMac[3] do not require the participation of the original researchers, and are instead populated from published work. Each approach fits a certain need, but the vast majority of data collected in neuroscience still does not find its way into any publicly accessible location. Here we consider some of the hurdles to more extensive data sharing, and how they may be overcome by the decentralization of database efforts without compromising the accessibility of the data.

In considering the effort involved in creating a database, the dominant concerns for the academic research community are sociological issues of who does the work and why[4]. This is different from a commercial environment where the rewards are primarily financial and can be used to direct intensive labor investment in tightly curated database systems (e.g. Inpharmatica, *www.inpharmatica.com*). Successful curated neuroscience databases in the public domain are often the work of a small group of dedicated individuals creating a resource which may be perceived as their intellectual property, not that of the data providers. This immediately explains why the groups remain small – because other researchers have very little interest in contributing to a project where most of the credit will go elsewhere, there being no accepted method for the redistribution of intellectual property among database contributors other than joint publications. This also gives an indication of how a successful large scale database system might be structured – as a conglomerate of many small units, each controlled by the providers of the data it contains.

Distributed, or "grass roots" databases would look quite different from existing databases because they require a modularization of the database functionality to match the facilities and competences of the participants. For example, individual sites need not run a database server or a search engine: these could be provided in a few sites run by informaticians. Likewise, analysis and visualization tools need not be part of any one site. It is sufficient that they are cataloged and available at at least one place, and that information about what forms of data are compatible with which tool is also available (though perhaps on a different site controlled by the software users). The primary goal in designing a successful distributed system for the academic environment is that it should be composed of distinct inter-operable units, where the perception of intellectual property of the whole is distributed to the units commensurately with their

real intellectual investment. After intellectual control, probably the most important issue for distributed databases is the minimization of redundant effort. This amounts to making software do as much of the work as possible, including tasks such as web site creation, cataloging and cross referencing. Software may also be able use knowledge bases to resolve differences in the way data is documented and to prompt users to supply the required information when it is not yet available.

In developing distributed databases for the storage and dissemination of biological data, the needs of quantitative neuroanatomists present an excellent target application. The data are sufficiently diverse to require new methods beyond those of traditional databases, but they are also typically well defined and documented because each study often represents weeks or even months effort on a single tissue preparation. For this reason, archiving neuroanatomical data is a more tractable problem than, for example, archiving data from slice electrophysiology where large volumes of data can be generated from several different experiments in a single day. The next section reviews the requirements of databases dedicated to neuronal morphology, and reviews the history of one such system that has now been available for four years. Experience with this database has motivated exploration of the non-curated approach and the development of a particular implementation known as "Axiope" (*www.axiope.org*). The architecture of this system is discussed in section three including screenshots and examples of its use for archiving neuronal morphology. Section four focuses on how the system can be used in neuroanatomy including issues of standardization and cooperation among data providers. The possible directions for future development are considered in section five.

2. Review of the Duke-Southampton archive of hippocampal neuronal morphology

The extraction of accurate quantitative data on the morphology of neurons from tissue preparations in which cells have been stained is difficult and time consuming process[5]. Although usually collected in the course of a particular research study with very specific aims, such data are often of interest to other researchers and in quite different contexts. In order to promote the reuse of morphology data collected by the labs of Turner[6] and Buzsaki[7], a simple web-based archive was established in 1998[8]‖ housing digitizations of 126 hippocampal neurons. These included CA1 and CA3 pyramidal cells, granule cells from the dentate gyrus and interneurons. They came from rats of a variety of ages, some of which had undergone kainic acid lesions leading to denervation in CA1[9]. Since being made freely available on the web, these data have been used in a number of independent studies including modeling of their electrophysiological properties[10, 11, 12] and as reference materials for tuning algorithms for computer generation of cell geometries[13].

Capturing the morphology of a neuron involves storing the 3-D positions and diameters of all parts of the axonal and dendritic arborizations. Although work is under way on fully automated digitization systems[14], the most reliable approach is still

‖ presently located at www.compneuro.org/CDROM/data

computer assisted manual digitization. In the most widely used system, Neurolucida (Microbrightfield Inc., Manchester, VT), the tissues is mounted on a motorized stage and viewed through a video microscope. The cell structure is traced by picking out successive points on each neural process with a variable size cursor. The cursor size is adjusted to match the current diameter, and the 3-D coordinates and diameter of a point are captured by the software each time the mouse is clicked. The resulting data is a set of points from which the full arborization can be reconstructed by connecting each point to its neighbors on the axon or dendrite. The Neurolucida system has its own format for storing these data along with other information needed in the digitization process and for visualization. Because this format is relatively complicated, an alternative was provided in the Duke-Southampton web site, now known as the ".SWC" format[8] in which each point was expressed on a separate line by seven quantities: an integer code for the point; a code for its type (axon, apical dendrite etc); three spatial coordinates; the diameter; and the code of its parent (the next point toward the soma).

As well as the cell structures themselves, the web site contained a cross-platform visualization and conversion tool which could be used to validate and correct structures (verify that they are fully connected or that there are no loops, for example), and to convert them to the formats used by the Neuron[15] and Genesis[16] modeling packages. The site also contained a simple facility for other labs to upload data and have it cataloged in the same way as the original hippocampal data. This was unnecessary, because there has been no interest in contributing further data to the archive, although a number of groups did express interest in the software used to construct the archive so that they could build their own.

Unlike the majority of recent database projects, which concentrate on extracting information from the literature, the Duke-Southampton archive contained unpublished data that was not available elsewhere. The experimental work leading to the data collection had been published, and properties of some of the cells, but not the structures themselves. Also, unlike most neuroscience database projects, there was very little intellectual input by the constructors of the archive: most of the effort had been experimental, with a thin layer of software to put it together. The key distinction, is that if substantial effort is involved in extracting and organizing information from the literature (as for example with the NeuronDB[2] or CoCoMac[3] databases), then it is natural that the database maintainers have intellectual control over the results. If, however, the archiving process is purely technical, then it is equally natural that the data providers should wish to retain intellectual control. For the Duke-Southampton archive this was only partially the case, since, the archive maintainer was frequently credited with the usefulness of the site, not the contributors. This explains the lack of interest in contributing more data to the archive. While the site did provide visibility for the data, it did not help researchers build up a scientific reputation based on being personally associated with the data they generate. Finding a framework which can simultaneously meet these two requirements is the principal motivation for the redesign of the data archiving system presented here.

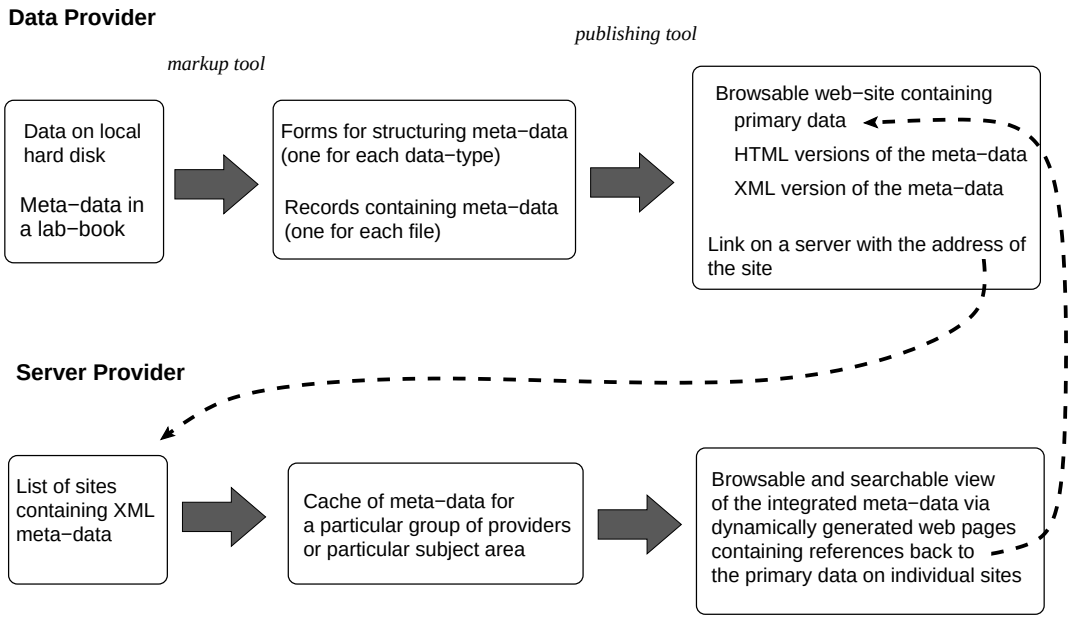

Figure 1. Distributed database architecture. The creation and management of the database system is divided into two distinct tasks with minimal interaction between them. Data providers use a markup and publishing tool to construct local machine readable descriptions of their data and to upload these to their own web space in a variety of formats. The publishing tool also notifies one or more servers of the location of the data. The servers then fetch the descriptive information (not the data itself) from several different sites and provide a searchable integrated view of what is available.

3. Database Architecture

The components of the present data archiving system are illustrated in figure 1. The most distinctive feature is that the two main tasks — the provision of data and the application of IT expertise — are widely separated, to the extent of even being located at different sites. A data provider uses a single program to performs all the steps necessary to construct a web-accessible archive from data stored in a local machine. Such archives or repositories are here termed "data-sites". They are hosted on the provider's own web-space, and require nothing more than standard software that is included in most modern operating systems. All the database functionality such as creating catalogs, providing search facilities, and compiling statistics is handled by external servers. The requirements for IT expertise are therefore concentrated in the maintenance and operation of these servers, removing any need for IT experts in individual labs. At present a server is available at the host site (*www.axiope.org*) for all types of data, but it is envisaged that separate consortia or large research groups may wish to host their own servers. The two main aspects of the system – data-site creation, and cataloging – are illustrated from the user's perspective in the next two subsections.

3.1. Creating a data-site

The main window of the data-entry and site creation tool (called "Builder" in this version) is shown in figure 2. The tasks to be accomplished are: (1) input of descriptive information (meta-data) for each file to be published; (2) creation of a local web site; (3) transfer of this site to a publicly accessible web server; and (4) notification of a catalog server that the new site exists. Steps 2, 3, and 4 are all handled by the software with minimal involvement of the user. There is a simple form to fill in containing the name of the web server and a few other configuration options, but once this information is within the system, the site can be rebuilt with no further action by the user. The main task of data providers is therefore the one task that the machine cannot perform – the creation of structured meta-data explaining what each file contains.

The panel on the left in figure 2 shows a browsable directory listing of local files. The squares are color-coded to indicate whether there is any meta-data included yet for the file and, if present, how complete the meta-data is (how many of the entries have been filled in). Using the terminology from library catalogs, files of meta-data are called "records" here. In this analogy, data files on the local computer correspond to the books in a library, and files of meta-data correspond to the records in the library catalog that contain the title, author, date etc for a book. Selecting an item from the directory browser allows the user to create a new record or to edit one that already exists. In the figure, a record for one of the morphology files from the Duke-Southampton archive is being edited.

To create or edit a record, it is necessary for the system to have a specification of the fields that belong in the record. Such a specification is essentially a list of the field names and is here termed a "form". By default, records are assumed to contain textual information that should be typed by the user, but the form may include further information to facilitate data entry. For example, forms may specify a set of possible values to be presented in a menu, or that a particular field should contain a file name so the user interface can provide a graphical file selector. There is also a reference field for making links between records so that if the same information is needed in many places, it may be entered only once and then referred to from other records.

If an adequate form already exists, then filling out the record amounts to selecting items from menus and typing into text boxes. If no suitable form is available, then the same tool that is used for filling out records, also allows the underlying form to be changed. Starting from the standard base form (which simply contains fields for the file name, date, summary and keywords) fields can be added, removed or modified with the buttons above the record. These options are only shown when the form is being modified. In this way forms can be rapidly built up and adjusted as required. In particular, the process of meta-data entry frequently brings to light other information that may be pertinent and could usefully be included in a record. Although this could always be included as notes at the end of the form, such notes are less useful than having the information in the body of the form because they generally cannot be processed

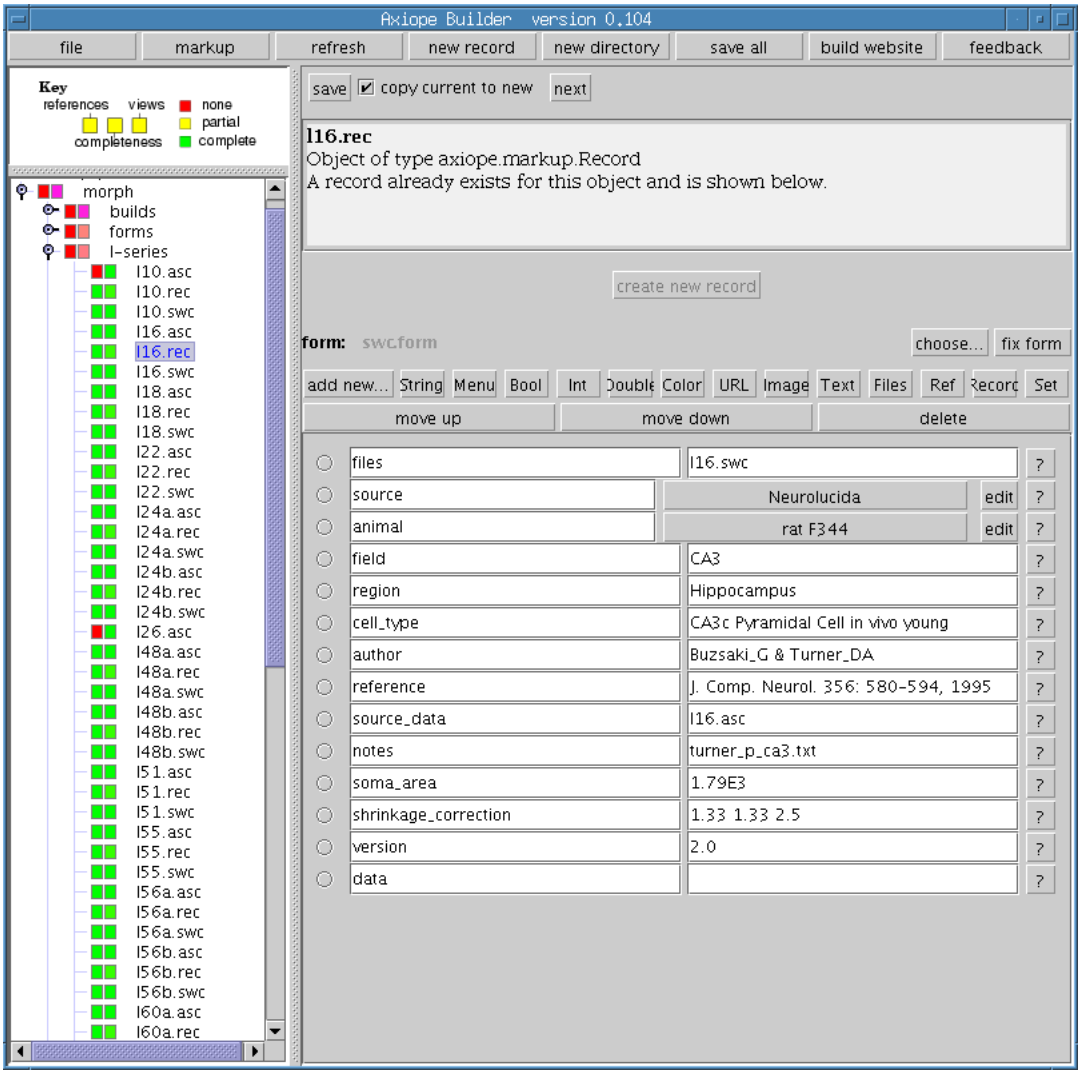

Figure 2. The data entry and publication tool. On the left is a conventional directory browser. Selecting an element from the list gives the option of creating a new file to describe that element. One such file is shown in the right-hand panel. In this example, the tool is in "form editing mode" which allows not only data to be entered about the current file, but also the underlying form to be changed using the long row of buttons and menus.

by computer. By making it very easy to modify the form to accommodate the new information, the present system ensures that all meta-data is in the most accessible format. Moreover, the modifications made to a form while editing one record carry over to any other record that uses the same form. When these records are revisited, there will be empty slots waiting for the new meta-data. The same on-the-fly form modification applies to all the possible fields, so, for example, new options can be added to menus at any stage, and will thereafter be available in all records that use the form.

At any point in the markup-entry process, a web-site can be built from the existing information. The site is structured rather like the markup-entry window, with a directory browser and file listings on the left and the objects themselves (records pointing

to data files) in the main part of the window. The site building process also saves all the meta-data as XML (eXtensible Markup Language, *www.w3c.org/XML*) files in a form similar to that adopted for NeuroML[17]. These contain a more structured version of the same information as in the HTML pages and are used by the cataloging and searching software.

3.2. Browsing and searching

Once a new data-site has been created and the software has notified a server of the site's address, then it can be accessed through the database engine (called "Surfer" in the current version). A screenshot of the morphology archive as viewed this way is shown in figure 3. The "Surfer" operates only on the XML versions of the records, and does not fetch any of the primary data from the data-site. Instead, it uses the meta-data to create web pages dynamically¶ The pages contain links to data on the primary data sites, so, for example, if there are images to be included in the page, these are transferred directly from the data-site to the users browser without any involvement of the server. Effectively, the server compiles pages containing references to the information and leaves it up to the browser to fetch and display it.

Although there are the conventional options for searching by names, keywords etc, the server also allows searching by browsing based on sorting the data on user specified fields. For example, in searching the morphology archive for CA1 cells, the server can be instructed to present all the data sorted by region. This will provide a list of links for CA1, CA3, dentate gyrus etc. Clicking the CA1 link will give a list of all the CA1 cells, which could then be sub-sorted by cell type and so on. This overcomes the often frustrating problem of being presented with a searchable database without knowing what terms to search for. If it returns no matches to "Dentate Gyrus" is this because it has no dentate gyrus cells, or just because they are marked up with a different label, such as "DG" instead? Correctly addressing seemingly trivial issues of usability like this is essential if neuroscience databases are to become a routine research tool in which users can be confident. This approach also highlights the advantage of search services being provided by groups other than the primary data providers. Even though providers may wish their data to be publicly accessible, they have a natural tendency to feel that, because it cost them such effort to assemble it in the first place, the end user should at least be expected to make a reasonable effort too, and should not be given to much at one time. This partly explains the presence of self-defeating access limitations on many web based resources. Third parties, on the other hand, such as the authors of the present system, have no qualms about delivering the entire contents of a database to the user in a single browsable file if this is what is most useful in a particular context.

¶ Dynamic web pages appear to the user just as static pages: the browser makes a request for a page and the server returns it. But whereas for static pages the server returns a file from the local file system, for dynamic ones it constructs the page in response to each request. The server can then tailor the contents of the pages to match the user's request, as in returning the results of a search.

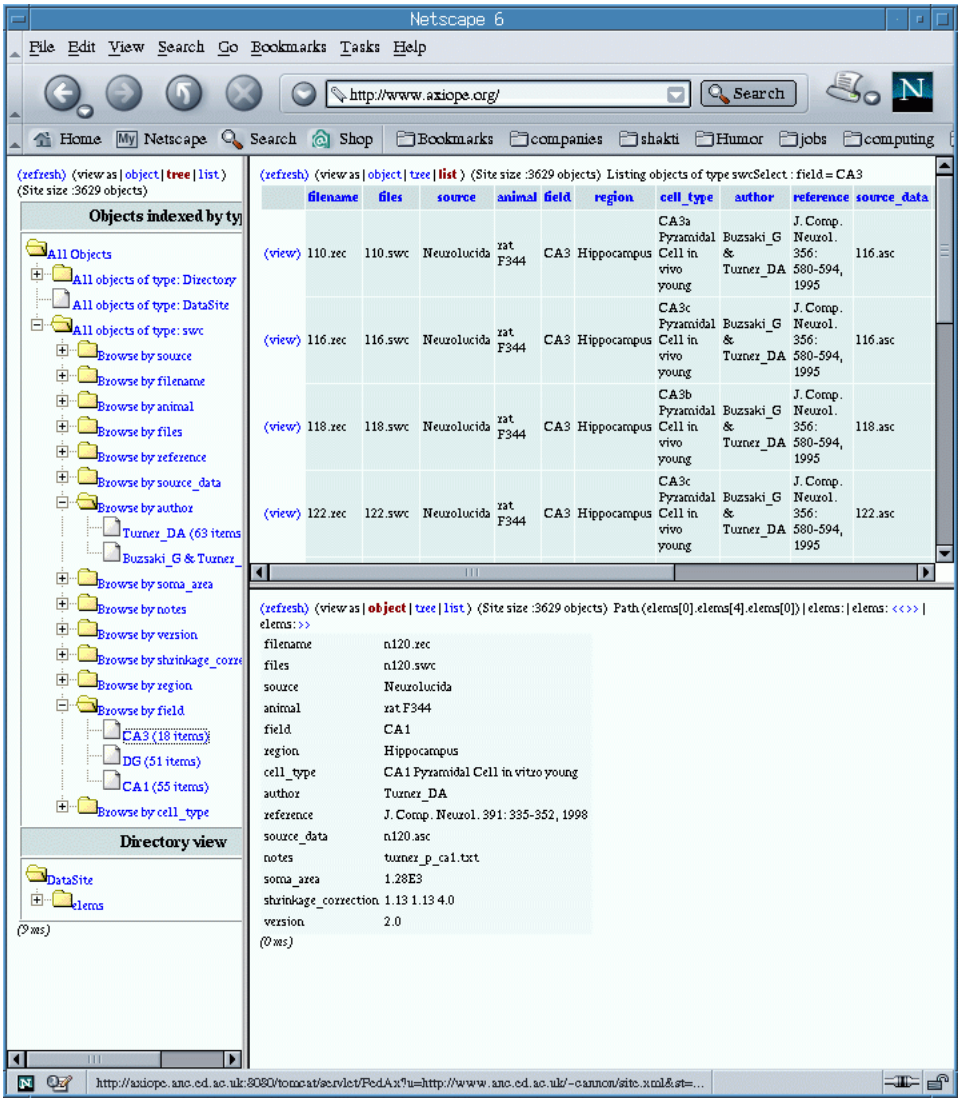

Figure 3. The morphology archive as viewed through the database engine. The metadata from the original site is used to stock an object database which can then browsed and searched according to any of the defined fields. Here the "Browse by Author" and "Browse by Field" options have been expanded. They present two alternative orderings of all items in the site. The upper right area shows the set of CA3 cells from the field-ordered view. The lower right panel shows the default record for a single cell. Links to images and the raw data point back to the original site.

4. Archiving and searching neuroanatomical data

The software described in the previous sections constitutes a general purpose system for creating structured representations of information (records), publishing these records, and searching the published data. It uses forms to specify the contents of records, but imposes no restrictions on origin, content, or life cycle of these forms. These questions are, nevertheless, important for the success of the system, because the efficiency of automated searching depends on the ability to cross-reference data from different

sites. This could be done by imposing strict constraints in acceptable forms, but such constraints are generally unappealing to data providers. It is therefore important to know how far the constraints can be relaxed without compromising the performance of the system.

4.1. Documentation schemes

Technical issues such as the standardization of data formats are often resolved by software engineers in the most technologically efficient way. For example, many libraries and other organizations are now adopting the Dublin Core (*www.dublincore.org*) for specifying how records describing books and other documents should be structured. Essentially this specifies, what information is required to describe a book, and exactly which names should be used for each items of information. For example, the first three required elements in the Dublin Core are "Title", "Creator", and "Subject and Keywords". Adoption of such standards ensures reliable cataloging and searching (for the essentially bibliographic information covered by the standard) but imposes certain constraints on the data providers. In general, for a database such as a library which is maintained by expert curators, such constraints are readily accepted as part the curator's professional activity.

For a distributed, voluntarily curated database, however, the imposition of strict requirement on the content of the information that is provided, may be a serious disincentive. Moreover, a complete neuroscience equivalent to the Dublin Core would be a very large specification indeed and therefore would pose even more of a hurdle to potential providers. The approach proposed here, therefore, is the complete opposite: not to ask "what would be the best way to have data provided so as to make the cataloging process as simple as possible?" but to ask "what is the bare minimum that providers must be asked to do so that the system can be made to work?". Asking the second question presupposes a certain investment in knowledge engineering in the cataloging sites, but it is more efficient that such an investment be made on a one-off basis in a small number of places rather than requiring all potential providers to learn and conform to an extensive set of standards.

The absence of standardization is not a goal in itself. On the contrary, it is hoped that widely used standards for the description of experimental data in particular domains of neuroscience will rapidly emerge. At this early stage, however, it is probably impossible to come up with good, or even widely acceptable, ones. Leaving the choice of standards up to the user should provide a mechanism for the most useful ones to gain widespread acceptance.

The main difficulty with standards efforts comparable to the Dublin Core in neuroscience is the heterogeneity of the data to be described. Not only the conclusions of an experimental investigation are required, but also the materials, methods and techniques employed since these almost inevitably have a bearing on the observations. Indeed, for experimental data to be made usefully available to others, the users need

all the information that the experimenters themselves require. That is, they need the equivalent of a well-kept lab book. The task of creating a data-site therefore merges seamlessly into that of creating a fully computerized local information management system. Conversely, were such systems in regular use, the task of creating a data-site would be rather simple – when the data provider was ready they could simply instruct the software to expose the parts that they wished to publish on the web.

The correspondence of a data-site to a "browsable lab book" makes it clear that the only people qualified to specify the details of what information should be included are the providers themselves. This does not rule out the use of existing naming systems such as NeuroNames[18], BDML[19](biophysical description markup language) or BrainML (*brainml.org*). Rather, it interposes a level of choice, since the user can adopt terms from any convenient systems in order to arrive at the most appropriate representation of their particular information. This is why the form construction process has been incorporated in the same "Builder" tool the data entry process in the current system. However, there is a risk that the ease of form creation and modification leads to the generation of many forms for essentially the same types of data. It should therefore be made even easier to query the database beforehand to see if a suitable form already exists than to create a new one from scratch. Whenever a data-site is created, the forms it employs are also included in the site. The pages for displaying records have links to the form used in building the record, so the forms can be easily downloaded and incorporated into a user's own site. The proposed form creation process is summarized in figure 4. It should be stressed, however, that issues of the life-cycle of forms are not fixed by the software: the figure shows one possible scenario, but it remains to be seen exactly how forms will be built, shared and reused in practice.

5. Discussion and future directions

Experience with the Duke-Southampton morphology archive led to the realization that intellectual property issues are central to the success of non-curated databases. The Axiope system has been designed to provide the necessary infrastructure for the creation and management of distributed data repositories while respecting the desire of data providers to maintain intellectual control of their work. The framework provided by present system allows a number of further issues of data management and reuse to be put into a concrete context.

5.1. Federation of data from different sources

The tools described in section three combine, by default, only those records from different sites which use the same form. Because developing good forms is likely to be an on-going process of gradual modification in many cases, there is a need for some form of federation of records constructed according to different forms, or according to different versions from the same family. The simplest way to do this is with a new record

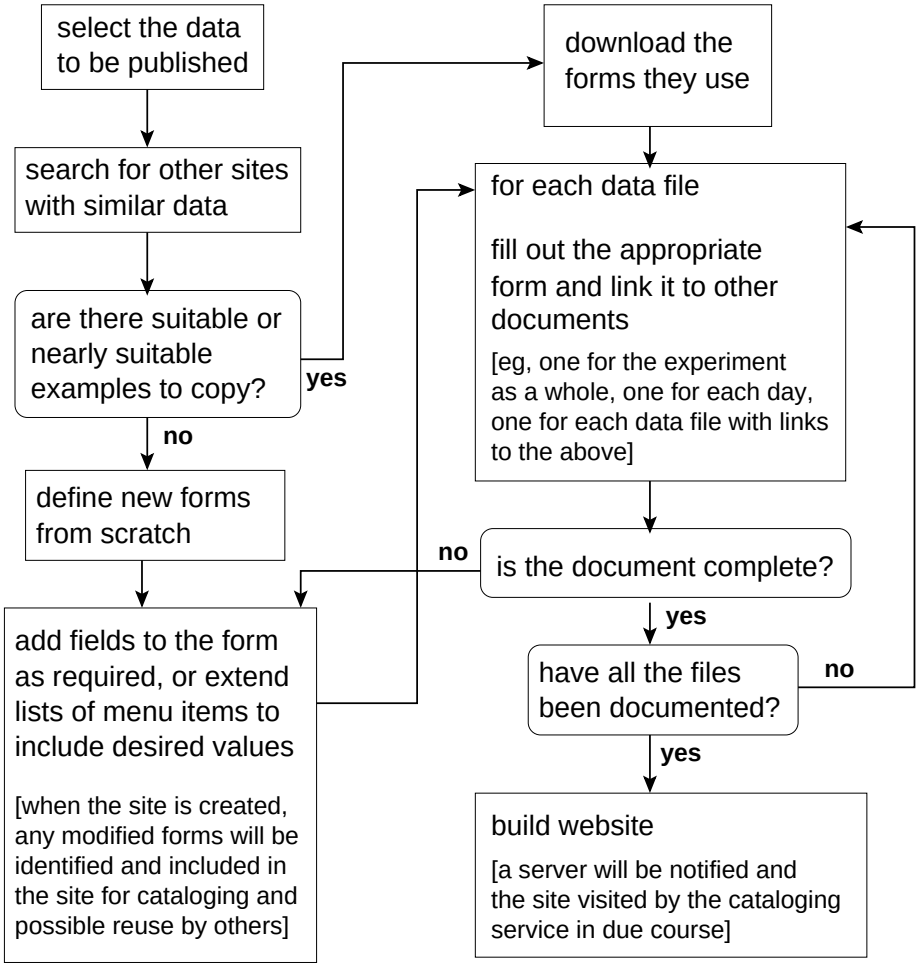

Figure 4. Data documentation with user defined forms. The flowchart shows the processes managed by the site building software and the questions raised for the user. It includes facilities for browsing forms which have already been defined and used by other researchers in order to minimize the creation of distinct forms for similar data. It depends on the catalog servers to track updates to forms so it can present the user with the most recent version. Note that, just as with a relational database, most applications will require a hierarchy of forms - one for the data which hardly changes over a whole series of experiments, one for the details of an individual file, and possibly several layers in between.

describing the mapping between the forms. This is a special case of the general need for additional information beyond that which is stored in the primary data-sites. As another example, an entry in a form may indicate that the data it refers to are actually measurements of the morphology of a neuron stored in a particular format. An entry in another form may indicate that the item it refers to is a software module capable of performing some form of analysis on the morphology data set. It would be useful if the catalog site could combine these two and present a user with the option of performing the analysis. Other sites might also wish to perform such analysis in advance and enter the results in the database. For this to work, either the cataloging software must be smart enough to spot the correspondence, or the data provider must be aware of the

analysis module and add a piece of information to their site, or there must be a facility for a user to spot the possibility and add the information themselves. The third route is the most plausible because it gives the responsibility for adding the information to the people who will benefit from it. Typically neither data providers nor software providers will be interested in exploring all the uses and applications of their work – only the end user has both the knowledge and interest to provide the necessary information.

The mapping between forms, and the information that a particular class of object is amenable to a particular form of analysis are examples of the very broad category of ontological information. Ontologies, as used in informatics, are specifications of the relationships among concepts. Often-cited examples include such statements as "plants and animals are living things; cats are animals; dogs are animals; cats have furry coats". Such information is useful because it can be employed with relatively straight-forward algorithms to improve access to information. For example, a search engine with data about cats in its database, and that also had the above ontology, could respond to the instruction "find me some living things with furry coats" even if the terms "living thing" and "furry coat" did not figure together in any of the individual database entries. Various systems are under development to assist in the creation of ontologies for diverse domains of knowledge (see *semanticweb.org* for further details or *protege.org* for a working public-domain tool).

The computer-assisted exploitation of neuroinformatics databases has a potentially vast requirement for ontological information to complement the primary data and meta-data. Just as with the data sites themselves, the optimal location for each section of this information may be a new data site under the control of the provider. However, it is also the case that relatively small amounts of ontological glue can make a very big difference to the performance of software systems. In this case, the information could be kept by the catalog server, perhaps citing the provider each time it is used so as to maintain an incentive for entering the information.

5.2. Quality Control

The biggest gain of curated databases over non-curated ones is in the standardization and rigorous assessment of material before it is included. This adds value to the data, but is a costly and time consuming process. It is also a rather severe filter which simply accepts or rejects information irrespective of the nuances of context or relevance. Within the framework of a non-curated distributed database, there is the possibility for an equivalent level of added value by a somewhat different mechanism. Instead of filtering other sites and keeping a reorganized subset, a site might also be set up to contain the quality assessments, cross referencing, synonym tables or other ontological information necessary to perform the filtering. That is, it would contain everything necessary for the conversion of part of the collective database into a curated database, but it would not itself contain the filtered data: that would remain in the sites of the originating labs.

Curation could then be done on-the-fly as and when necessary, making it clear to the user that what they were seeing was the product of several distinct resources. These would include the original data from various sites, value assignments, and the software that brings it all together. Another advantage is that the criteria used would then be fully formalized and future developments of the curation software might provide improved results without further input to the quality assessments. Another possibility is that data providers may, quite reasonably, wish to impose restrictions on the use of their data which would preclude its filtering and storage in another database. It is rather unlikely that such restrictions would, or even could, prevent the existence of another site providing quality assessments from which such filtering could be performed. On a less legalistic note, it is possible that data providers welcome the opportunity to provide quality assessments themselves. Such assessments might indicate, for example, what they regard as tentative, and what they consider the most important or robust results.

5.3. Data Enhancement

A simple example of how software modules from one data site can be combined with data from other sites was given in section 4.1. More complicated applications could involve performing a search on data that does not yet exist but for the generation of which there is a known pathway. For example, the L-measure analysis system[20, 21] performs sophisticated statistical analysis on the morphology of neurons. It would be possible to reconfigure such a package so that it could be used as part of a complex search. Then, for example, a search could be restricted, say, not just to cell morphologies of a particular type, but to cell morphologies for which a particular output of a given software module yields values in a specified range. In particular, one of the output quantities from L-measure is the number of bifurcation points in a structure, so a query could be composed asking for only those cells for with more than thirty bifurcations. In some cases, this information might have already been computed and stored with the cell structure, but for others it would require the software module to be run as part of the search. This then raises the question, as in any distributed application, of where the calculation is performed - on the site hosting the module, or the site doing the search, or the client requesting the information, or on the site with the data, or somewhere else entirely? The answer depends on many issues such as the volume of raw data involved, available network bandwidth and computing resources available. At present these are best resolved on a case by case basis until sufficient applications have accumulated to draw reliable conclusions.

5.4. Conclusion

Developing systems to allow the publication, sharing and reuse of neuroanatomical data involves a range of different tasks. Some tasks require information technology and software engineering expertise and others requiring detailed knowledge of the data itself. Here it has been argued that a highly distributed system best fits the spread of expertise

within the academic community. In particular, the choice of structures for describing experimental data should be left to the people who collect the data. It is in the interests of providers and users to develop standard nomenclatures: all the software need do is provide an environment in which this is possible. A distributed system also allows data providers to maintain complete control of their data, overcoming the widespread reluctance to transfer ownership to third parties.

Realizing that successful distributed databasing relies on fruitful collaboration between informaticians and experimentalists, with an equitable distribution of intellectual property is a first step to the design of usable systems. The system described here is one way to resolve some of these issues. Whether it, or a quite different approach is eventually widely adopted can best be settled by building them and observing how they are used.

Acknowledgements

This work is funded through the United Kingdom Medical Research Council, a visiting postdoctoral fellowship to Robert Cannon from the FWO, Belgium, and through EC grants QLG3-1999-00763 and QLRT-2000-02256 to Erik De Schutter.

References

[1] J D Van Horn and M S Gazzaniga. Databasing FMRI studies – towards a 'discovery science' of brain function. *Nature Reviews Neuroscience*, 3(4):314–318, 2002.

[2] J S Mirsky, P M Nadkarni, M D Healy, P L Miller PL, and G M Shepherd. Database tools for integrating and searching membrane property data correlated with neuronal morphology. *J Neurosci Methods*, 82(1):105–121, 1998.

[3] K E Stephan, L Kamper, A Bozkurt, G A Burns, M P Young, and R Kotter. Advanced database methodology for the collation of connectivity data on the macaque brain (CoCoMac). *Philos Trans R Soc Lond B Biol Sci.*, 355(1393):37–54, 2000.

[4] S H Koslow. Sharing primary data: A threat or asset to discovery. *Nature Reviews Neuroscience*, 3(4):311–313, 2002.

[5] D A Turner, R C Cannon, and G A Ascoli. Web-based neuronal archives. In *Neuroscience Databases - A Practical Guide*, page in press. Kluwer Academic, Norwell, Mass, 2002.

[6] D Mott, D A Turner, M Okazaki, and D V Lewis. Interneurons of the dentate-hilus border of the rat dentate gyrus: morphological and electrophysiological heterogeneity. *J. Neurosci.*, 17:3990–4005, 1997.

[7] D A Turner, X G Li, G K Pyapali, A Ylinen, and G Buzsaki. Morphometric and electrical properties of reconstructed hippocampal CA3 neurons recorded in vivo. *J. Comp. Neurol.*, 356:580–594, 1995.

[8] R C Cannon, D A Turner, G Papyali, and H V Wheal. An on-line archive of reconstructed hippocampal neurons. *Journal of Neuroscience Methods*, 84(1-2):49–54, 1998.

[9] G K Pyapali and D A Turner. Denervation-induced alterations in CA1 pyramidal neurons following kainic acid lesions in rats. *Brain Research*, 652(2):279–290, 1994.

[10] P Vetter, A Roth, and M Hausser. Propagation of action potentials in dendrites depends on dendritic morphology. *J. Neurophysiol.*, 85(2):926–937, 2001.

[11] S J Nasuto, J L Kirchmar, and G A Ascoli. A computational study of the relationship between

neuronal morphology and electrophysiology in an Alzheimer's Disease model. *Neurocomputing*, 38-40:1477–1487, 2001.

[12] J L Kirchmar, S J Nasuto, R Scorcioni, S D Washington, and G A Ascoli. Effects of dendritic morphology on CA3 pyramidal cell electrophysiology: a simulation study. *Brain Research*, in press, 2002.

[13] D E Donohue, R Scorcioni, and G A Ascoli. Generation and description of neuronal morphology using L-Neuron: a case study. In G A Ascoli, editor, *Computational Neuroanatomy: Principles and Methods*. Human Press, Totowa, NJ, 2002.

[14] W Koh and B H McCormick. Brain microstructure database system: an exoskeleton to 3D reconstruction and modeling. *Neurocomputing*, in press, 2002.

[15] M L Hines and N T Carnevale. Neuron: a tool for neuroscientists. *The Neuroscientist*, 7:123–135, 2001.

[16] J M Bower and D Beeman. *The Book of Genesis*. Teleos Publishing, Los Angeles, 1994.

[17] N H Goddard, M Hucka, F Howell, H Cornelis, K Shankar, and D Beeman. Towards NeuroML: model description methods for collaborative modelling in neuroscience. *Philos Trans R Soc Lond B Biol Sci*, 29(352):1209–1228, 2001.

[18] D M Bowden and R F Martin. Neuronames brain hierarchy. *Neuroimage*, 2:63–83, 1995.

[19] D Gardner, K H Knuth, M Abato, S M Erde, T White, R De Bellis, and E P Gardner. Common data model for neuroscience data and data model exchange. *J Am Med Inform Assoc*, 8(1):17–33, 2001.

[20] G A Ascoli, J L Krichmar, R Scorcioni, S J Nasuto, and S L Senft. Computer generation and quantitative morphometric analysis of virtual neurons. *Anat Embryol (Berl)*, 204(4):283–301, 2001.

[21] G A Ascoli, J L Krichmar, S J Nasuto, and S L Senft. Generation, description and storage of dendritic morphology data. *Philos Trans R Soc Lond B Biol Sci*, 356(1412):1131–1145, 2001.