



# NeuroSpaces: separating modeling and simulation

Hugo Cornelis\*, Erik De Schutter

*University of Antwerp, Universiteitsplein 1, Antwerp 2601, Belgium*

---

## Abstract

Modeling starts with sets of mathematical equations that have been put together by people who have a good intuition of the concepts behind the biological reality. A software system for neuronal modeling thus must manage sets of equations as well as the biological concepts.

A rigorous theoretical analysis on the software requirements for such a system has led us to a new fundamental data model. Based on this data model we have implemented an extensible software system that combines mathematical and neurobiological modeling and that still reflects concepts like heterogenous components and hierarchical networks. When connected to a computation engine, the system turns into a simulator that is efficient in memory requirements, easy to use and allows complete inspection of the mathematical model.

© 2002 Elsevier Science B.V. All rights reserved.

*Keywords:* Modeling; Analysis; Simulator

---

## 1. Introduction

During the last twenty years traditional monolithic software design has gradually been abandoned in favor of architectures with a small extensible core or kernel. The kernel manages a set of modules and components that engage in shared activities via semantically defined interfaces [6,10]. Such systems are called collaborative systems. The current neuronal simulation packages—developed during the last decade—don't share this philosophy. It makes them hard to extend and difficult to use for large and complex neuronal models.

Recent developments try to overcome this situation in two ways: first there have been attempts to integrate the existing packages into a collaborative system [7]. Although these attempts could hold a promise, they have been unsuccessful so far because the

---

\* Corresponding author.

*E-mail address:* [hugo@bbf.uia.ac.be](mailto:hugo@bbf.uia.ac.be) (H. Cornelis).

simulation package itself still suffers from its monolithic design and is difficult to interface with other software components. Second, attempts have been made to make calculation libraries accessible as object-oriented class hierarchies [14]. These attempts have no direct relation with neuronal modeling with the consequence that few people in the neuroscience community are aware of the project. The classes are sometimes ill defined or do not address a number of important issues related to neuronal modeling.

## 2. Problem statement

After analyzing the structural software requirements for a neural simulation package, we discovered that there is currently no data model available that covers all the modeler's needs. An example is the need to compose hierarchical networks. In Genesis such a network is composed with scripts that construct all the populations and then add links between individual cells. However, after the setup phase of the model, the topology of the network as well as the projections between the individual populations are lost i.e. the model has become a flat network instead of being a hierarchical network—the modeler's concept of the model. Put in other words, projections reside at different levels of the network hierarchy, yet the requirement to inspect *all* the connections on one cell must not be violated, even if the cell is involved in many projections. Currently no single data model is able to fulfill both of these two requirements. Other examples of problematic needs are scaling of conductances and compound models of heterogenous intracellular mechanisms. Reducing these problematic needs we ended with a set of ultimate abstract requirements.

## 3. Description of the work

Building further on known theoretical data structures that have been studied in informatics [12,13], we designed a new data model that is able to address all the abstract requirements (see Fig. 1 for an overview). The theoretical basis of this new data model is given by an ordered tree with relative paths to tree nodes encoded as integers and the possibility to have links between the ancestors of a node [1]. By assigning mathematical or biological semantics to the nodes of the tree, it becomes a natural representation of a detailed neuronal model of a cell or a network. The integer encoding scheme allows grouping of components at any biological level and memory efficiency via prototyping of repetitive sub-models. Trivial manipulations of the integers associated with such a relative path allow a flat representation of components of the same type such that e.g. projections in a hierarchical network can be inspected via a single connection matrix and connections converging on the same compartment can be queried.

The implementation of this data model for neuronal modeling is called NeuroSpaces. It consists of a number of independent software components that cover different needs of extensibility (Fig. 2). An example is the loadable module manager that supports the implementation of extensions covering specialized modeling semantics, such as setting

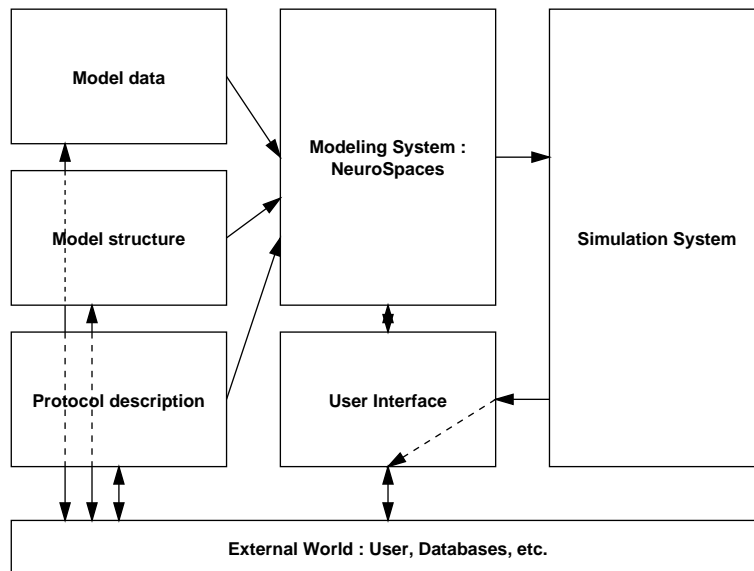


Fig. 1. Context of the design of NeuroSpaces: At the left are descriptions of the system to be simulated: the model data and structure and the experimental protocol if applicable. All these things are specified in a declarative way such that they can go into or come from a database. At the right is the simulation engine, which could be any of the currently available packages. For reasons of memory efficiency we selected the numerical solver of Genesis to do our testing. In the center, a software component—called NeuroSpaces—that only supports the setup and validation of modeling. In use it is quite close to a hierarchical database. The split between modeling and simulation allows separate specialization of software components. During model construction, the user only communicates with NeuroSpaces, during the simulation, all results are handed to the user via the user interface, such that the enormous complexity of the simulation engine remains hidden.

up three-dimensional projection schemes or randomizing parameters. Another part of the system is a file parser that gives explicit support for model reuse and grouping into libraries.

Except for computer science related research NeuroSpaces also includes a number of benefits for neuronal modelers such as memory efficiency and complete model validation. A major design principle has been the separation of the specification of the mathematical model and the way of simulating this model as done by computation engines. Computation engines can now be implemented in the most efficient way while their complexity is hidden from the user by NeuroSpaces [3].

NeuroSpaces has been linked with Genesis and its highly optimized numerical computation engine (hsolve [4,3]) and can be used to build models and simulate their behavior. Using subcellular components like dendritic segments and synaptic channels as building blocks, we constructed libraries of detailed models of the different cell types found in the cerebellar cortex. These libraries have in turn been used to build networks with the same topology as the ones currently used to investigate the function of the cerebellar cortex [5,2,11]. Since these libraries cover the complete spectrum from the subcellular up to the network level, the memory requirements of complex models and

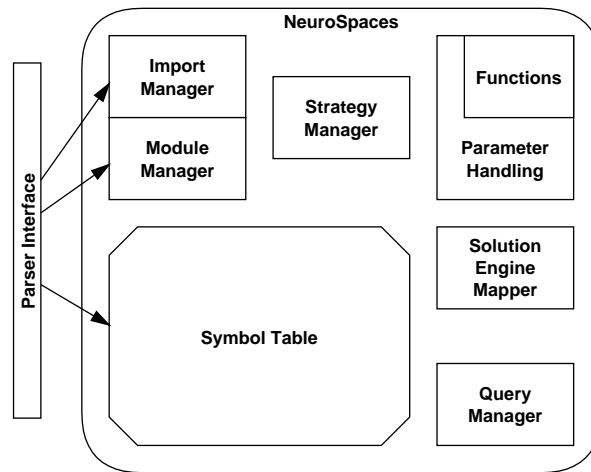


Fig. 2. The internal design of NeuroSpaces: descriptive files go through a parser that interoperates with a loadable module manager and an import manager. Then the model is stored in a compact symbol table, but using a specialized data structure it gives the illusion of a hierarchical database that contains all mathematical and algorithmic details of the model. A special module for parameter handling implements scaling of conductances and relative coordinates. The solution engine mapper allows to instantiate specialized computation engines, specific for certain model structures.

the setup time of computation engines can easily be evaluated. The structure of the network libraries can be used as a basis to build other network types.

#### 4. Future work

After having tested this software system further our future plans are twofold: first we will be working on database integration. The libraries we are currently using are self-contained and declarative. This makes interaction possible with XML, the general agreed upon database exchange format [8]. We are planning to implement additional software to setup simulations directly from databases.

Second we are investigating how to distribute simulations on parallel machines [9]. Since the mathematical model is completely available for inspection, it can be used to calculate a workload for transparent parallelization of large simulations.

#### References

- [1] H. Cornelis, E. DeSchutter, Treespaces, Tech. Rep., UIA, University of Antwerp, Universiteitsplein 1, Wilrijk, Antwerp, December 2001.
- [2] H. Cornelis, E. DeSchutter, The purkinje cell tutorial, Available from the Genesis source code distribution. <http://www.genesis-sim.org>, May 2002.

- [3] H. Cornelis, E. DeSchutter, Tutorial: Simulations with genesis using hsolve. Internal Document, November 2002 (This will probably be included soon in the Genesis source code distribution. <http://www.genesis-sim.org>).
- [4] E. DeSchutter, D. Beeman, in: the Book of Genesis, 1st Edition, Springer, Berlin, 1988 (Speeding up large simulations, Chapter 22).
- [5] E. DeSchutter, J. Bower, Simulated responses of cerebellar purkinje cells are independent of the dendritic location of granule cell synaptic inputs, *Proc. Natl. Acad. Sci. USA* 91 (1994) 4736–4740.
- [6] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns, Elements of Reusable Object-Oriented Software*, Professional computing series. Addison Wesley, Reading, MA, 1995.
- [7] N. Goddard, G. Hood, F. Howell, M. Hines, E. DeSchutter, Neosim: portable plug and play neuronal modelling, *Neurocomputing* 38–40 (2001) 1657–1661.
- [8] N.H. Goddard, M. Hucka, F. Howell, H. Cornelis, K. Shankar, D. Beeman, Towards neuroml: Model description methods for collaborative modelling in neuroscience, *Philos. Trans. Roy. Soc. Ser. B: Biological Sciences* 356 (2001) 1–20 (Theme Issue organized and edited by Rolf Kötter on “Neuroscience databases—tools for exploring brain structure-function relationship”).
- [9] F. Howell, J. Dyhrfeld-Johnsen, R. Maex, N. Goddard, E. DeSchutter, A large-scale network model of the cerebellar cortex using pgenesis, *Neurocomputing* 32 (2000) 1041–1046.
- [10] M. Hucka, A. Finney, H. Sauro, H. Bolouri, J. Doyle, K. Kitano, *Foundations of Systems Biology*, MIT Press, Cambridge, MA, 2001 (The ERATO Systems Biology Workbench: an Integrated Environment for Multiscale and Multitheoretic Simulations in Systems Biology, Chapter).
- [11] R. Maex, E. DeSchutter, Synchronization of golgi and granule cell firing in a detailed network model of the cerebellar granule cell layer, *J. Neurophys.* 80 (1998) 2521–2537.
- [12] R. Sedgewick, *Algorithms in C, Parts 1-4, Fundamentals, Data Structures, Sorting, Searching*, 3rd Edition, Addison-Wesley, Reading, MA, 1999.
- [13] R. Sedgewick, *Algorithms in C, Part 5, Graph Algorithms*, 3rd Edition, Addison-Wesley, Reading, MA, 2002.
- [14] J.-F. Vilbert, Xnbc v9, a user friendly simulation and analysis tool for neurobiologists, *Neurocomputing* 38–40 (2001) 1715–1723.