

Object-Oriented SNNAP: From Design to Implementation

Yidao Cai, E vyatar Av-Ron, John H. Byrne, Douglas A. Baxter
 Department of Neurobiology and Anatomy,
 University of Texas-Houston Medical School, Houston TX 77030

Introduction

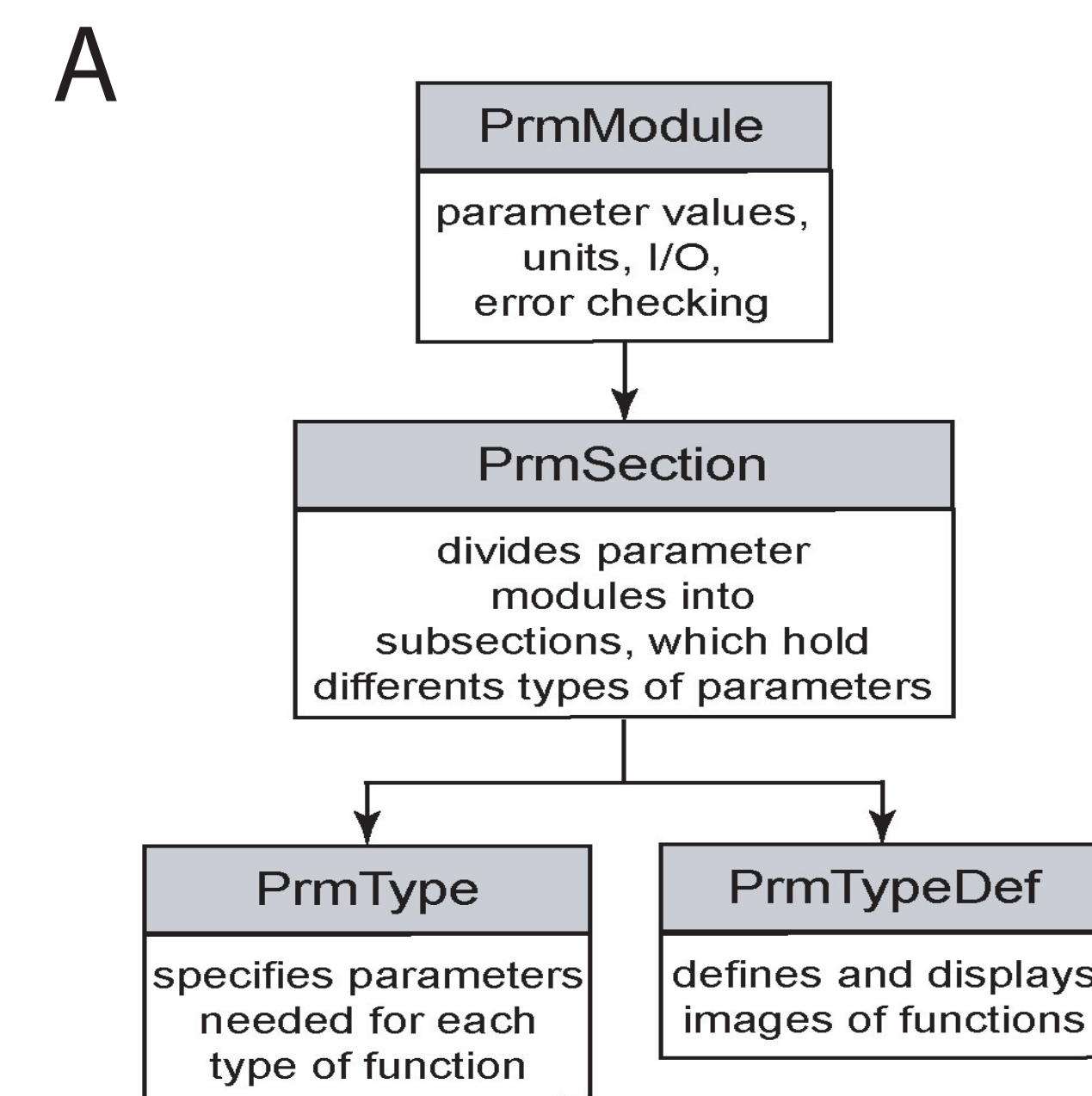
SNNAP (Simulator for Neural Networks and Action Potentials) is a tool for modeling neurons and neural networks. SNNAP is written in Java and is thus platform independent, and can be run under any operating systems. SNNAP is user-friendly, with a graphic user interface to input model parameters and control simulations.

Although the current version of SNNAP is written in the OO programming language Java, it is based on a functional architecture originally implemented in the programming language C. As a result, it is difficult to add new functionalities to SNNAP as codes implementing one function often scattered in more than one files and changes for one function may affect other functions. An OO design will minimize this problem. It will greatly simplify the means of incorporating new functionality such as new computational elements (e.g., biochemical and gene models), graphic components (e.g., network designer), drag-and-drop tools for building models, better error checking functionality, and easier maintenance. In addition, an OO design will permit implementation of a multithreaded architecture for parallel processing of model simulations.

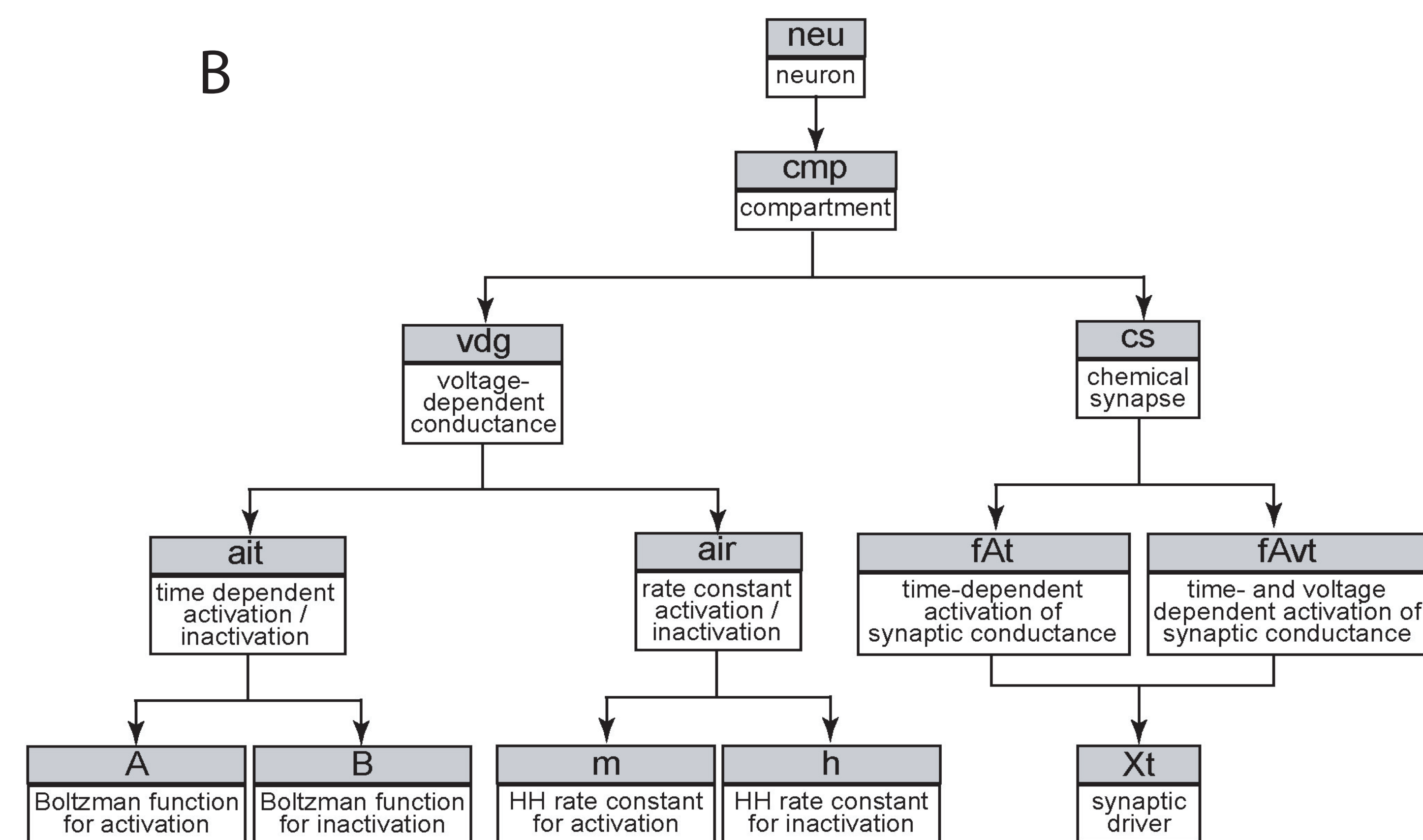
We implemented the core functionality of SNNAP using the OO design and with better error reporting, online help and simulation history keeping. In addition, the use of a newly developed data compression algorithm and the JAVA 2D graphics resulted in better display, higher quality printing and faster simulation speed.

1. OO design results in cleaner implementation

A. Class hierarchy of OO SNNAP parameter modules. The base class PrmModule, is inherited by all the classes that contain model parameters, both to store values for simulation and display values in appropriate equations.

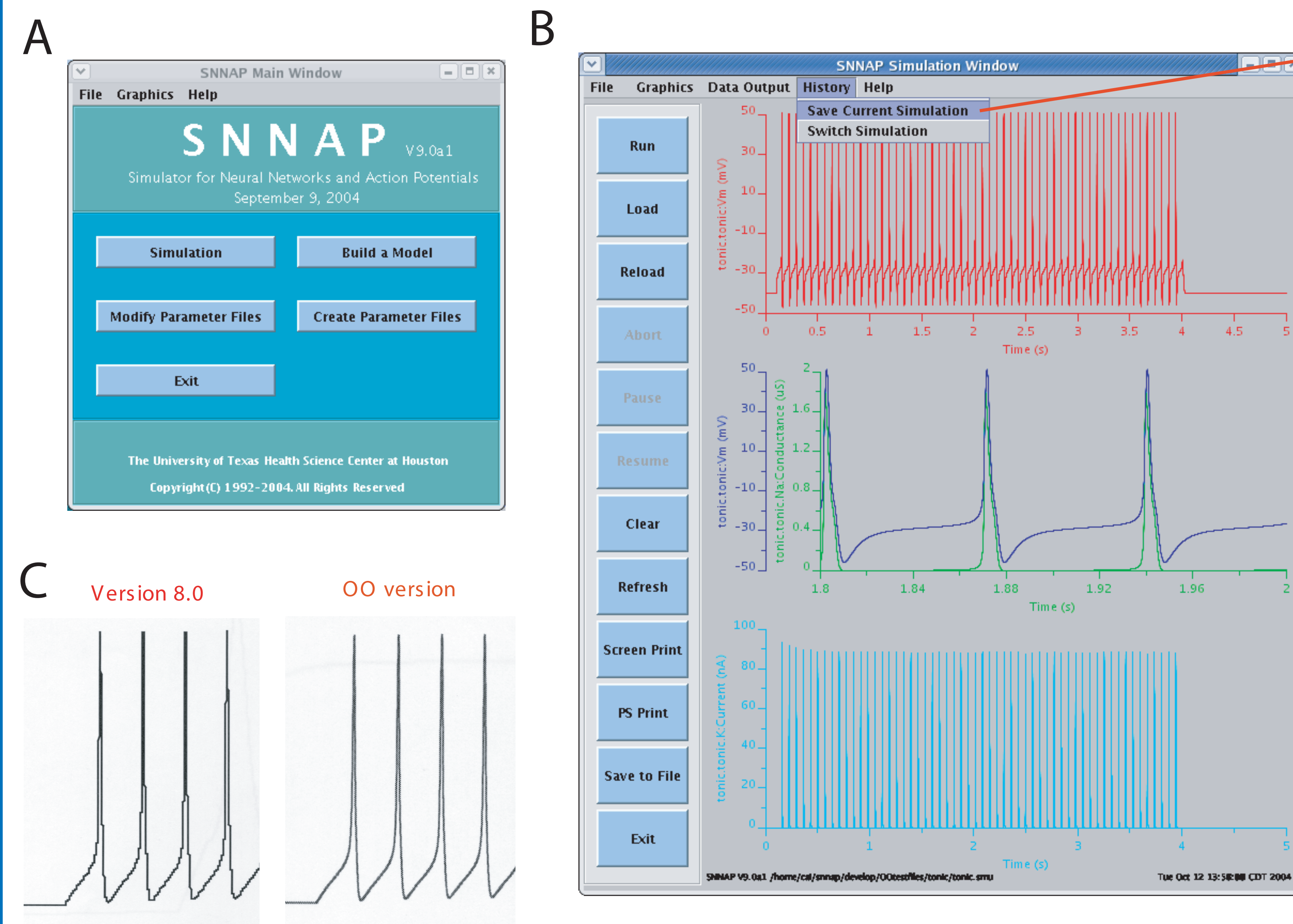


B. An example of the OO structure for neural network modeling in the new version of SNNAP. All element classes are derived from the base class. The more complex elements are at the top and the more basic objects are towards the bottom. A Neuron is composed of compartments, which contain currents and SynapticConnections. A current can have one of two formalisms, rate constant activation/inactivation or time-dependent activation/inactivation. SynapticConnections is composed of either time-dependent or time- and voltage-dependent conductances, both using a synaptic driver.



2. Better graphics and printing quality

A. The main SNNAP window is used to launch a simulation and build or modify a model. B. The simulation window shows an action potential resulting from a simulation of a Hodgkin-Huxley type model. C. Comparison of printing quality of version 8.0 (screen quality) and OO version (300 dpi).



3. Improved graphics efficiency and simulation speed

OO SNNAP uses a more efficient compression algorithm for graphic display. It first compressed and saves data into memory at print quality resolution (assuming 10 inch width, 600 dpi), then further compresses the data for optimal screen display.

	Screen Vector Data			Simulation Time (msec)		
Simulation condition	0.05 s, 1 us	1 s, 10 us	1 s, 1 us	0.05 s, 1 us	1 s, 10 us	1 s, 1 us
Raw data points	50000	100000	1000000	50000	100000	1000000
Version 8.0	1626.00	16817.25	20340.25	1041	2684	15645
OO impl. v8.0 algm	1207.25	15743.75	17545.75	1193	2808 (+4.6%)	20321 (+29.9%)
OO new algorithm	590.50	851.75	854.50	1094	1868 (-30.4%)	11708 (-25.1%)

Screen vector data are the actual data used for the graphic engine to plot curves on screen. Numbers are averages of 4 test curves.

How Do I Get SNNAP?

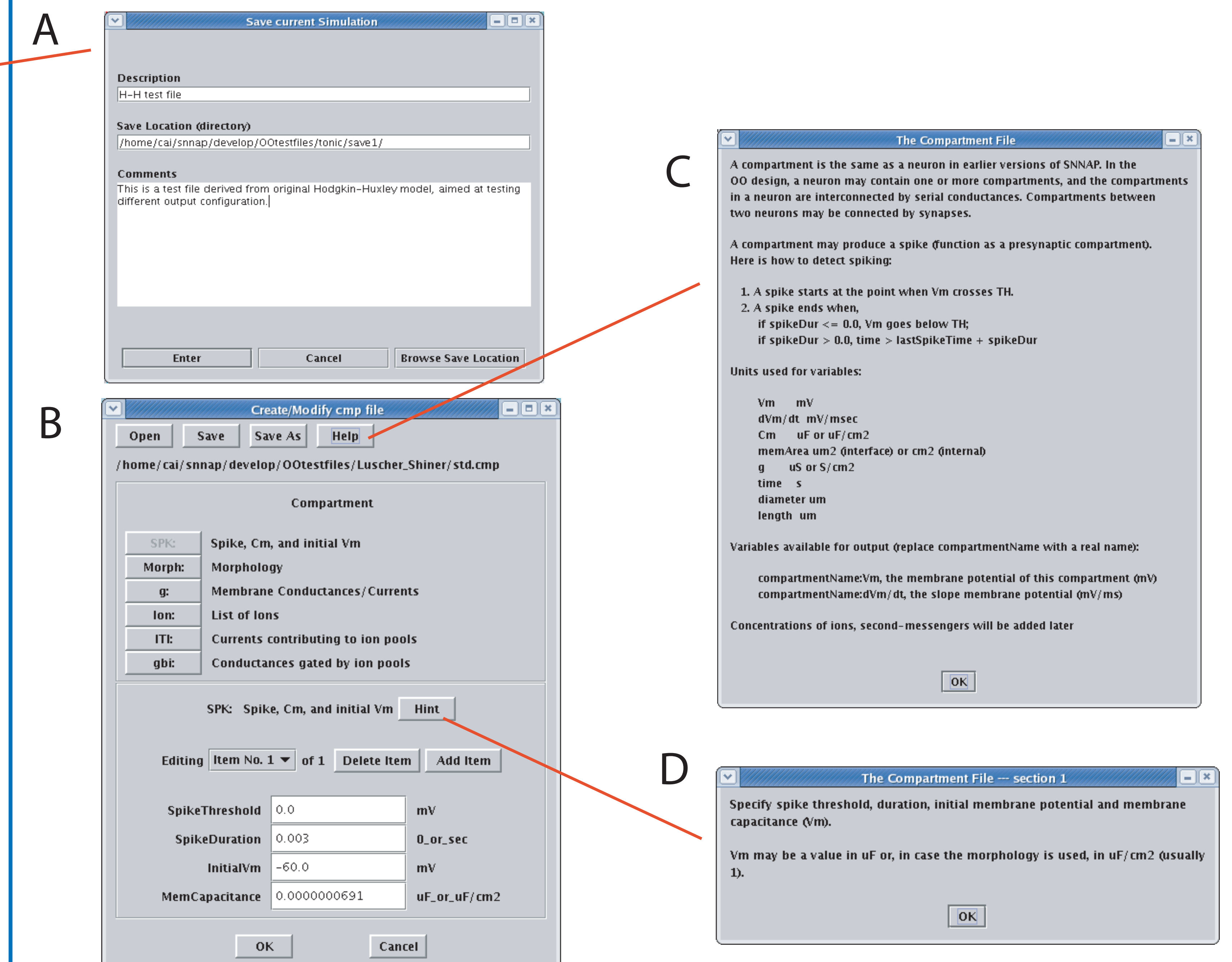
It's simple. SNNAP is available free of charge. Just visit the SNNAP web site and download the files.

<http://snnap.uth.tmc.edu/>

This work was supported by NIH grants R01 RR 11626 and P01 NS 38310.

4. Convenient tools for users

A. Ability to save a simulation set with your comments. No more worry about work being lost. B. Tool for building a model. C and D. Individually tailored help files are every where.



5. Better documentation for developers

The class files are organized into packages.

The java doc provides useful information about each class, design ideas, history of changes, variables, field summary, constructor summary and details.

Inside the source files, there are comments explaining why a particular implementation was chosen.

