ASL/NSL: A Multi-level Computational Model for Neural Simulation[1]

Alfredo Weitzenfeld
Departmento Académico de Computación
Instituto Tecnológico Autónomo de México (ITAM)
Río Hondo #1, San Angel Tizapán, CP 01000
México DF, MEXICO
email: alfredo@lamport.rhon.itam.mx
tel: (525) 6284060, fax: (525) 6162211

Extended Abstract

In this paper we present an overview of the ASL/NSL computational model for the simulation of neural systems. This work is motivated by the quest to simulate animal-like behavior as faithfully as possible based on existing ethological, physiological and anatomical neural data. At the higher level these models are described in terms of *schema* [1] modules corresponding to behavior agents (ethology) simulated with ASL (*Abstract Schema Language*) [4]. At the lower level, neural networks are described in terms of neural modules (physiology and anatomy) simulated with NSL (Neural Simulation Language) [5]. Many models have been developed and simulated with ASL/NSL (see [6] for examples of models involving perception, visuomotor coordination, motor control as well as technological neural applications). For example, in Figure 1 we show a simplified diagram of the toad's prey acquisitions and prey avoidance model [2], (schema level 1) involving *perceptual* and *motor* schemas. The external visual input is processed to generate appropriate motor actions: forward, orient, snap and duck. Schemas at this level are decomposed and delegated to the next level down (schema level 2) where schemas perform more specific tasks. Prey approach and predator avoid schemas delegate their tasks to a *schema assemblage* composed of a prey/predator recognizer, a prey/predator selector, depth and heading translators and maps. Next level down, different neural modules: *Retina*, *Tectum*, *Maximum Selector* and *Cue Interaction*, if available, implement the actual neural network processing.

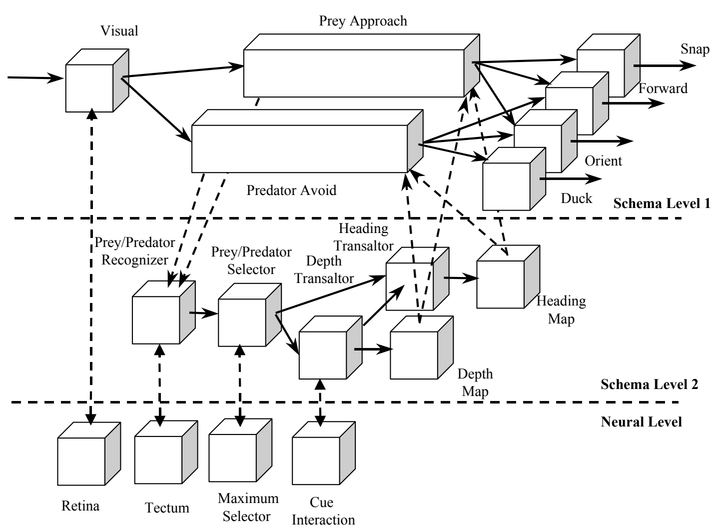

Figure 1. Schema hierarchy for the toad's prey acquisition and predator avoidance models. The top two levels correspond to schema levels (1 and 2) and the lower level corresponds to neural modules.

Schemas and Neural Networks

To better understand the underlying computational model, the ASL/NSL defines a tree-like schema or module hierarchy as shown in Figure 2. Starting by the root module (known as the model), modules are further decomposed into additional *submodules*, having no limit on how many levels this may reach. At the same abstraction level, modules are interconnected (solid arrows), while at different levels modules have their task delegated (dashed

[1] We thank the NSF-CONACyT collaboration grant (#IRI-9522999 in the US and #546500-5-C018-A in Mexico), the CONACyT REDII grant in Mexico, as well as the "Asociación Mexicana de Cultura, A.C.".

arrows). Networks of submodules – *module assemblages* – are seen in their entirety in terms of a single higher-level module and may be implemented independently from each other in both top-down and bottom-up fashion, an important benefit of modular design. At the higher abstraction levels, the detailed module implementation is left unspecified, only specifying the module's interface and what is to be achieved. At the lowest level, schemas are implemented by neural modules.

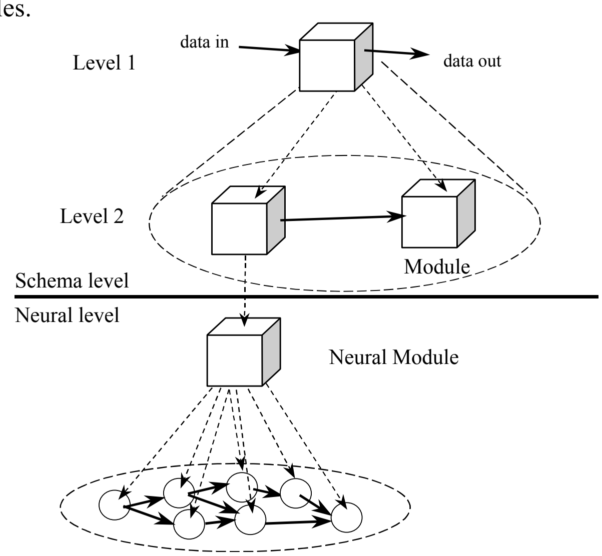

Figure 2. The ASL/NSL computational model is based on hierarchical interconnected modules. A schema module at a higher level (level 1) is decomposed (dashed lines) into additional interconnected (solid arrow) schema submodules (level 2). At the lowest level, neural modules are implemented by neural networks (circular objects).

As a computational unit, every module incorporates its own local structure and control mechanisms. Every module defines an external interface made of a set of unidirectional input and output *ports* supporting data passing between modules together with a set of public methods that can be externally invoked from other modules. Communication between modules is in the form of asynchronous message passing for both data and methods. Internally, communication is hierarchically managed through anonymous data port reading and writing. Externally, communication is managed through dynamic port *connections* (solid arrows) - links between output ports in one module to input ports in another module - and *relabelings* (dashed arrows) - module ports at one level in the hierarchy are linked to similar input or output ports at a different level. The hierarchical port management approach enables the development of neural architectures where modules may be designed and implemented independently and without prior knowledge of the complete model or their final execution environment, encouraging component reusability and permitting module execution in a distributed fashion [7].

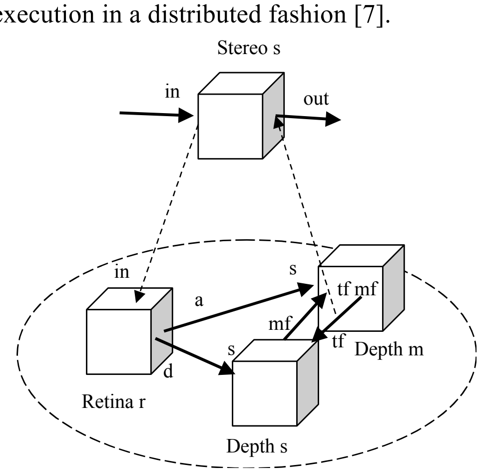

Figure 3. The *Stereo* module contains an *in* input port and an *out* output port. It is further decomposed into a *Retina* module containing an input port *in* and two output ports, *d* and *a*, for disparity and

accommodation, respectively. The *Depth* module consists of an input port *s*, receiving data from the *Retina*, a second input port *tf*, receiving input from the other *Depth* module, and an output port *mf*.

For example, let's consider the *depth perception* problem where a three dimensional scene is presented to the two eyes. The depth perception model developed by House [3] uses two systems to build a depth map, one driven by *disparity* cues - difference in retina projection - while the other is driven by *accommodation* cues - receiving information about focal length. The corresponding simplified ASL/NSL model consists of a *Stereo s* root module and three interconnected submodules: *Retina r*, *Depth m* (accommodation) and *s* (disparity), as shown in Figure 3.

References

[1] Arbib, M.A., Schema Theory, in the *Encyclopedia of Artificial Intelligence*, 2nd Edition, Editor Stuart Shapiro, 2:1427-1443, Wiley, 1992.

[2] Cobas, A., and Arbib, M.A., Prey-catching and Predator-avoidance in Frog and Toad: Defining the Schemas, J. Theor. Biol 157, 271-304, 1992.

[3] House, D., Depth perception in frogs and toads: A study of in neural computing, *in Lecture notes in Biomathematics*, 80, Springer-Verlag, 1989.

[4] Weitzenfeld, A., ASL: Hierarchy, Composition, Heterogeneity, and Multi-Granularity in Concurrent Object-Oriented Programming, *Proceedings of the Workshop on Neural Architectures and Distributed AI: From Schema Assemblages to Neural Networks*, USC, October 19-20, 1993.

[5] Weitzenfeld, A., Arbib, M.A., NSL, Neural Simulation Language, in *Neural Networks Simulation Environments*, Editor J. Skrzypek, Kluwer, 1994.

[6] Weitzenfeld, A., Arbib, M., Alexander, A., *NSL - Neural Simulation Language: System and Applications*, MIT Press, 2000 (to be published).

[7] Weitzenfeld, A., Peguero, O., Gutierrez, S., NSL/ASL: Distributed Simulation of Modular Neural Networks, in *Proc of MICAI 2000*, Acapulco, Mexico (to be published).