

ModelDB: New Features and Future Goals

Thomas Morse¹, Luis Marenco¹, Sushel Kambampati¹, Michael Hines¹,
Michele Migliore^{1,2}, Gordon Shepherd¹, Ted Carnevale³

¹Yale University School of Medicine, New Haven CT, ²National Research Council, Palermo, Italy,

³Department of Psychology, Yale University, New Haven CT

Tom.Morse@Yale.edu

Abstract. ModelView is a powerful model exploration tool that summarizes and displays information about Neuron models that can be hard to find searching the model code alone. ModelView-provided summary information assists an investigator by providing familiarity with the specifics of the model so questions about the model are easier to pose and less time consuming to answer. In addition new in-development XML driven interoperability tools will help investigators in understanding models and work towards facilitating model language translation to multiple simulator platforms. A new page was presented that assists modelers by allowing them to add models on one concise page without the need to have an account. Another new feature that was presented showed how alternate model code versions can be accessed from the same model. Finally a complex search tool was demonstrated that allows arbitrary boolean combinations of fields for searches for models in ModelDB.

1. Introduction

Parts of this talk are from a recent electronic demonstration at the Human Brain Project annual meeting (Hines et al, 2004). We will review three new features in ModelDB, demonstrate an evolving complex search tool, and present our future goals. The new features are 1) modelView, a new feature where NEURON simulator model information obtained during model run time forms a browsable description of the neuronal or network model, 2) a public model create page that does not require an account, and 3) a new feature which allows the inclusion of alternative model code versions in ModelDB models. The complex search tool can be used to search for models matching arbitrary combinations of interactively selected fields. Future goals include interoperability development with Visiome (Usui 2003; <http://platform.visiome.org>) and continued development of simulator independent XML descriptions of models (which include the morphology, ion channels, connectivity, and more).

2. ModelView

The availability of a large number of models in ModelDB (Migliore et al 2003), <http://senselab.med.yale.edu/senselab/modeldb>, helps investigators test their intuition of model behavior and provides building blocks for future modeling applications to the interpretation of experimental findings.

However the NEURON (Hines and Carnevale 2001) model legacy code entered by publication authors was generally not developed with presentation as a high priority. The original code can be difficult to analyze and it sometimes happens that variables are reset so that the values at run time are different than the first values indicated in the top of the code. ModelView overcomes these problems by providing a (run-time state) preview of the properties of a model (anatomy and biophysical attributes). Having this information available for viewing in ModelDB lets investigators quickly develop a conceptual picture of the model structure and compare parameter differences between runs. It makes it possible to ask detailed questions about the model that would have been time-consuming to answer without ModelView.

ModelView in Action

Example: CA1 pyramidal neuron: as a 2-layer NN and subthreshold synaptic summation by Poirazi et al 2003

This example demonstrates how ModelView can explore a NEURON model. We look at the report generated after starting the tool (fig 1).

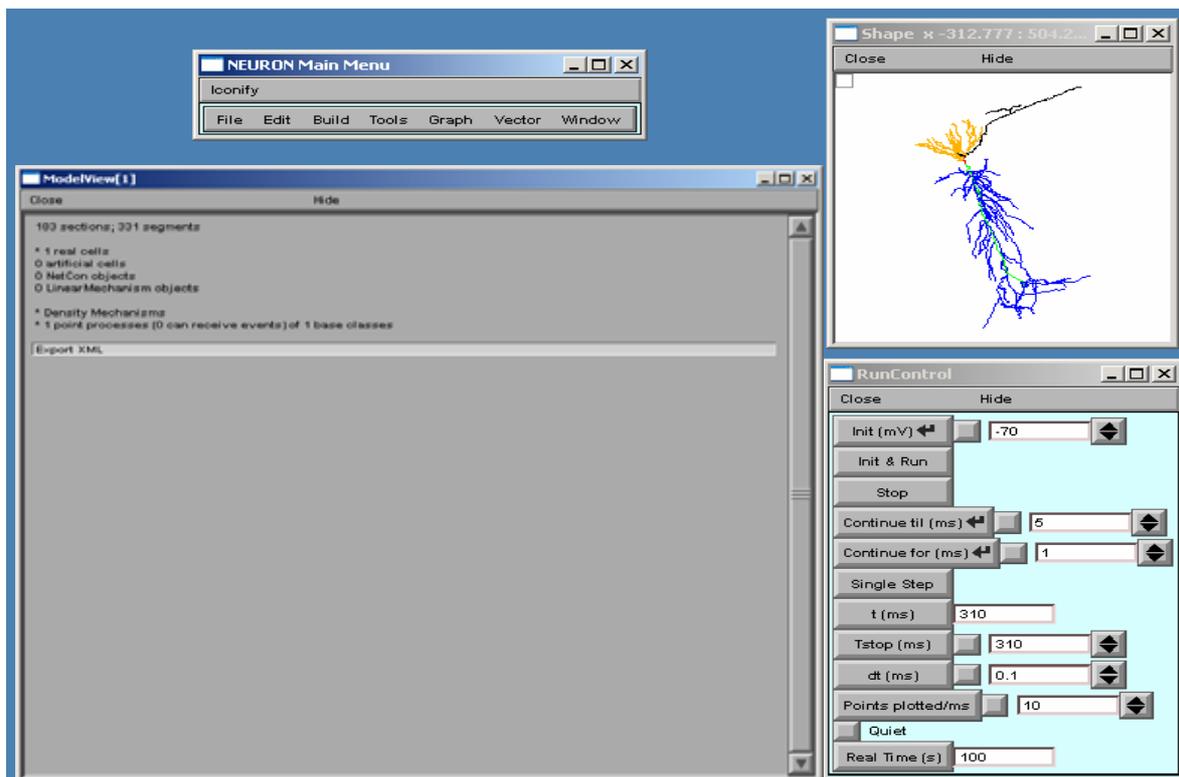


Figure 1. Starting ModelView

ModelView opens its hierarchical browser with a top-level summary of the model. In NEURON a cell is divided into sections which are further divided into computational nodes called segments. The first item in the summary is the total number of sections and the total number of segments, for all the cells in the model. The second and final summaries report the numbers of real and artificial cells and density mechanisms and point processes, the later two include ion-channels, receptors, and electrodes. The Poirazi example contains 1 real cell and density mechanisms and a point process. We explore the model by clicking on the asterisk lines to expand the text tree (or contract if already expanded). Clicking on “* 1 real cells”, and then clicking on the newly appeared “*root soma[0]” displays an analysis of the occurrence of the duplicate and the unique sets of parameters distributed throughout the cell (fig 2 left).

After the total number of sections and segments in this particular cell and the related nseg parameter distribution, the number of inserted mechanisms is reported. Expanding the number of inserted mechanisms (23 here) lets the investigator know which ion-channels are used in the model (fig 2 right).

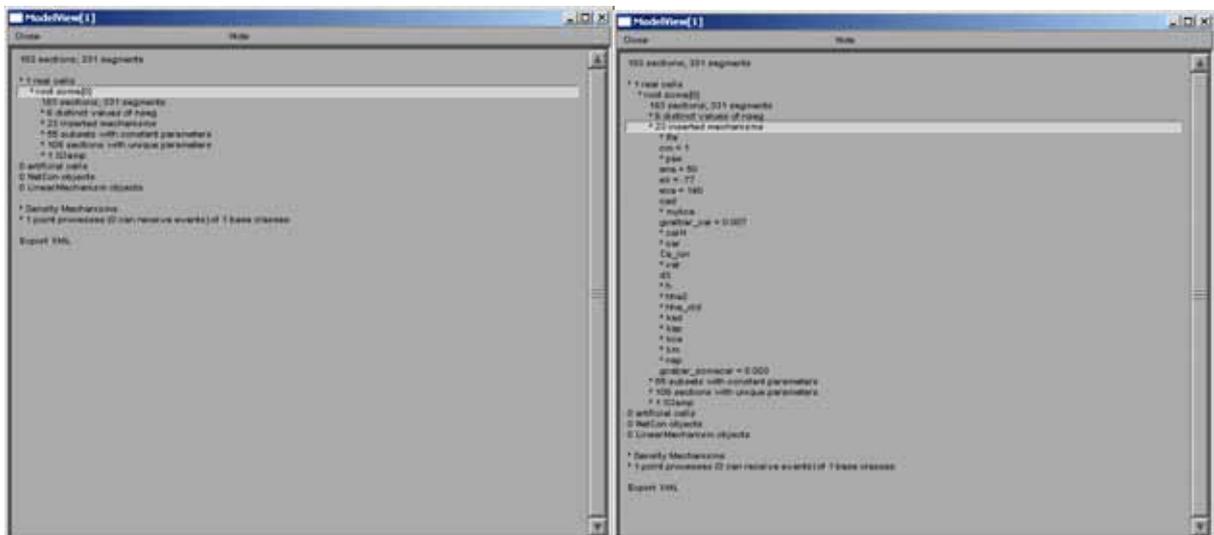


Figure 2. Exploring sections (left) and mechanisms (right) in ModelView.

Expanding the “* pas” line (fig 3) shows how ModelView truncates text display when there are too many values to display with a message “... or more distinct values”.

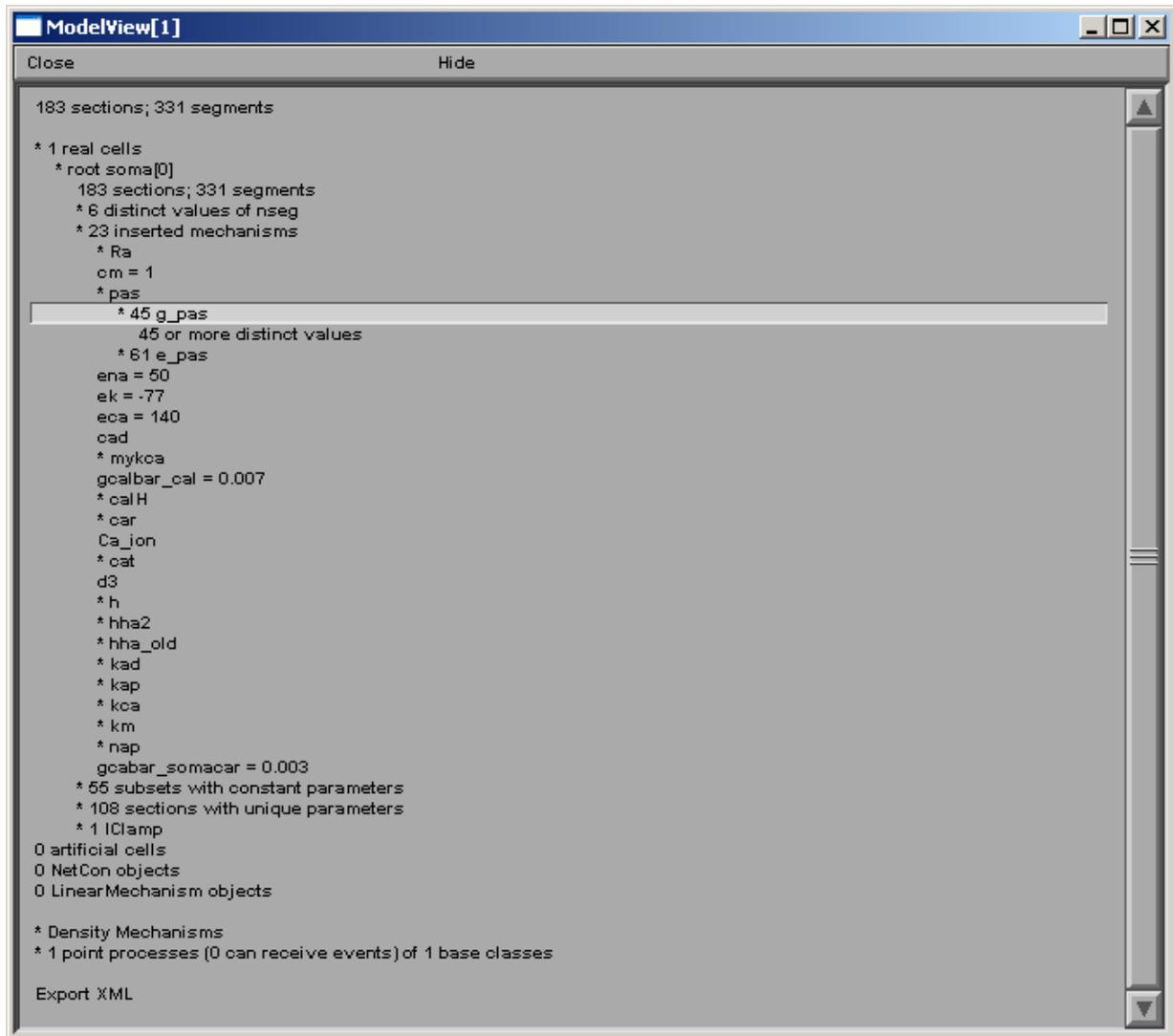


Figure 3. The parameter values in the mechanisms are truncated when too numerous.

If one of these too-lengthy-to-display items is clicked upon, a graph of the values of the parameters along an axis indicating distance from the soma and a 3D shape image with values expressed as color intensities are displayed (fig 4). The image on the left shows the range graph when `g_pas` selected (right when `e_pas` selected).

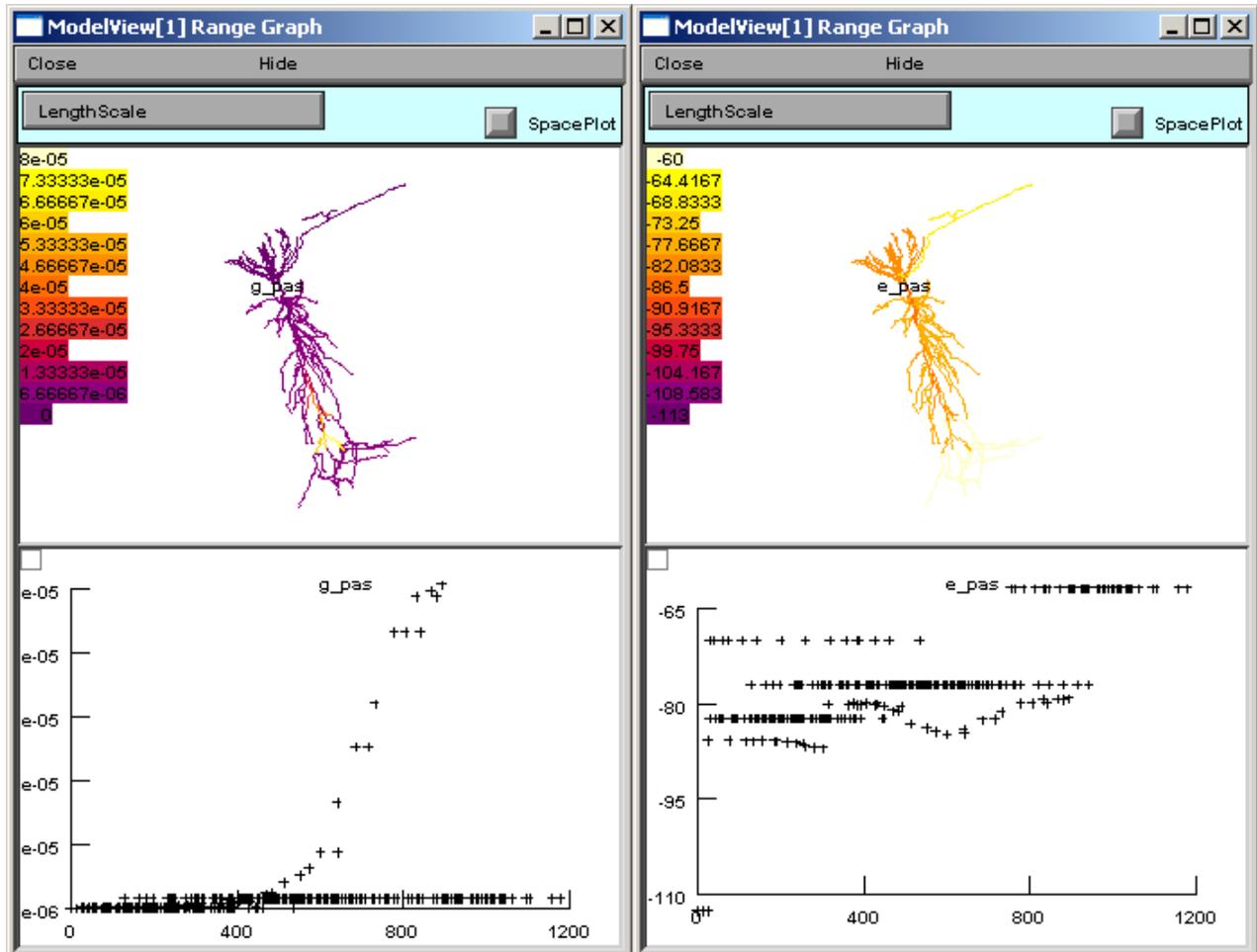


Figure 4. Plots of spatially distributed parameter values for `g_pas` (left) and `e_pas` (right).

ModelView analyzes the sections to find the sets of sections that share a parameter (or a set of parameters) set to the same value(s). There are 64 sets of subsets of constant parameters in this model. The number of sections that have unique values of a parameter are also calculated and displayed (109 here). These large numbers of sets are indicative that there are heterogeneous values for some parameters. For a simpler model, these numbers help identify the combinations of values that are used, and also their spatial distribution, while for a more complex model, such as this one, the line items are links to graphs as described above. When a parameter from these lists is selected (fig 5 left), for example $eca = 140$, the part of the tree that contains that value of that parameter is highlighted in a shape plot (fig 5 right).

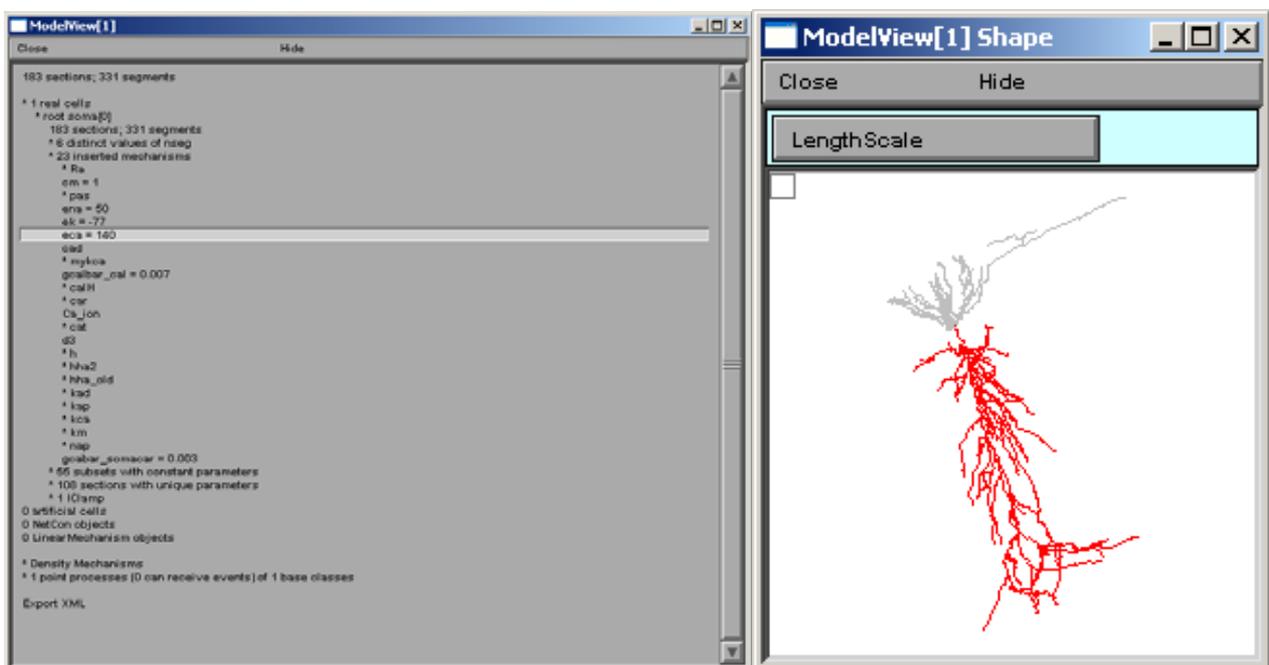


Figure 5. Graphical indication of where a parameter is set to a particular value.

The final group of top-level items, density mechanisms and point processes will expand into text trees that summarize and create graphs that display the distribution of these items. The Homogeneous Parameters item lists the parameters that occur with just a single value, unlike Heterogeneous Parameters, which contain lists of parameters which take multiple values. When a homogeneous parameter is selected, e.g. el_hha2, then a graph of the distribution of that parameter is displayed (fig 6).

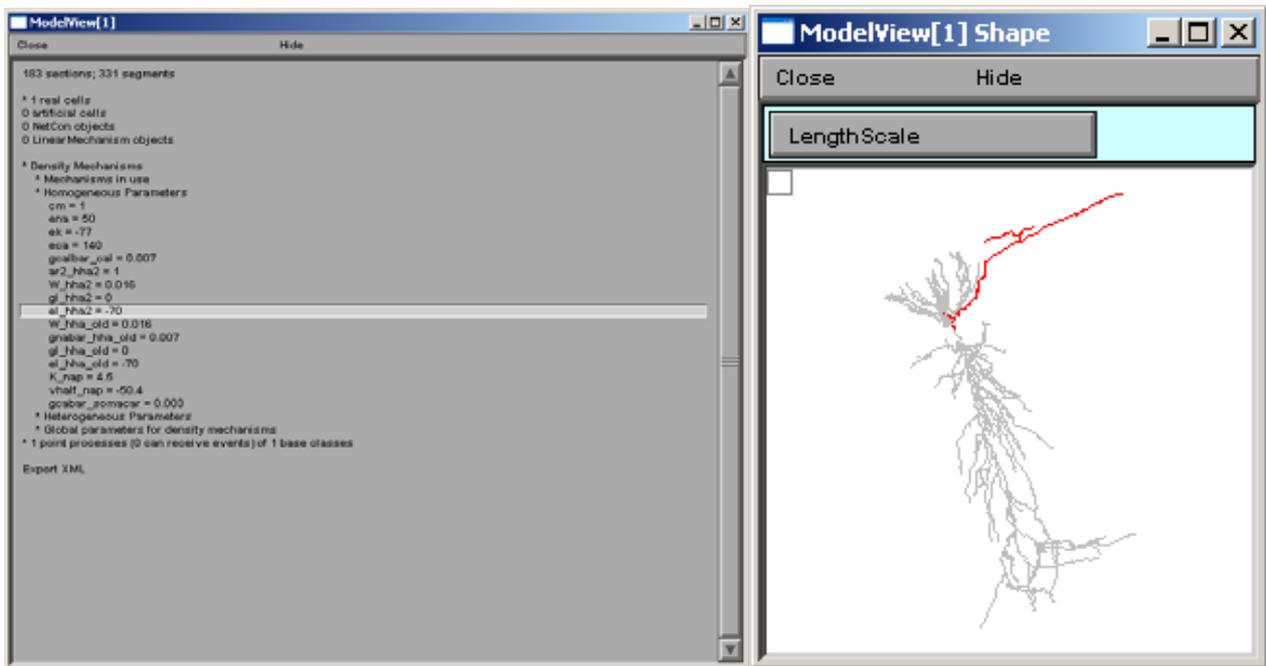


Figure 6. Another example of highlighting where a parameter is set to a particular value.

If a heterogeneous parameter is selected, e.g. `ar2_hha_old`, a term used in a Na channel activation function, the range plot shows the overall complexity, while the text provides a place to explore the details of the distribution (fig 7).

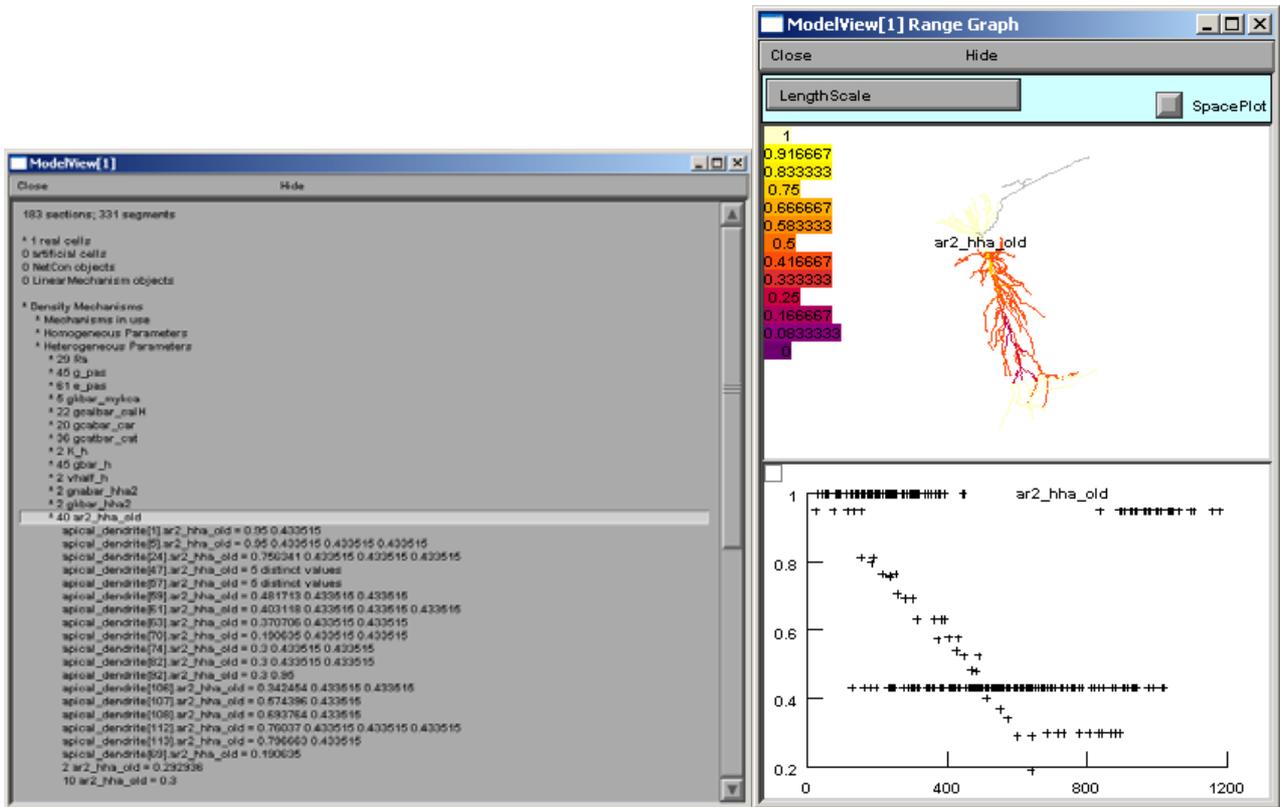


Figure 7. A heterogeneous parameters graphically indicated values and domain.

Finally, in this example model, there is one point process, a current clamp, which when selected in the ModelView browser, will show in a shape graph that it is positioned at the soma (fig 8).

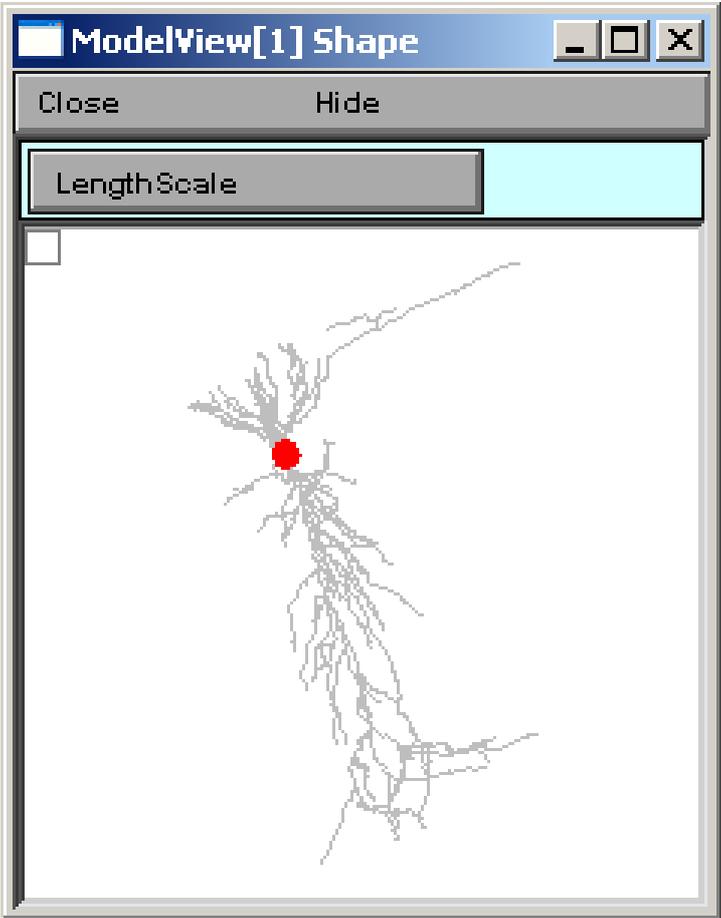


Figure 8. Position of Current Clamp.

3. ModelView and ModelDB

We plan on incorporating a ModelView window as an alternative window to the model file browsing window in ModelDB. Investigators will be able to browse the morphology and biophysical specification for the NEURON models stored in ModelDB. This browser is under development, but preliminary views (fig 9 below) illustrates its similarity to the NEURON version.

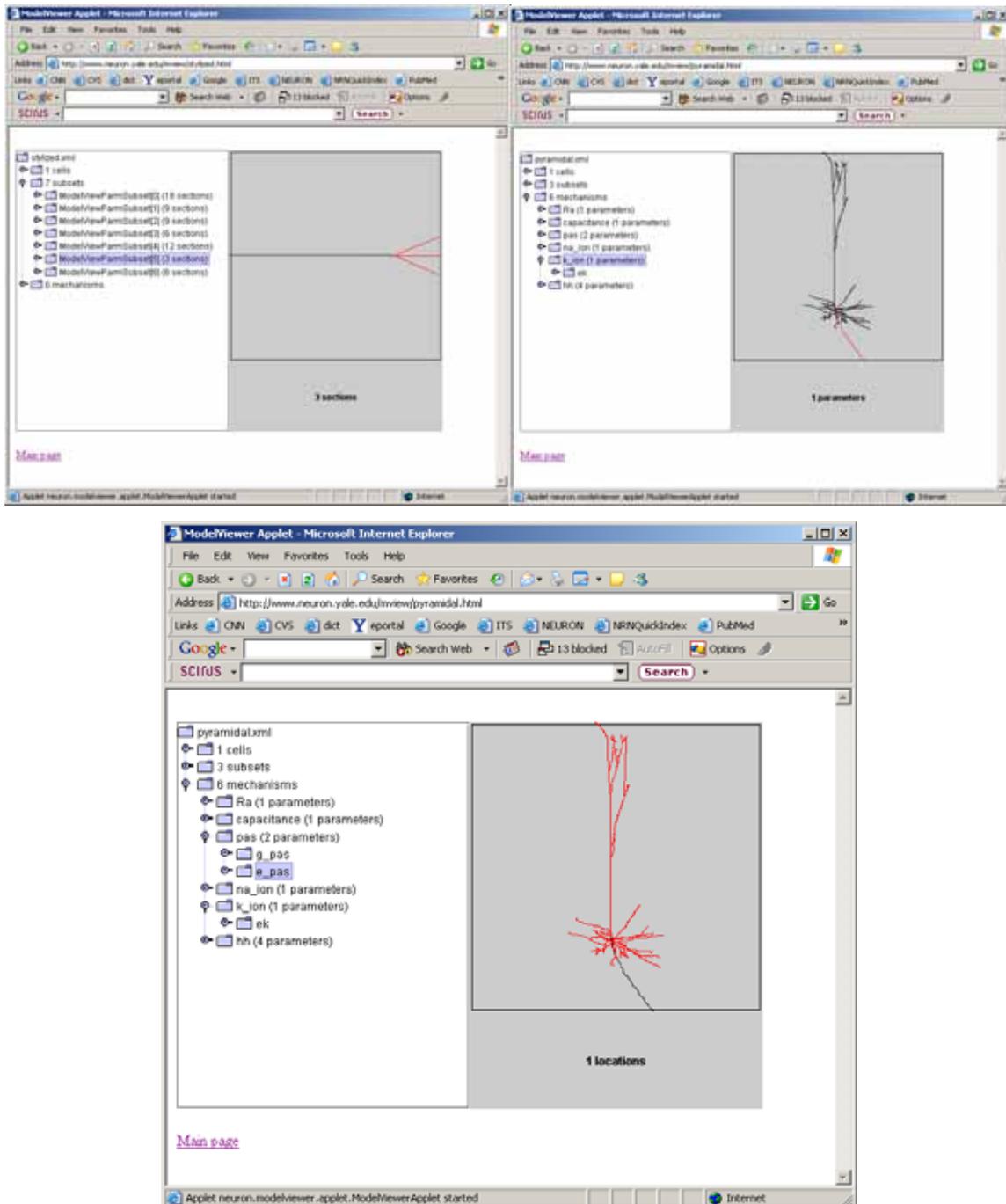


Figure 9. Preliminary ModelView in action on the web.

4. ModelView and Interoperability Future Goals.

ModelView can output an XML description of the model's morphology and biophysical specification. ModelView uses MorphML to describe the morphology of the cell. We are collaborating with the MorphML group (see reference) to evolve the MorphML specification for interoperability. The biophysical XML specification is under development and we have plans to provide translation tools to previous and developing standards. We will continue to refine the XML output to make it suitable for communication with neuroinformatics tools as well as other modeling software such as Catacomb and GENESIS. We will adopt, develop through informal collaboration and communication, and promote NeuroML (see reference) (whose goal is to establish standard descriptions of computational models that are simulator independent), BrainML (see reference) (whose goal is to develop standards in the general description of neuroscience data). We will also attempt to implement the tools required to import and export models with a third specification, CellML (see reference) (whose purpose is to store and exchange computer-based biological models.)

5. Public Model Creation

A new feature of ModelDB is the ability to create a model without an account. A web page is provided where the modeler can upload model code and provide a name and description and password all on one page. A URL for the model acts as the login and the user specifies a password.

We hope that having this form will make it easier for modelers to submit their model since they will be able to do so without first requesting an account.

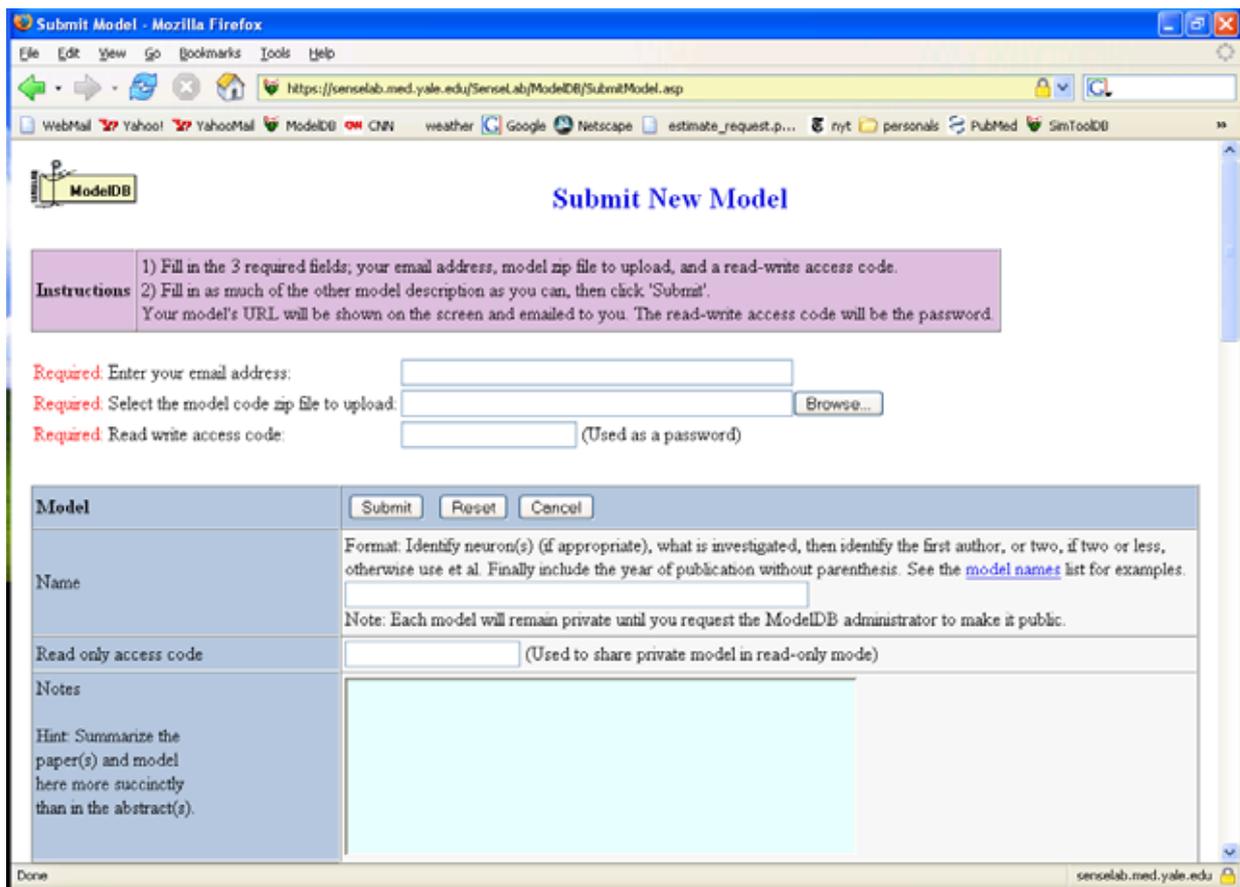


Figure 10. A new method of submitting a model without an account.

6. Alternative Model Code

Another new feature of ModelDB is the ability to store alternative model code. This allows the storage of earlier versions, versions targeted for specific platforms, or the ability to store a local version of a model that is also supplied as a web link.

The figure shows a model entry from Poirazi et al 2003 which contains as its default model code the most recent version which runs under both windows and unix. The authors original linux code and another linux version which was created for the auto-launch feature to work in ModelDB are also available as indicated in the pop-down list of model version titles displayed in the bottom middle of the figure. Selecting a model version from the pop-down list changes the files displayed and available through the download button.

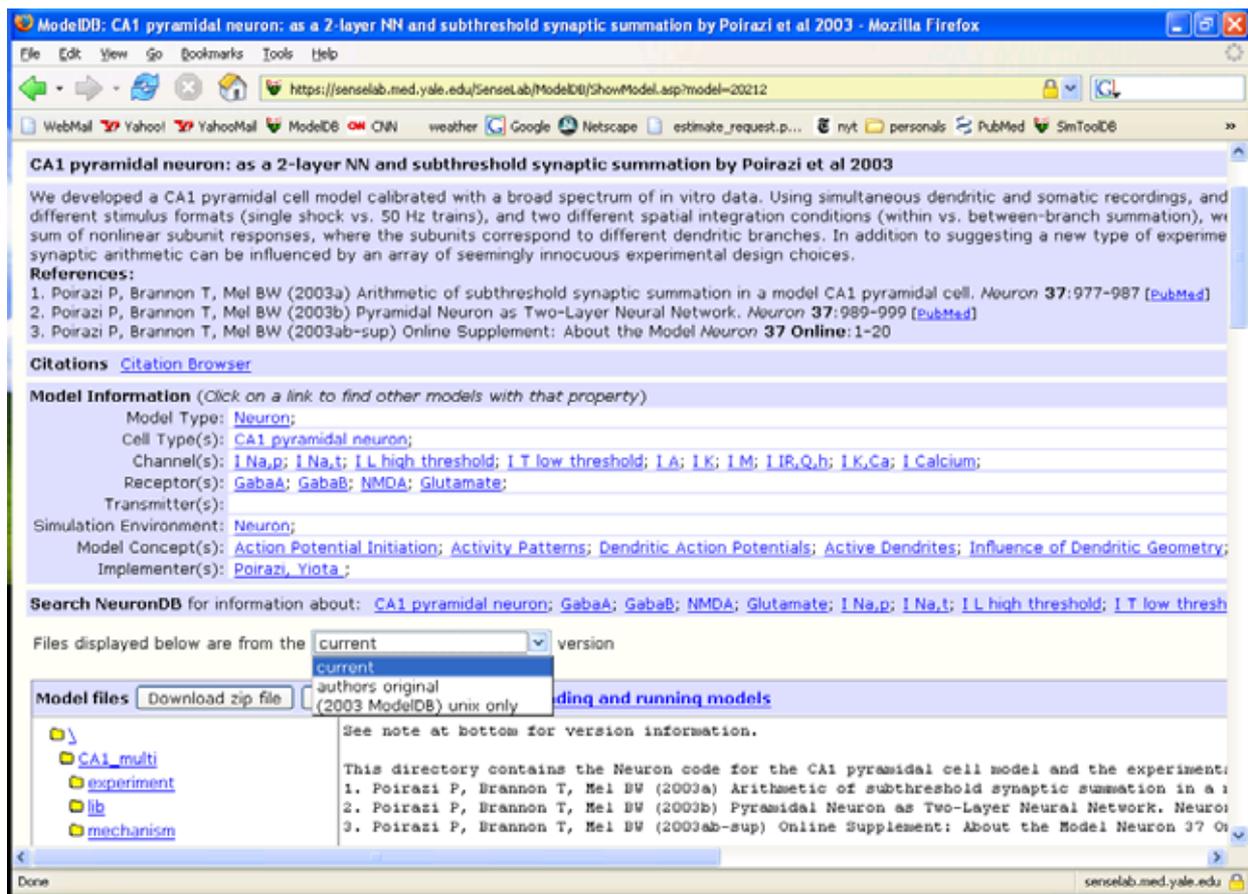


Figure 11. Display and access multiple model versions from a pop-down list.

7. Complex Search

A complex search tool has been available for some time and continues to be refined. This tool provides searches of arbitrary combinations of fields. In this example shows the results of a search for models on the topic of coincidence detection that involve hippocampal cells. This search was setup by browsing to the complex search tool under Utilities in the architecture pages of SenseLab. The concept field was interactively added and the look-up list was selected to find the coincidence topic, and then the neuron type field was added and the hippocampal neurons were added by selecting them (while holding down the shift key) from the lookup list. Finally the search button was pressed (fig 12).

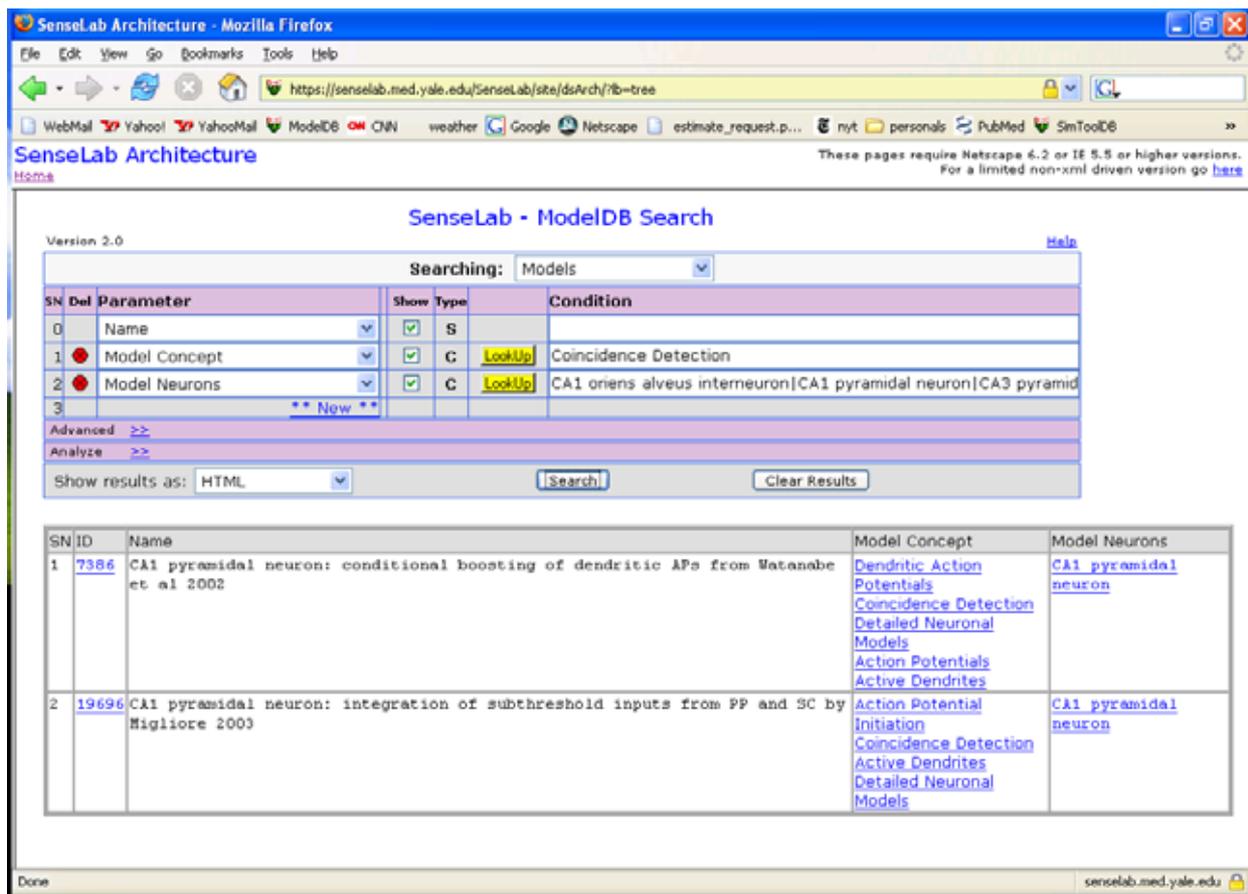


Figure 12. Complex search tool.

Acknowledgement

We are grateful for the support of NIH's Human Brain Project Grant number 5P01DC004732-04.

References

1. BrainML: <http://brainml.org/>
2. CellML: <http://www.cellml.org/>
3. Hines, M.L. and Carnevale, N.T. NEURON: a tool for neuroscientists. *The Neuroscientist* 7:123-135, 2001.
4. Hines, M.L., Kambampati S, Morse T.M. ModelView: Compact Display of Parameters for NEURON Models. Human Brain Project Annual Meeting, NIH, Bethesda MD, April, 2004
5. Migliore M, Morse TM, Davison, AP, Marengo L, Shepherd GM, Hines ML. ModelDB Making models publicly accessible to support computational neuroscience. *Neuroinformatics* 1:131-134, 2003.
6. MorphML: <http://germain.umemat.maine.edu/faculty/crook/MorphML.html>
7. NeuroML: <http://www.neuroml.org/>
8. Usui S. Visiome: Neuroinformatics Research in Vision Project. *Neural Networks* 16:1293-1300, 2003.