Neurospaces Parameter Handling

Hugo Cornelis and Erik De Schutter

October 19, 2003

Abstract

In neuro-biological modeling, no good software technologies currently exist that are memory efficient and give access to all the mathematical details of a large model. Here we present enabling technologies that deal with transformations of parameters of a large model. The discussed techniques allow to store models in a memory effective way.

1   Problem Statement

In neuro-biological modeling, a modeler needs access to all the mathematical details of a model to validate the correctness of the model. Mainly two categories of data models are being used to store a neuronal model in the main memory of a computer. On the one hand, the neuronal model can be stored in a hierarchical way (see figure 1). This is used by simulators like Genesis and has the advantage that all the details of the model are accessible as attributes of the individual model components, something that is crucial to validate complicated models[1, 2]. Nevertheless this data model is very memory expensive. Since complicated neuronal models tend to be large, this data model impedes the simulation of these complicated models. As a result, in P Genesis, the parallel version of Genesis intended for the simulation of very large and complicated neuronal models, the data that describes the model, is distributed over a parallel computer, which hampers again model validation, since not all the details of the neuronal model are accessible from one node of the parallel computer[6].
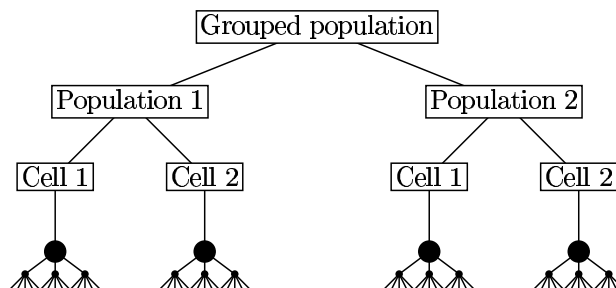

Figure 1: Storing a neuronal model in a hierarchical way. This model shows the memory layout to store 4 cells (and their morphological data) in two cell populations. All the parameters of the model are accessible, but it takes much memory to store complicated models.

On the other hand, to solve the above mentioned problems, a number of modeling packages use a reference based data model to store neuro-biological components in the main memory of a computer[3, 5]. The references allow to store repetitive components in a memory inexpensive manner, so the advantage of this data model is memory efficiency (see figure 2). The drawbacks are twofold : first, the repetitive components are shared by the references and thus always exactly the same, which, in the context of biological relevance, is not always desirable. Second and more importantly, it becomes impossible to distinguish between two axonal connections going to or coming from a shared subcomponent of two different references. As a consequence, projections between populations have to be specified with a descriptive component of the neuronal model, but the individual connections of a projection cannot be validated.

One of the main goals of our research when implementing the modeling package Neurospaces, has been to allow memory efficient storage of neuro-biological models, while keeping the validation requirement[4]. In reality this means that we want to give the modeler access to the hierarchical data model used by Genesis, but implement it behind the scene using references.
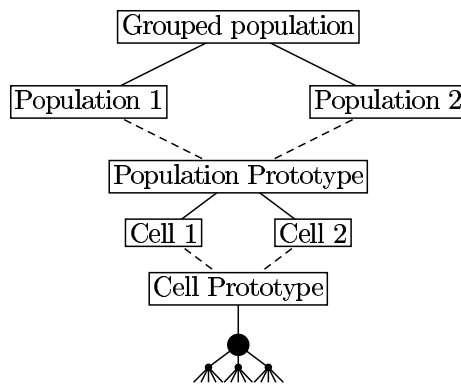
Figure 2: Storing a neuronal model using references. Again the model shows the memory layout to store 4 cells (and their morphological data) in two cell populations. Many parameters of the model are shared between the references, but it takes little memory to store large models.

2   Categories of Parameters

Model validation and inspection, is always done by accessing individual parameters. By categorizing parameter types of a neuronal model, it becomes possible to store them in an efficient way. In Neurospaces, parameters can have symbolic values, direct numerical values and transformed numerical values. We discuss here two categories of parameters in a neuro-biological model, known to Neurospaces:

1. Coordinates: The coordinate of a biological component always gets a direct default value of zero (unless overwritten by the modeler). The transformed coordinate is defined to be the direct coordinate of the component plus the transformed coordinate of the parent of the component.

   The advantage of this technique is that it allows not to specify coordinates for components that are at the same location as the parent component, e.g. the position of a channel in a segment defaults to the location of the segment. Furthermore when coordinates are attributed to references in a referenced based data model, the coordinates of different references will propagate to the shared components below the references, such that the shared components have different locations depending on the reference used to access them. Nevertheless the shared components are stored only once in the computer memory. Additionally, a reference can implement a transformation on the coordinates. This is illustrated in figure 3.

2. Conductances: Maximal channel conductances in neuroscience papers are often given as conductance densities, expressed in conductance per unit area. Neurospaces knows about the difference between conductance density and maximal conductance. The conductance density is a direct attribute of the channel, while the transformed value of the conductance density is the absolute maximal conductance of the channel. The transformed value is obtained by retrieving the surface of the parent biological component of the channel and multiplying the surface with the conductance density of the channel.

Neurospaces knows many more categories of parameters: segment capacitance, segment axial resistance, concentration time constant etc... A function that implements the parameter transformation is called a parameter handler. In the program code of Neurospaces, the parameter handlers are implemented at the level of the types of the biological components, e.g. the channel type uses a parameter handler that is responsible for scaling of conductance density to an absolute maximal conductance. Parameter handlers can be shared by biological types e.g. the coordinate handler is shared by all the types such that all the components in a model have a zero default coordinate (that can be overwritten).

3   Evaluation

The advantage of the above summarized techniques is that it allows user extensions to be written in a modular way, and it keeps the reference based data model invisible to a modeler (i.e. all the
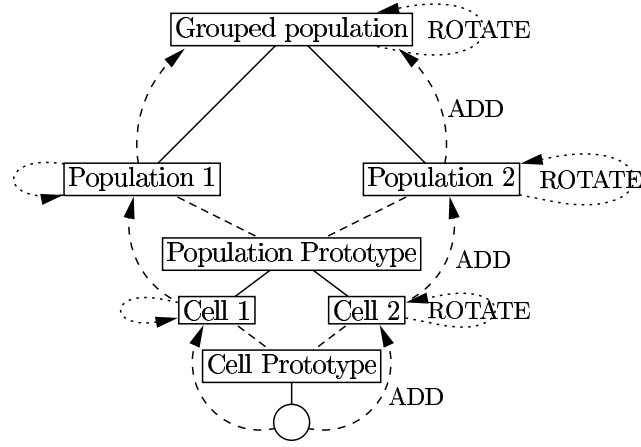
Figure 3: Calculation of the coordinate of the cell soma for two cells belonging to two different cell populations. The direct value of the coordinate of the soma is not set, so it defaults to zero. This value is added to the transformed coordinate of the cell. The transformed coordinate of the cell is defined to be the direct coordinate of the cell (the position of the cell in the population) plus the transformed coordinate of the population. This process continues upwards until the top level is reached. At each level additional geometrical transformations can be implemented, e.g. to do a rotation of a population, or to put a grid of cells in a folium like structure.

parameters are accessible like in a hierarchical data model). One of the main results is the memory efficiency of Neurospaces. Figure 4 shows the memory layout for a scaled-down version of a model of the cerebellar cortex assembled with Neurospaces[7].
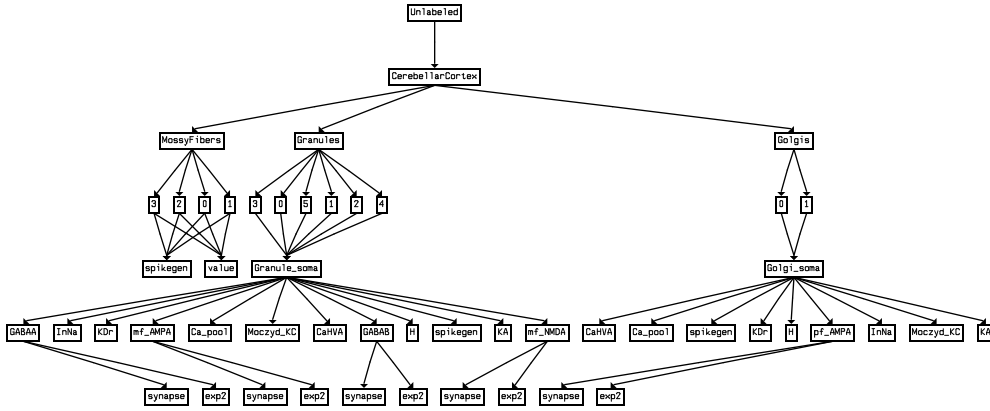

Figure 4: Run-time memory layout for a model of the cerebellar cortex modeled with Neurospaces.

The convergence of many arrows in the figure indicates that many nodes are shared by references which results in considerable memory savings. The evaluation of the memory performance of Neurospaces when storing large models is shown in figure 5.

The proposed techniques are simple and easy to implement. Neurospaces satisfies the requirement of model validation and at the same time consumes little memory for large neuronal models. The main disadvantage is the time needed to compute the transformed parameters (mainly the coordinates) of all the biological components in the neuronal model. The transformed coordinates are needed during model construction, e.g. when projections between the populations are computed based on the positions of the cells and the morphology trees.

4   Future

The main focus of future research will be to reduce the CPU overhead during model construction. Complementary design techniques will be analyzed to speed up the responsiveness of the software
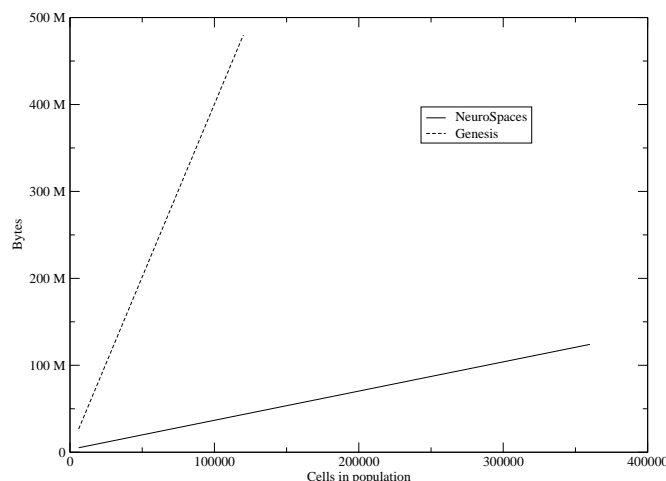
Figure 5: Comparing the memory use of Neurospaces for a model of a population of granule cells with the memory use for the same population modeled with Genesis. For a small population, Neurospaces uses a bit more memory than Genesis (not shown), but as soon as the model becomes big, the memory savings are huge. Given the derivatives of the two lines being approximately 1/3 and 4, the memory savings when using Neurospaces are about a factor 12.

system. For coordinates these techniques include the use of spatial indexing structures like R-trees and grid files. The main argument to implement such data structures is that they will reduce the time complexity of a spatial query from linear time to logarithmic time.

References

[1] *GENESIS Reference Manual*, May 2001.
http://www.genesis-sim.org.

[2] BOWER, J. M., AND BEEMAN, D., Eds.
The Book of GENESIS, second ed.
Springer-Verlag, 1998.

[3] CANNON, R. C., HASSELMO, M. E., AND KOENE, R. A.
From biophysics to behavior: Catacomb2 and the design of biologically plausible models for spatial navigation.
Neuroinformatics 1, 1 (2003).

[4] CORNELIS, H., AND DESCHUTTER, E.
Neurospaces : Separating modeling and simulation.
Neurocomputing 52–54 (2003), 227–231.

[5] GODDARD, N. H., HUCKA, M., HOWELL, F., CORNELIS, H., SHANKAR, K., AND BEEMAN, D.
Towards NeuroML : Model description methods for collaborative modelling in neuroscience.
Philosophical Transactions of the Royal Society, Series B: Biological Sciences 356 (2001), 1–20.
Theme Issue organized and edited by Rolf Kötter on "Neuroscience databases - tools for exploring brain structure-function relationships".

[6] HOWELL, F., DYHRFJELD-JOHNSEN, J., MAEX, R., GODDARD, N., AND DESCHUTTER, E.
A large-scale network model of the cerebellar cortex using pgenesis.
Neurocomputing 32 (2000), 1041–1046.

[7] MAEX, R., AND DESCHUTTER, E.
Synchronization of golgi and granule cell firing in a detailed network model of the cerebellar granule cell layer.
Journal of Neurophysiology 80 (1998), 2521–2537.