ELSEVIER

Neurocomputing 38–40 (2001) 1657–1661

NEUROCOMPUTING

www.elsevier.com/locate/neucom

NEOSIM: Portable large-scale plug and play modelling☆

N. Goddard[a,b], G. Hood[b,*], F. Howell[a], M. Hines[c], E. De Schutter[d]

[a]*Institute for Adaptive and Neural Computation, Division of Informatics, University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL, Scotland, UK*

[b]*Pittsburgh Supercomputing Center, Mellon Institute Building, 4400 Fifth Ave., Pittsburgh, PA 15213, USA*

[c]*Departments of Computer Science and Neurology, Yale University, USA*

[d]*Theoretical Neurobiology, Born-Bunge Foundation, University of Antwerp, Universteitsplein 1, 2610 Wilrijk, Belgium*

Abstract

NEOSIM is a new simulation framework addressed at building large scale and detailed models of the nervous system. Its essence is a set of interfaces and protocols that enable a plug and play architecture for incorporating existing simulation modules such as NEURON [4] and GENESIS [1] as well as future visualisation and data analysis modules. From the start it has been designed to exploit parallel and distributed computers to reduce simulation run times to manageable levels, without the additional modelling effort required for earlier publicly-available parallel simulation tools. In this paper, we present the design of the NEOSIM framework, and discuss its applicability to a range of modelling studies. © 2001 Published by Elsevier Science B.V.

Keywords: Large scale modelling; NEURON; GENESIS

The high degree of connectivity and complexity of neuronal systems means that modelling even small systems requires a large number of neurons and much larger numbers of synapses. Incorporating biophysical detail into the models of neurons and synapses tests the limits of workstation processors and memory resources [6].

☆Supported by: NIH (MH-57358), NSF and the Flemish (FWO G.0113.96W) and Belgian Governments (IUAPP4/22). EDS is supported by the FWO-Vl.

* Corresponding author. Tel.: + 1-412-268-7063; fax: + 1-412-268-8200.

E-mail address: ghood@psc.edu (G. Hood).

0925-2312/01/$ - see front matter © 2001 Published by Elsevier Science B.V.
PII: S0925-2312(01)00528-8

The availability of networked workstations and parallel supercomputers enables, in principle, the construction of models of several orders of magnitude greater size or detail. Previous neuronal modelling environments have exploited this opportunity [3,5], but on the whole have not shielded the modeler from the complexities of parallel programming, resulting in a significant cost in model development time and difficulty. For example, the modeller may be required to write a parallel program in the scripting language to control distribution of the model across the parallel machine, or to control the simulation run, which introduces additional implementation-dependent conceptual difficulty entirely unrelated to the neuroscientific modelling process. The NEOSIM project targets efficient yet transparent use of parallel computers: modelers can focus on modelling issues and ignore the underlying computational device.

Two decades' experience of the construction and use of neuronal modelling environments, and modern software-engineering techniques, have demonstrated the need for a component-based architecture which enables distributed development of both the modelling environment and scientific models. This is the NEOSIM approach, both in the execution of the project and the design of the software.

1. NEOSIM—what?

At the heart of NEOSIM is a kernel which manages simulation time and the distribution of events between "entities". These entities perform the numerical simulation of neurons. Example entities include the numerical cores of the NEURON and GENESIS simulation packages, CVODE and HSOLVE, but the architecture is extendable to allow user components (written in C + + or Java) to be added later (Fig. 1). Thus single neuron models constructed with these packages can be wired up

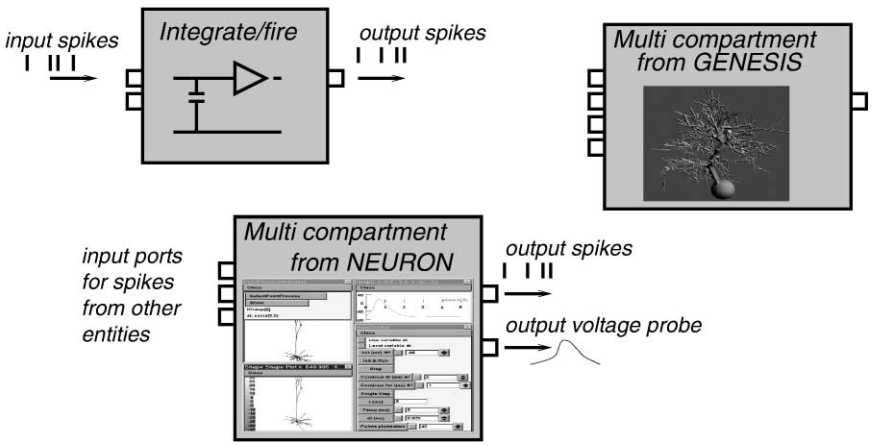

Fig. 1. Single cell models can be imported from NEURON and GENESIS, and communicate using events.

into higher level network models using the NEOSIM connection commands. A complementary key aspect of the NEOSIM framework is a set of interface and protocol specifications which enable domain-specific components to interact effectively. For example, one interface specifies a generic form for descriptions of branching and compartmentalising structure of compartmental models, and another for the communication of ion-channel descriptions and simulations. Components adhering to these interfaces and protocols can be combined so that, for example, a new ion-channel specification component can be substituted for the existing one without any other code reorganisation. Essential supporting software technologies such as version control, simulation repeatability, and database interfaces are a part of the framework.

2. NEOSIM—how?

NEOSIM entities communicate exclusively by sending and receiving timestamped events. The typical event in a spiking neural network simulation is a single spike. NEOSIM can take advantage of the axonal delay between spike generation and delivery to loosen the synchronisation constraint between sender and receiver; the simulation engine can let sending and receiving entities be at different simulation times at a given real time. This allows the use of parallel discrete event simulation techniques [2]. The sender and receiver can be run on different processors without the communications burden of keeping them in exact lockstep. Further optimisations are possible if a variable timestep integration method (as in CVODE) is used for individual neurons, as they will be updated less often when their state is changing slowly.

The event infrastructure in NEOSIM is also used to handle more complex events including probes for voltage and ion concentrations, and in general any user supplied "object", including for example an entire cell morphology which might be used by a visualisation entity. It is hard to predict the future requirements of modelling; as with the facilities for plug-in simulation components, new events can be added without requiring kernel or other component modifications.

3. NEOSIM—future network descriptions

The GENESIS environment includes some powerful commands for building and connecting networks of neurons ("createmap" and "volumeconnect") [1]. However, extending these commands to fit a particular modelling task requires modifying the source code. NEOSIM uses object-oriented techniques to support construction of generalised populations of cells, and user-defined projections from one population to another (Fig. 2). It is particularly important for large-scale networks that projections are handled efficiently, as the time to set up connections can grow at an alarming rate with the number of neurons; $O(N^2)$ algorithms are common.

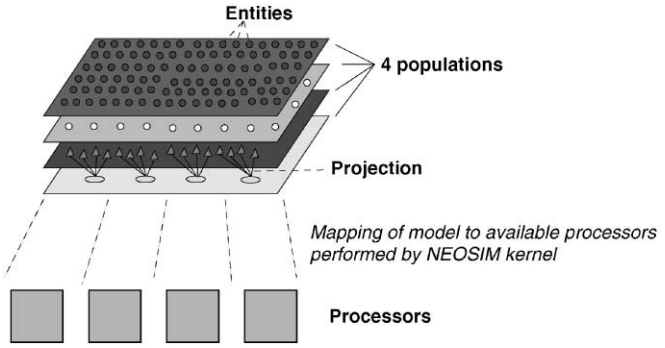

Fig. 2. Networks are specificed at a high level in terms of populations and projections.

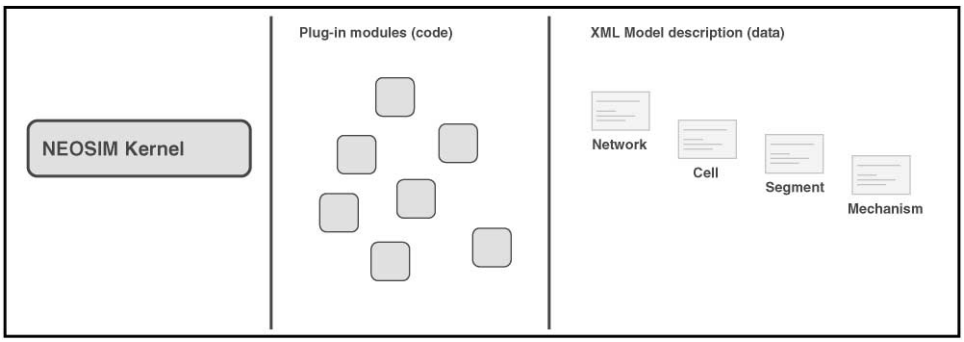

Fig. 3. A NEOSIM model is composed of plug-in modules and XML description files for each level of a simulation model.

4. NEOSIM—computer technologies

As a framework for supporting heterogenous simulation environments NEOSIM has to support different computer languages. It currently uses C and C + + (for the simulation modules) and Java (for visualisation modules and the portable kernel), with the interface between languages provided by the Java Native Interface. The plug-in modules facility (Fig. 3) is provided by Java archive (Jar) files and shared object libraries for native code. Independence from parallel-processor specifics is achieved via parallel programming software layers including MPI, RMI and OpenMP.

5. Results

We have a working NEOSIM kernel implemented using Java and running on networks of workstations, using remote method invocation as the object-oriented

transport layer. The portability of Java means that the kernel can use a mixture of Linux, NT, Alpha, Solaris and SGI workstations at the same time.

Numerical integration components and model specification components derived from NEURON have been integrated to allow single cell models specified using the NEURON script language (hoc and NMODL) to be run under the NEOSIM kernel.

The high speed solver used by the GENESIS simulation language (HSOLVE) has also been adapted to have an event-based interface, and a large-scale Purkinje model has been run as a NEOSIM entity. This uses the compiled cell specification format specific to GENESIS.

Visualisation of network activity can be very useful for obtaining insight into network behaviour. We have developed prototype visualisation tools using Java3D which connect to the event streams generated by the neurons and display 3D network structure and dynamic behaviour.

6. Discussion

The aim of the NEOSIM project is to produce and distribute software tools of use to the computational neuroscience community, particularly focussed on supporting large-scale network modelling. The approach is particularly suitable for networks with known axonal delays, and the parallel discrete event simulation approach works well with variable timestep integration methods. The approach is less suited to situations with very tightly coupled entities (for example those using gap junctions, or a model of a single cell, or a rate-coded connectionist network) as these require frequent, often lockstep, communication. However, these types of simulations can be incorporated as single entities within a larger model so that, for example, a connectionist network, or an electrotonically-connected network, could be a part of a larger simulation model. Currently, work is focussed on stress-testing NEOSIM with a number of large scale modelling studies prior to releasing the software. These include models of hippocampus [7], the cerebellar cortex [6], the basal ganglia and the thalamocortical loop.

References

[1] J.M. Bower, D. Beeman (Eds.), The Book of GENESIS: Exploring Realistic Neural Models with the General Neural Simulation System, 2nd Edition, Springer, New York, 1998.
[2] R. Fujimoto, Parallel discrete event simulation, Commun. ACM 33 (1990) 10.
[3] N. Goddard, G. Hood, Parallel genesis for large scale modeling, in: J.M. Bower (Ed.), Computational Neuroscience: Trends in Research 1997, Plenum Publishing, NY, 1997, pp. 911–917.
[4] M.L. Hines, N.T. Carnevale, The NEURON simulation environment, Neural Comput. 9 (1997) 1179–1209.
[5] P. Hammarlund, O. Ekeberg, Large neural network simulations on multiple hardware platforms, J. Comput. Neurosci. 5 (4) (1998) 443–459.
[6] F. Howell, J. Dyhrfjeld-Johnsen, R. Maex, E. De Schutter, N.H. Goddard, A large scale model of the cerebellar cortex using PGENESIS, Neurocomputing 32–33 (2000) 1041–1046.
[7] W.B. Levy, A sequence predicting CA3 is a flexible associator that learns and uses context to solve hippocampal-like tasks, Hippocampus 6 (1996) 579–590.