Neural Systems Integration
Michael Arnold, Dan Hammerstrom, Marwan Jabri, Terrence Sejnowski

What are the practical issues involved in building large neural systems that model a substantial number of areas in the brain? The creation of such systems is beyond the scope of small research groups, yet there is the need to address the issue outside of the context of large, well-funded, highly structured and heavily managed research initiatives, that is to address the issue within the more fragmented and less well-funded context of many small independent collaborating research teams. The essence of the solution is to integrate the work from many researchers, but we wish to avoid solutions which impose harsh constraints on the individual researcher, in particular the imposition of specific modeling environments or sets of tools. Such impositions often meet with resistance, due to a perceived reduction in immediate effectiveness, a requirement for extensive training and a lack of suitability. It should also be possible to integrate existing work, not just future work that may be carried out within some particular simulation methodology.

One approach is for the integrators to re-build published work within some target modeling environment, and it is possible to argue the value of this approach on two counts. Firstly it validates the original work, and secondly it is the only way of providing the integrators with the requisite sufficient understanding of the component. This approach is costly so we turn instead to a solution based on the integration of a component model as a piece of software.

What constraints do the investigation of large neural systems place on their implementation? Such systems are interesting because they address cognitive function within a semi-complete system, which implies their necessary investigation within the context of complex behavioural tasks. The target implementation may therefore vary with the system being investigated, with requirements varying from interacting with a simulated world in virtual time, to a real world in real time, to specific robotic or prosthetic platforms. The constraints on target implementations are therefore varied and largely unknown.

We are therefore faced with the task of integrating disparate, sometimes pre-existing, software components into a range of frameworks, targeting a variety of physical implementations. This diversity will not just include different physical memory models, but also components implemented within Field Programmable Gate Arrays (FPGAs) or as analogue VLSI.

We are investigating an approach based around the concept of a Network Model Interface (NMI) which defines the interaction between a generic component and a generic implementation framework. It is used to wrap up a component prior to its integration. Other wrappers exist to allow the deployment of NMI-wrapped components within a number of different frameworks, including Matlab. The NMI does not define a framework and it trys not to address framework issues, apart from defining three primary areas of interaction: the passing of data, the passing of control and timing information, and the passing of configuration information. The aim is to seperate the implementation of a component from its deployment. The NMI does impose a master-slave relationship where the framework is the master. The scope of a component is strictly limited to itself. It has no knowledge of the other components within the system and is directly dependent on the framework for all information external to itself.

The NMI is designed specifically for models that fit into a network paradigm of nodes and connections. A component usually represents a set of nodes and defines its connectivity with, and the communication of data with, the rest of the system in terms of sets of input or output nodes. The network paradigm is crucial in that the scope for all possible interactions between components is explicit in the network connectivity and must be implemented through the uni-directional communication of data. The passing of configuration information and the passing of control are considered separate to the passing of data, as there are no mechanisms to pass configuration information or control between components. Even if a particular component does not conceptually reflect a network of neurons, its communication within the system may always be abstracted using the network paradigm. The NMI is implemented as an object-oriented class where a specific type of component exists as a derived class. Data is communicated in packets and multiple arbitary data types are supported through the use of a templated packet type derived from a base packet type.

We are exploring the use of NMI within a system encompassing the ventral pathway, pre-frontal areas, the frontal eye fields, the basal ganglia, the cerebellum and a model of the external world, applied to a task combining working memory directed visual attention with sequence learning and planning. Existing implementations include shared memory architectures and targeted implementations include distributed memory architectures and the use of FPGAs. In particular we have used NMI to integrate from collaborating researchers pre-existing software models for the ventral pathway, frontal eye fields and some pre-frontal areas.

What is left largely unaddressed by this approach of integrating software components are some important systems integration and engineering issues. This is not meant to refer to the software engineering issues but to those relating to the practice of integrating existing components versus their re-implementing as new components. In practice, the function of a component will never be transparent. This makes it difficult to both validate the component and to gain sufficient understanding of the component. These and managing the knowledge required to do the integration are the real issues in building large neural systems, not issues of parallel programming and software methodologies. This is reflected in those approaches based around the use of markup languages to structure the information within the domain of biological models. The approach based around the NMI differs from those and others in that it focuses on how large neural systems can be implemented in the short term and with limited resources. This acknowledges the unknown nature of the integration problems and the need to identify them within the context of actual experience. In defining this approach we have tried to identify a shortest path to allow us to study the problem at first hand.