NeoCortical Builder: A Web Based Front End for NCS

Jakub Berlinski[1]
Nathan M. Jordan[1]
Sergiu M. Dascalu[1]
Marlon D. Chavez[1]
Devyani Tanna[1]
Laurence C. Jayet Bray[2]
Cameron Rowe[1]
Roger V. Hoang[1]
Frederick C. Harris, Jr.[1]

Brain Computation Lab
http://www.cse.unr.edu/brain/

[1]Computer Science and Engineering
University of Nevada, Reno
Reno, NV, USA

[2]Bioengineering
George Mason University
Fairfax, VA, USA

Abstract

The NeoCortical Builder (NCB) is a web-based application for building brain models and creating simulation input and output parameters. NCB is also able to launch simulations on the NeoCortical Simulator (NCS). NCS is a neural network simulator which takes various brain models with corresponding simulation parameters and provides output. NCB was designed to streamline the process of model creation and the creation of simulation input and output parameters. NCB has the ability to drastically change the way various types of scientists interact with brain simulators. NCB creates a modern graphical interface to streamline the creation of brain simulations.

keywords: Computational Neuroscience, Brain Models, Simulation, Graphical User Interface.

1 Introduction

Brains are complex systems which are difficult to fully understand. The problem is brains are extremely complicated to model and in many different brain modeling applications the interface is confusing or outdated. One example of a competing product is NEURON [2]. NEURON also simulates brain models but its interface is outdated and sometimes confusing. Along with having an interface, NEURON also allows users to use text based input files. NCS has tried different ways to create brain models and simulation parameters, such as name and duration, for brain simulations. The latest method involved using text based input files. These files were complicated to construct as they required very specific syntax that the user would need to know. NCB solves this problem by providing a modern graphical user interface which the user will interact with to create their brain model as well as their simulation parameters. This graphical user interface will be implemented as part of a bigger front end web application.

Web based applications are becoming more prevalent because the Internet is more reliable and faster than ever before. Ease of use and accessibility are key features of web based applications and NCB provides that for any user that uses it. To run NCB all a user needs is access to a device capable of running a modern web browser. The goal of NCB as a web based application is to provide the end user with a portable way to create brain models.

The rest of this paper is structured as follows: Section 2 covers what NCS is and the technologies used to create NCB. Section 3 covers the design of NCB. Section 4 goes into detail of model creation. Section 5 discusses conclusions and future plans for NCB.

2 Background

2.1 NCS

The NeoCortical Simulator is a neural simulation tool developed at the University of Nevada, Reno [3]. NCS has the ability to model millions of neurons in real time. It allows neuroscientists to build, test, and visualize a brain model that they design using NCB. NCS can run on a single computer or across a cluster of computers. Most importantly, NCS uses both CPUs and NVIDIA GPUs to generate the maximum computational power [6].

The goal of NCS is to create an approachable neuro simulator that can be used by any neuroscientist with-out the need to know how to write any code. Using NCS, users will be able to quickly launch simulations and receive their outputs in real time with the click of a button. NCS also aims to be an extensible application where different parts of a model are computed with different inputs that the user defines. This allows NCS to have many more combinations of components which create the most realistic simulation possible. Finally, NCS aims to do all of this in the most efficient manner. Harnessing the power of GPUs, NCS will be able to use the maximum amount of resources to provide the simulation of neurons in the most efficient way [5].

2.2 Technologies

Many web technologies were used in the creation of the NeoCortical Builder. The core technologies that were used to build NCB are HTML5[8] and CSS[7]. The following technologies were also used: JavaScript, JQuery, AngularJS, Bootstrap, X-editable, and Flask.

- JavaScript is an object-oriented computer programming language which is used by NCB to create interactive effects within the web application [9]. For example, JavaScript would detail what would happen when a certain element is selected by the user. Additionally, JavaScript is used as a way to hold brain models and simulations on the back end.

- JQuery is a library for javascript which simplifies DOM manipulation [12]. NCB uses JQuery to edit the HTML of the web page to show or hide various different elements. JQuery is being replaced with AngularJS because AngularJS is a more robust library for manipulating the web page.

- AngularJS is a javascript library which allows for single web page applications. Usually when creating a web page there are many different pages that are accessed when using the site. For example, selecting "about" on a web page will load a whole new HTML file for that page. Angular allows one big html file where elements are hidden or shown instead of having multiple files [4]. NCB uses AngularJS as a framework to set up a single page web application.

- Bootstrap is a set of design templates which use HTML and CSS to create various types of elements on the web page [13]. NCB uses many different bootstrap design templates from web page structuring to the design of buttons and dropdown menus.

- X-editable is a javascript library which allows DOM elements to be made editable. The elements can then be edited through either a pop-up or in-line within the webpage [10]. In NCB, this was one of the most important features because it allowed users to edit the various brain model and simulation parameters quickly and easily.

- Flask is a python based web server framework [11]. NCB uses flask for communication with the running simulation.

3 Design Overview

The front end web application for NCS will be made up of many different parts as shown in Figure 1. The NeoCortical Builder will be in charge of two major components, the brain builder tab will be used to create brain models and the simulation builder tab which will create simulation runtime parameters. The next two tabs will be done by the NeoCortical Repository team (NCR) [1]. The first of their two tabs will be the reports tab which will show various graphs and other information pertaining to the running simulation. Their second tab will be the model repository tab which will be in charge of holding all the brain models in various databases. The final section will be the virtual robot tab which will use the simulation output to affect a robot.

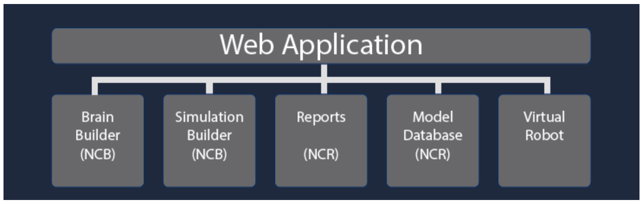


Figure 1: The complete web application framework.

3.1 Brain Builder

The brain builder tab is split into a three column layout as shown in Figure 2. The left column is where the pre-built models will be loaded from various model databases. Additionally, the models in this tab may be filtered instantly by typing into the filter input box as shown in Figure 3. Selecting a model will expand a quick view which details the model parameters as shown in Figure 4. In this column there is also a button which prompts the user to import models from a file or export models to the user's computer. There is an options button which allows the user to customize the colors of various portions on this tab.

The middle column is where the model currently being worked on is shown. In this column, there is a button which allows the user to add various brain

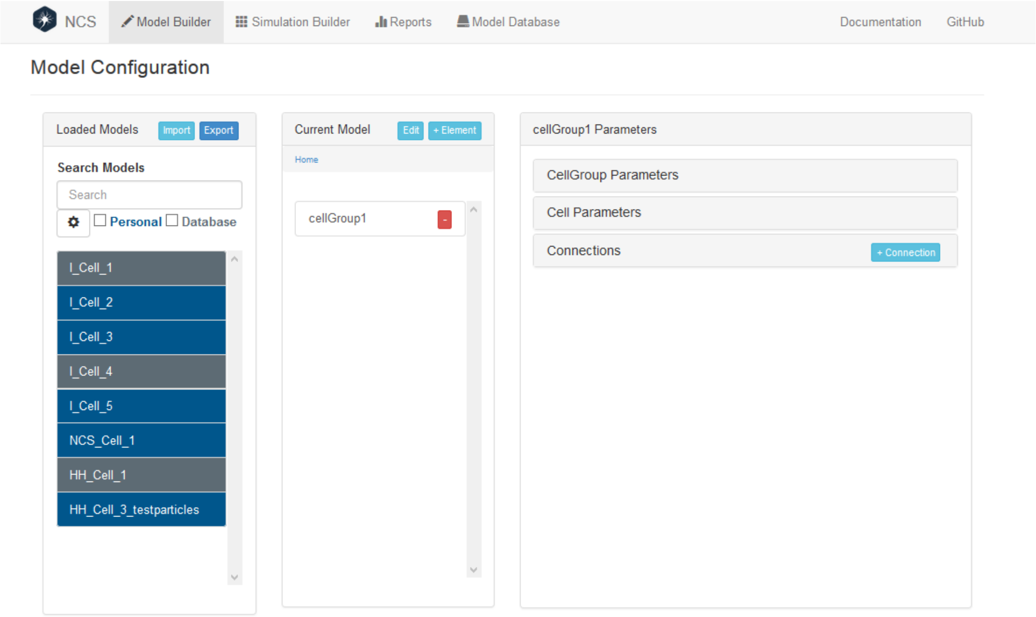


Figure 2: The model builder tab within the NCB front end web application.

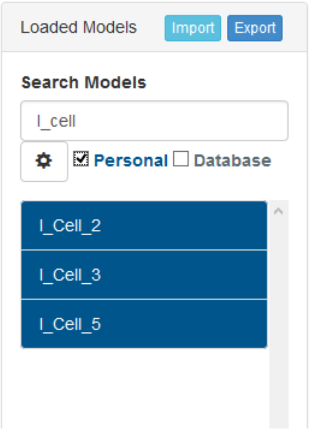


Figure 3: Filtering the models in the left menu.

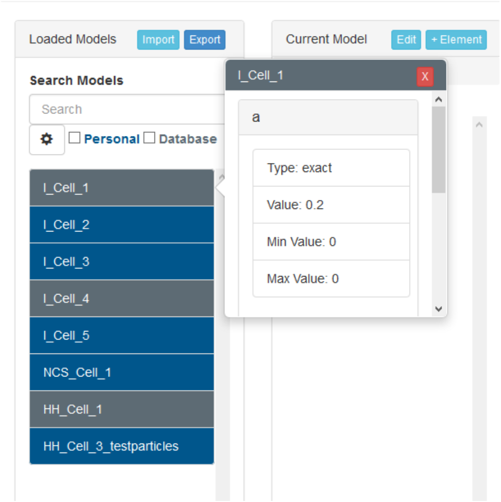
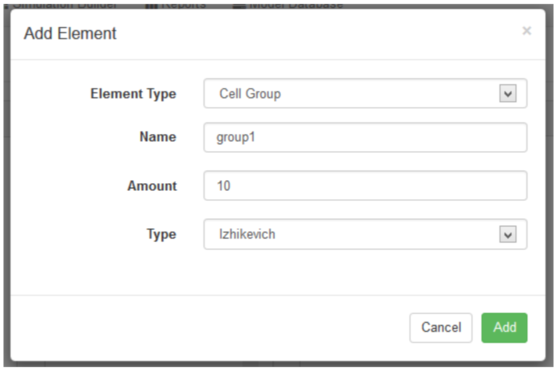


Figure 4: The popover when a user selects a pre-built model in the left menu on the model builder tab.

model elements including neurons, groups, and aliases shown in Figure 5. After an element is added, the user is able to select the red minus button to remove that element. Groups are able to be nested within each other and when this occurs a breadcrumb trail is created across the top of the column shown in Figure 6. Additionally, there is a second button which allows the user to edit the current model parameters including name, description, and author. Selecting a model in this column will populate the right column.

The right column is where all the selected model's parameters are shown. The parameters are split into three main categories. The categories are cell-group parameters, cell parameters, and connections. Certain cells like the Hodgkin-Huxley cell-type may have channels so in these cases a third category will be dynamically created to show this. To add connections the user can select the "+ Connection" button and a pop-up appears prompting the user for various parameters which are needed to create a connection. Furthermore, all parameters will be editable when the user selects them. When a user clicks on an editable parameter a small pop-up appears where the user can then enter their new parameter value as shown in Figure 7.



Element Type: Cell Group
Name: group1
Amount: 10
Type: Izhikevich

Figure 5: When the user selects the "+ Element" button they are shown this modal which asks for various element details.

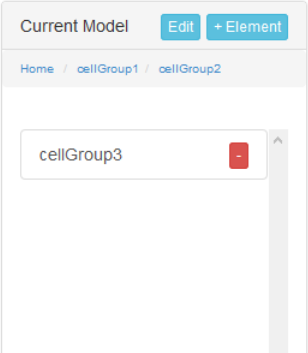


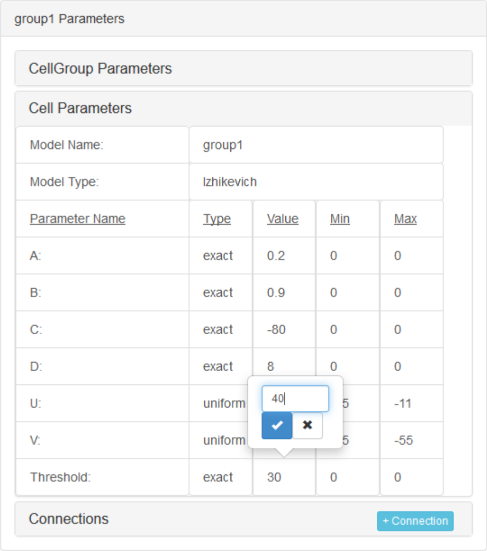
Figure 6: Multiple cell-groups nested within each other.

3.2 Simulation Builder

The simulation builder tab, as shown in Figure 8 is also split up into a three column layout. The left column is where the user may add simulation parameters. Some simulation parameters include name, duration, and seed. These simulation parameters are used as a way to identify a specific simulation and also to give NCS some important information about how a simulation should be ran.

The middle column is where the user will input the runtime parameters for the simulation. Some input parameters are stimulus type, frequency, input elements, start and end times, and many more. These parameters indicate where currents will be applied on the built brain model.

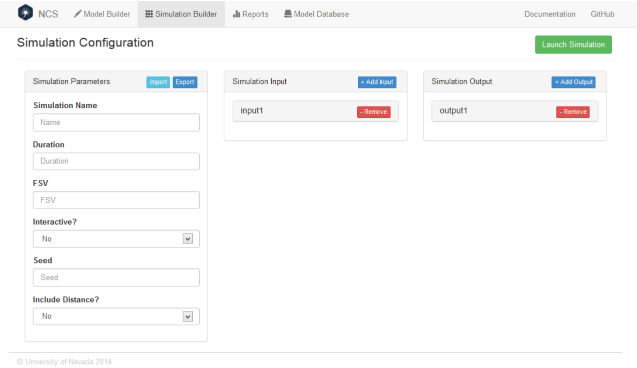
The right column is where the user will input the output parameters for the simulation. Some output parameters are output types, report types, output elements, frequency, start and end times, and many more. These parameters indicate specific elements the user wants to observe.



Model Name:	group1
Model Type:	Izhikevich

Parameter Name	Type	Value	Min	Max
A:	exact	0.2	0	0
B:	exact	0.9	0	0
C:	exact	-80	0	0
D:	exact	8	0	0
U:	uniform	40	5	-11
V:	uniform		5	-55
Threshold:	exact	30	0	0

Figure 7: Editing a model parameter in the right column.



Simulation Configuration

Simulation Parameters

Simulation Name: Name
Duration: Duration
FSV: FSV
Interactive?: No
Seed: Seed
Include Distance?: No

Simulation Input: input1

Simulation Output: output1

© University of Nevada 2014

Figure 8: The simulation builder tab within the NCB front end web application.

4 Usage Scenario

When the user first opens the NCB web page they are shown the front page which is Figure 2. As we mentioned in the previous section, there is a three column layout with the first column being the preloaded models, the middle column being the current model, and the right column being the various parameters of the current models selected elements.

The first thing a user would want to do if they were creating a fresh model is to select the add element button. After the user selects that button they are shown the pop-up modal, shown in Figure 5, which asks for the type of element to add and various parameters like the name and amount. The user may add as many elements as they need using this method. Figure 9

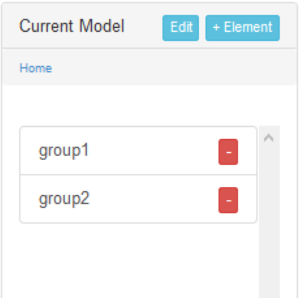
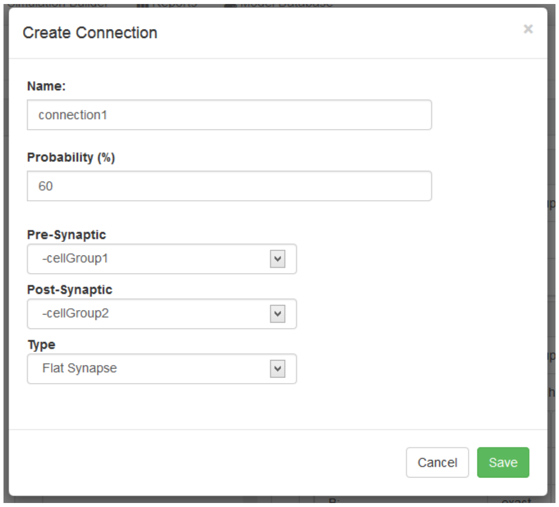


Figure 9: Two cell-groups have been added to the current model and they are shown in the center column.


Create Connection

Name: connection1

Probability (%) 60

Pre-Synaptic -cellGroup1

Post-Synaptic -cellGroup2

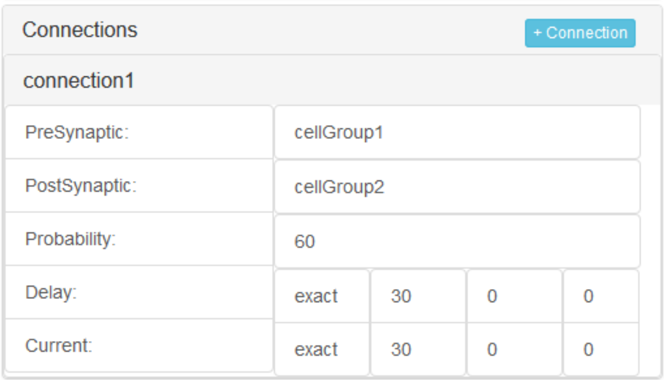
Type Flat Synapse

Figure 10: The modal which prompts the user for values for a connection.

shows a model with two cell-groups.

Next the user would edit some parameters in the model they are building. When the user selects a cell group the right column becomes populated with the selected element's parameters. Selecting one of the values brings up a small pop-up where the user may update and save their new value as shown in Figure 7. All parameters may be edited to take a specific set of values. When a user selects "exact" they need to only enter one value for that parameter. When a user selects "uniform" they are required to enter two values, a minimum and a maximum. Finally if a user selects the "normal" option they are required to enter a mean and standard deviation value.

Currently in our working model there are two cell-groups that are disconnected from each other. To make a complete brain model the user will need to connect them to each other using a connection. To add a connection between different elements, the user must select the "+ Add Synapse" button and they will be shown the modal in Figure 10. Here the user must enter a name for their connection, a probability that the connection will fire, two elements which are being connected, and finally a type of connection. The two types of connections that NCB supports are flat synapses and NCS synapses. After adding a connection, it will appear as a collapsible element within the connections in the right column as shown in Figure 11.



connection1

PreSynaptic:	cellGroup1			
PostSynaptic:	cellGroup2			
Probability:	60			
Delay:	exact	30	0	0
Current:	exact	30	0	0

Figure 11: An example of a connection between two elements, group1 and group2.

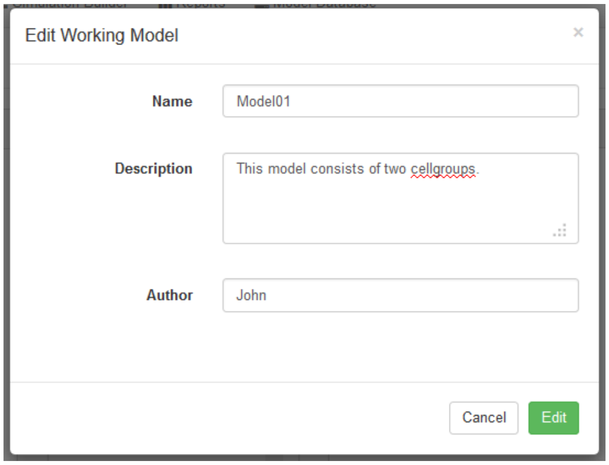
After adding a connection between these two elements the model is simple but complete. The user may want to personalize their current model by giving it a name, a description, and setting the author. To do this the "Edit" button will be selected and a modal pops up where the user may enter these values. This is shown in Figure 12. After the model is edited the user may choose to export the model to their computer. Furthermore, if the user does not want to create a new model he or she may import a previously built model. With a complete model the user may navigate to the simulation builder tab and begin editing their simulation parameters.

In the left column on the simulation builder page the user will enter information about the simulation's name, how long the simulation should run, the seed that will be used, and various other information. For the simulation the user will also need to fill out what type of input and output the user wants from the simulation.

5 Conclusion and Future Work

5.1 Conclusion

NCB provides many features that make the creation of brain models and simulations easier and more intuitive. With a graphical front end for creating brain models, the user does not need to get into creating convoluted text files that may very quickly get very large and unreadable. NCB makes the power of NCS



Edit Working Model

Name: Model01

Description: This model consists of two cellgroups.

Author: John

Figure 12: The modal which prompts the user for values to personalize their model.

accessible to any scientist who wants to simulate brain activity. NCB has been designed with simplicity in mind to allow anybody the ability to use it.

5.2 Future Work

Although NCB contains many features for creating brain models, it is still only half of the final web application. NCB deals with the creation of brain models and simulations while NCR deals with the saving of the models in a database and displaying the reports from the simulation. In the future both of these projects will be combined to create the final and complete front end web application for NCS.

References

[1] E. Almachar, A. Falconi, K. Gilgen, D. Tanna, N. M. Jordan, R. V. Hoang, S. M. Dascalu, L. C. J. Bray, and J. Frederick C. Harris. NeoCortical Reporsitory and Reports: Database and Reporsitory for NCS. In *Proceedings of the International Conference on Software Engineering and Data Engineering (SEDE-2014)*, 2014.

[2] N. T. Carnevale, M. L. Hines, and J. W. Moore. NEURON for empirically-based simulations of neurons and networks of neurons. Available at `http://www.neuron.yale.edu/neuron/`, 2012. (Retrieved September 24, 2012).

[3] P. H. Goodman, R. V. Hoang, L. C. Jayet Bray, and F. C. Harris, Jr. The Neocortical Simulator NCS. Available at `http://www.cse.unr.edu/brain/ncs/`, 2012. (Retrieved September 24, 2012).

[4] Google. AngularJS. `https://angularjs.org/`, 2014. (Retrieved May 19, 2014).

[5] R. V. Hoang. *An Extensible Component-based Approach to Simulation Systems on Heterogeneous Clusters.* PhD thesis, University of Nevada, Reno, 2014. `http://www.cse.unr.edu/~fredh/papers/thesis/PHD-010-Roger-Hoang/dissertation.pdf` (Retrieved July 24, 2014).

[6] R. V. Hoang, D. Tanna, L. C. Jayet Bray, S. M. Dascalu, and F. C. Harris, Jr. A Novel CPU/GPU Simulation Environment for Large-Scale Neural Modeling. *Frontiers in Neuroinformatics*, 7, 2013.

[7] M. D. Network. CSS. `https://developer.mozilla.org/en-US/docs/Web/HTML`, 2014. (Retrieved May 19, 2014).

[8] M. D. Network. HTML5. `https://developer.mozilla.org/en-US/docs/Web/HTML`, 2014. (Retrieved May 19, 2014).

[9] M. D. Network. JavaScript. `https://developer.mozilla.org/en-US/docs/Web/JavaScript`, 2014. (Retrieved May 19, 2014).

[10] V. Potapov. X-editable. `http://vitalets.github.io/x-editable/`, 2012. (Retrieved May 19, 2014).

[11] A. Ronacher. Flask. `http://flask.pocoo.org/`, 2014. (Retrieved May 19, 2014).

[12] The jQuery Foundation. jQueryUI. `http://jqueryui.com/`, 2014. (Retrieved May 19, 2014).

[13] Twitter. Bootstrap. `http://getbootstrap.com/`, 2014. (Retrieved May 19, 2014).