NEUROCOMPUTING



Neurocomputing 44-46 (2002) 1077-1081

www.elsevier.com/locate/neucom

# NeuroML for plug and play neuronal modeling $\stackrel{\text{tr}}{\sim}$

N.H. Goddard<sup>a,\*</sup>, D. Beeman<sup>b</sup>, R. Cannon<sup>a</sup>, H. Cornelis<sup>c</sup>,
M.-O. Gewaltig<sup>d</sup>, G. Hood<sup>e</sup>, F. Howell<sup>a</sup>, P. Rogister<sup>a</sup>,
E. De Schutter<sup>a</sup>, K. Shankar<sup>b</sup>, M. Hucka<sup>b</sup>

<sup>a</sup>Institute for Adaptive and Neural Computation, Division of Informatics, University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL, Scotland, UK <sup>b</sup>California Institute of Technology, USA <sup>c</sup>Borne-Bunge Foundation, Antwerp, Belgium <sup>d</sup>Honda Research and Development, Germany <sup>e</sup>Pittsburgh Supercomputing Center, Carnegie Mellon University, Pittsburgh, USA

### Abstract

Modern software systems for simulation, database access, visualisation and data analysis, supporting distributed, extensible, evolutionary development, are designed around a small core that loads plug-in components. We have designed such a system for the neurosciences using an XML-based protocol, NeuroML, to exchange information between components. NeuroML supports high-level descriptions of data, models, references, and other types of information. We have built simulation kernel plug-ins, visualisation plug-ins, and model-description GUI plug-ins which interoperate in this framework. We describe the current status of these plug-ins and our future plans for further plug-in components. (c) 2002 Published by Elsevier Science B.V.

Keywords: Collaborative tools; Simulation; Database; NeuroML

# 1. Introduction

We are interested in methodologies which will improve the ability of neuroscientists to collaborate in exchange and development of data and models. There are two key

0925-2312/02/\$-see front matter © 2002 Published by Elsevier Science B.V. PII: S0925-2312(02)00420-4

 $<sup>\</sup>stackrel{\text{\tiny th}}{\sim}$  Supported by: NIH (MH-57358).

<sup>\*</sup> Corresponding author.

E-mail addresses: nigel.goddard@ed.ac.uk (N.H. Goddard), dbeeman@dogstar.colorado.edu

<sup>(</sup>D. Beeman), robert@anc.ed.ac.uk (R. Cannon), hugo@bbf.uia.ac.be (H. Cornelis), marc-oliver. gewaltig@ipm.net (M.-O. Gewaltig), ghood@psc.edu (G. Hood), fwh@anc.ed.ac.uk (F. Howell), paulro@anc.ed.ac.uk (P. Rogister), erik@bbf.uia.ac.be (E.D. Schutter), kshankar@bbb.caltech.edu (K. Shankar), hucka@caltech.edu (M. Hucka).

technical requirements for supporting this kind of collaborative work. First, there must be computational technologies to enable collaboration at the level of exchange of software components, and we have been developing such methods over the last few years [6,9]. Second, it is crucial for neuroscientists to have access to tools which support discussion, development and exchange of data, methods, and models, including databases of models [1]. Recently, the methodological bases for these informational tools have been developed. In this paper, we introduce the software architecture and refine NeuroML, the XML-based protocol we have previously introduced [7]. We have built simulation kernel plug-ins, visualisation plug-ins, and model-description GUI plug-ins which interoperate in this framework. We describe the current status of these plug-ins and our future plans for further plug-in components.

#### 2. NeuroML

Software systems for simulation, database access, visualisation and data analysis in neuroscience need to be developed by distributed teams in an extensible and evolutionary manner. Modern software systems with these needs are designed around a small core that loads plug-in components. Integration between the components is provided in part by protocols for exchanging information. NEOSIM [6] is such a plug-in framework for simulation components that implement single-cell and ion-channel processes. The Modeler's Workspace [9] is a complementary plug-in framework for user-level components such as databases, GUIS, and simulators. The Modeler's Workspace uses NeuroML as the information exchange protocol, while NEOSIM uses NeuroML's model description features as the (machine-level) descriptive language mediating between model-specification components (e.g., a GUI) and computational simulation components (e.g., a solver). Fig. 1 illustrates the architecture.

In Goddard et al. [7] we proposed a set of conceptual schemas for describing models of neural systems, as an initial step towards NeuroML. We proposed that model descriptions be given in a declarative form, so that models can be described by database schemas, and simulated by encapsulated code modules. Models composed of well understood and accepted components can be described merely by naming these



Fig. 1. Module interfaces allow tools to interact without being tightly coupled.



Fig. 2. A standard model description format allows exchange of models between simulators.

components. Database schemas can be created corresponding to these named components. Simulation code modules which create and execute these components can be written and invoked using the components name. We identified three key requirements for these model-description components: clarity, portability and modularity, and introduced a representational framework supporting these requirements. The representational framework is based loosely on object-oriented programming concepts, and the particular choice of first-level classes was inspired by the work of Gardner et al. [5].

This representational framework gives rise to NeuroML as an XML-based language for describing concepts used in neuroscience. XML, the Extensible Markup Language [2] is a language used to express self-describing, semi-structured representations of information. We use XML Schemas [10] to describe model templates, and actual models are encoded in NeuroML using these Schemas. The XML Schemas ensure that model descriptions in NeuroML adhere to the structural requirements of the model templates and can support other software tools such as visual editors. Extensibility is provided by allowing software developers and users to define new templates which refer to dynamically loaded code modules that implement the new functionality. The NeuroML website, www.neuroml.org, contains current information about the models, code modules, and extensions to base NeuroML, which are available. It also contains toolkits to aid developers and a facility for distributed sharing of models and modules along the lines of Napster. Fig. 2 illustrates the value of this approach.

## 3. NEOSIM

NEOSIM is a framework to support large-scale multi-level modelling of the nervous system. It:

- (1) Allows plug-in of different simulation components for simulation, visualisation and I/O.
- (2) Is specified in a way which allows running on workstations as well as networks of workstations and parallel machines without having to write a parallel program.
- (3) Is based on discrete event simulation.

The NEOSIM kernel provides just the basic support necessary for building and running large simulation models. All the interesting behaviour of model components is provided by plug-in modules. These modules are intended to be developed independently by different groups, and can communicate with each other using the kernel interface. For example, a model could be built using a visualisation component written in one lab, a neuron simulation component from the NEURON simulator, another from the GENESIS simulator and a number of home grown Java or C++ components. When (and if) the time comes to scale up a model to more realistic sizes, the NEOSIM kernel will support the distribution of the model onto multiprocessors, networked workstations or parallel machines without having to modify the model description.

Plug-in components currently under development supporting NeuroML as a descriptive language and the NEOSIM kernel for simulation runtime include:

- (1) Single-cell solution engines derived from Catacomb [4] and NEURON [8] and GENESIS [3].
- (2) Single-cell model description GUIs from Catacomb and NEURON.
- (3) Channel mechanism description GUI from Catacomb.
- (4) Network structuring GUIs.
- (5) Network-level structure and activity visualisation components.

For current information on the facilities available in NEOSIM, see the website www.neosim.org.

## 4. Conclusion

The work described here has been motivated by the increasingly large-scale needs in neuroscience for handling data and models. Structuring the exploding volume of experimental data according to the models which use it is likely to lead greater synergy between modelling and experimental efforts. NeuroML is a markup language that can support, amongst other things, database information structure and simulation model execution. The inclusion of an ability to interface imperitive code modules to the declarative templates via a well-defined, parameterised interface, provides a path for speculative research-oriented structures to be supported and, crucially, to gradually migrate into the standard model description language. For both database entries and simulation execution, the model description capabilities are naturally extensible to support evolution of conceptualisation of brain function. Our major aim in this work is to facilitate collaborative modelling work which requires the ability to share and reuse model components developed at other times in other labs on other computational systems. We are

1080

well on the way to supporting these requirements, using the NeuroML-based tools for description and the NEOSIM framework for execution.

## References

- D. Beeman, J.M. Bower, E. De Schutter, E.N. Efthimiadis, N. Goddard, J. Leigh, The GENESIS simulator-based neuronal database, in: S. Koslow, M. Huerta (Eds.), Progress in Neuroinformatics, Lawrence Earlbau, London, 1996.
- [2] J. Bosak, T. Bray, XML and the second-generation web, Scientific American, May, 1999. Also available via the world wide web at http://www.sciam.com/1999/0599issue/0599bosak.html.
- [3] J.M. Bower, D. Beeman, The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System, 2nd Edition, Springer, New York, 1998.
- [4] R. Cannon, Catacomb—an interactive model description and computation package, this volume, 2002.
- [5] D. Gardner, M. Abato, K.H. Knuth, R. DeBellis, D.M. Erde, Dynamic publication model for neurophysiology databases, Philos. Trans. Roy. Soc. 356 (2001) 1229–1247.
- [6] N. Goddard, G. Hood, F. Howell, M. Hines, E. De Schutter, NEOSIM: Portable large-scale plug and play modelling, Computational Neuroscience: Trends in Research, Elsevier, Amsterdam, 2001.
- [7] N.H. Goddard, M. Hucka, F. Howell, H. Cornelis, K. Shankar, D. Beeman, Towards NeuroML: model description methods for collaborative modeling in neuroscience, Philos. Trans. Roy. Soc. 356 (2001) 1209–1228.
- [8] M.L. Hines, N.T. Carnevale, The NEURON simulation environment, Neural Comput. 9 (1997) 1179– 1209.
- [9] M. Hucka, K. Shankar, S. Emardson, The Modeler's Workspace Database Organization and Interaction, Division of Biology, The California Institute of Technology, Pasadena, California, U.S.A., 2000, unpublished manuscript.
- [10] H.S. Thompson, D. Beech, M. Maloney, N. Mendelsohn, XML Schema Part 1: Structures (W3C Working Draft 7 April 2000). Available via the World Wide Web at http://www.w3.org/TR/xmlschema-1/.