

# Minimally Redundant Declarative Connectivity Specification for Hierarchically Structured Cell Populations

R. C. Cannon

Institute of Adaptive and Neural Computation  
University of Edinburgh  
5 Forrest Hill, Edinburgh EH1 2QL, Scotland

February 22, 2003

## Abstract

One way in which models can be useful for studying brain function is as a guide to intuition through implementing and exploring candidate mechanisms. In this scenario, models are partly constrained by biological relevance, and partly by the need to achieve particular behaviors such as recalling locations within an environment or learning rules. Although, any particular model could be implemented in a general purpose programming language, it proves more efficient to define higher level constructs that already meet constraints of biological plausibility. These can then be used to rapidly design and explore systems that are consistent with these constraints. Because of their physical compactness, the specification of cell bodies is relatively straightforward (layers, nuclei, regions, populations et al). For the connectivity between cells, however, the situation is substantially more complex. A range of possible methods is considered, from ad-hoc manual wiring, through hierarchies of connectivity matrices to full volume-based specifications. The emphasis is not so much on generating statistically plausible connectivity, but on implementing wiring diagrams that are required for designing large neuron-based circuits to perform specified functions.

## 1 Summary

Previous studies using design based modeling have successfully implemented models of spatial navigation (eg models at <http://askja.bu.edu>) using very simple stereotypical connectivity patterns such as 1-1 and all-all mappings between populations of integrate and fire cells. In these models, much of the complexity that allows them to achieve particular functions is in the synaptic learning rules. However, reliance on weight learning alone is a considerable

hindrance to designing more complex systems, such as models of cortical columns used to explore rule learning. Or, rather, it can be convenient to cut out any modeling of the phase in which axons make contact with particular cells and not with others, and jump straight to a study of the function of particular network architectures. For this, it is necessary to be able to express connectivity patterns in ways that are both intelligible to the designer, and implementable by a software system without needing any further development work.

As a simple example, consider the case of two populations, each composed of sixteen sub-populations where each sub-population contains sixteen cells. Suppose the design calls for connections to be set up such that cell  $i$  in sub-population  $j$  of the first population synapses on cell  $j$  in subpopulation  $i$  of the second. There are therefore 256 connections to be made. This could, of course, be done by hand, but the process is tedious, and, more importantly, disguises the structure of the projection: if it were decided to use 32 cells instead, the existing work would be useless. Alternatively, it could be effected with a very brief *procedural* description, effectively telling the computer how to make the connections in some scripting language. This would map to different sized populations easily, but it has two disadvantages. First, the procedure introduces a spurious implicit ordering and secondly it would be hard to read or to process mechanically in any way other than by running the script. These two objections suggest that a better expression of the connectivity would be purely declarative. The problem with the manual construction suggests that a good description should also be minimally redundant.

A number of possible approaches have now been implemented within the Catacomb modeling environment ([www.compneuro.org/catacomb](http://www.compneuro.org/catacomb)). The first approach is a form of dynamic routing, where the exact connectivity is not implemented in specific connections (although, of course, for plastic synapses there eventually *is* one synapse per connection). Instead, events propagating through the network accumulate tags according to the route they have taken. These tags are used by an implementation of the connectivity specification to decide on future routing. This has certain advantages in terms of economy. For example, where one spike propagates to many cells with a time delay, there need only be one event until a bifurcation in the route is reached. The main weakness of this approach is in the difficulty of querying the system since there is no state information associated with the connectivity.

The most promising approach at present involves a combination of different types of relays that either merge or join their inputs, and either clone or separate their outputs. In this context, “merging” is used for the case that events arriving from different sources become indistinguishable after they leave the relay. By contrast “joining” corresponds to the indexing of events in such a way that their origins can always be deduced. The corre-

sponding behaviors on output are termed “cloning” (as for example happens to a single spike when an axon branches) and “dividing” which is a way to reallocate joined inputs according to their index. The relay-based network architecture can be combined with small manually generated connectivity matrices to implement larger stereotypical structures. For example, the case outlined above, can be expressed as a hierarchy of matrices where only two small matrices need actually be defined.

Although they eliminate the need for custom programming, and reduce the effort involved in designing certain types of network, these declarative models highlight many further challenges in developing specification systems for less stereotypical networks where it may be necessary to express connectivity and connection delays both in terms of a cell’s logical location (position within a hierarchy) and its physical location within a volume of tissue.