North-bound interfaces

These features have resulted in GENESIS being one of the few simulators used for sets. It has been used for post-run analysis of this model using files generated by signed for the simultaneous display and analysis of a large number of dynamic data (Figure 2). Davis (Data Viewing System), is a general-purpose data viewer de displays, as in the visualization of a spreading wave of activation generated with Similarly, scripting with GENESIS GUI libraries (XODUS), allows easy creation of lines of script are needed to specify a sophisticated simulation. This approach

The scripting language and the modules are powerful enough that often only a few as well as routines that maximize speed with faster, more stable numerical

solution of the systems of differential equations that describe dendritic trees,

5) Graphical objects used for creating custom displays, and

Function Bindings

MOOSE (e.g. Neurospaces, GENESIS Namespace, findsolve)

1) Modules can be run separately on different machines. For example, the GUI
model environment, while the simulator can run elsewhere either serially or in parallel on more powerful machines.

Consequently, G3 no longer includes parsers, script interpreters, run time schedul

 Reduction of complexity for new module development.

A. Graphical User Interface

G3 is being developed in a way that allows different simulation systems to use its features. GENESIS, MOOSE, Neurospaces, NeuroML representa

The majority of simulators including GENESIS 1 and 2 are C/C++ programs. G3 is

The GENESIS 3.0 Project

with an active users group (BABEL), have also made it easy for individuals to learn

independently. This has important advantages as it facilitates the interoperability of software obtained from different sources by:

1) Reduced complexity of software modules compared to a unitary system.

2) A kinetics library and its GUI Kinetikit for modeling biochemical reactions.

3) Backward compatibility with G2. SLI is compatible. Currently refining simulation include:

- MOOSE reimplements the core GENESIS simulator code with much faster and

- Reduced complexity for new module development.

- Reduced complexity for new module development.

The adoption of the CBI paradigm in the development of G3 has been prompted by

3) Backward compatibility with G2. SLI is compatible. Currently refining simulation include:

- MOOSE reimplements the core GENESIS simulator code with much faster and

- Reduced complexity for new module development.

The majority of simulators including GENESIS 1 and 2 are C/C++ programs. G3 is

The GENESIS 3.0 Project

A universal graphical user interface and database for collaboration, education, and research in computational neuroscience.

Background

The General Neural Simulation Platform (GENESIS), and its parallel version (GENESIS) was the first software system in computational biology to encourage models to develop and share model features and components. From the outset, it was developed to support the biologically realistic simulation of new systems, ranging from subcellular components and biochemical reactions to complex models of single neurons, simulations of large networks, and systems-level models. Since its release for general use in 1998, GENESIS has provided one of the foundations for the ongoing course in Methods in Computational Neuroscience at Woods Hole MA, as well as courses offered in the United States, Mexico, Brazil, India, and in more than 50 universities around the world, where it has been used both as an instructional tool in research courses, and as a simulation based tool for neurobiological education in general. The most recent release of GENESIS, version 2.3, became available in March 2006, and runs under most UNIX-based systems with the X Window System, including Linux, OS/X and Windows with Cygwin.

The release of the free internet edition of The Book of GENESIS (Bower & Beeman, 2003) and the browser-based self-paced GENESIS Modeling Tutorials (Beeman, 2005), which may be downloaded from the GENESIS web site, coupled with an active users group (BABEL), have also made it easy for individuals to learn GENESIS modeling without the necessity of attending courses or workshops. This substantial support for the use of GENESIS has also provided the base for an extensive and growing user community in biological simulation. This evidenced by the rate of growth in the number of peer reviewed scientific papers using GENESIS outside of the Bower laboratory (Figure 1).

GENESIS 1 and 2

Most current neuronal simulators are typically non-scalable in design, the result of development by a single person of a protected property. In contrast, the architec

2) A kinetics library and its GUI Kinetikit for modeling biochemical reactions.

3) Objects for constructing intracellular ion concentrations from channel expressions, for modeling the spatial distributions of ion concentration pools, ionic pumps, and buffers.

4) Devices for ports and channels (e.g. pulse generators, spike train generators, voltage clamp circuitry) and analysis (e.g. spike train analysis

5) Graphical objects used for creating custom displays, and

6) The html viewer with elements and functions for the efficient implicit simulation of systems of differential equations that describe dendritic trees, as well as routines that maximize speed with faster, more stable numerical

The scripting language and the modules are powerful enough that often only a few lines of script are needed to specify a sophisticated simulation. This approach allows one to create a new GENESIS simulation by modifying one of the many exa

and use for simulation are those provided with GENESIS, the Modeling To

The GENESIS 3.0 Project

With the growing interest and involvement of both neurobiologists and technolo

C. Neurospaces Project

Compliant with the CBI architecture, the Neurospaces project embodies several completely modular software components implemented in a state-space theoretical framework (Figure 6). It supports a global namespace that is separated from solving instances, thereby separating and allowing for optimization of the numerical-core independently of the modeling package. Neurospaces supports GENESIS files, NeuroML, and .one cell morphology files, and exports NeuroML files. Currently, the following G3 compatible components have been implementing:

1) Hecco: A fast computational module based on the GENESIS kernel that can be instantiated from C, Perl, or other scripting languages.

2) Neurospaces: Modular web-based with biological entities and end-user concepts instead of mathematical equations.

3) Simple Scheduler in Perl: SSC can be connected to build Neurospaces and Hecco incorporate them correctly, such that they work together on a single simulation.

Other modules in development include:

1) Handoff: A fast computational module based on the GENESIS kernel that can be instantiated from C, Perl, or other scripting languages.

2) Neurospaces: Modular web-based with biological entities and end-user concepts instead of mathematical equations.

3) Simple Scheduler in Perl: SSC can be connected to build Neurospaces and Hecco incorporate them correctly, such that they work together on a single simulation.

Other modules in development include:

1) Handoff: A fast computational module based on the GENESIS kernel that can be instantiated from C, Perl, or other scripting languages.

2) Neurospaces: Modular web-based with biological entities and end-user concepts instead of mathematical equations.