A Multi-level Approach to Biologically Inspired Robotic Systems[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

Abstract

The study of biological systems has inspired the development of a large number of neural network architectures and robotic implementations. Through both experimentation and simulation biological systems provides a means to understand the underlying mechanisms in living organisms while inspiring the development of robotic applications. Experimentation, in the form of data gathering (ethological physiological and anatomical), provides the underlying data for simulation generating predictions to be validated by theoretical models. These models provide the understanding for the underlying neural dynamics, and serve as basis for simulation and robotic experimentation. Due to the inherent complexity of these systems, a multi-level analysis approach is required where biological, theoretic and robotic systems are studied at different levels of granularity. The work presented here overviews our existing modeling approach and describes current simulation results.

1 Introduction

The study of biological systems comprises a cycle of biological experimentation, computational modeling and robotics experimentation, as depicted in Figure 1. This cycle serves as framework for the study of the underlying neural mechanisms responsible for behavior in animals and serving as inspiration in designing robotic systems.

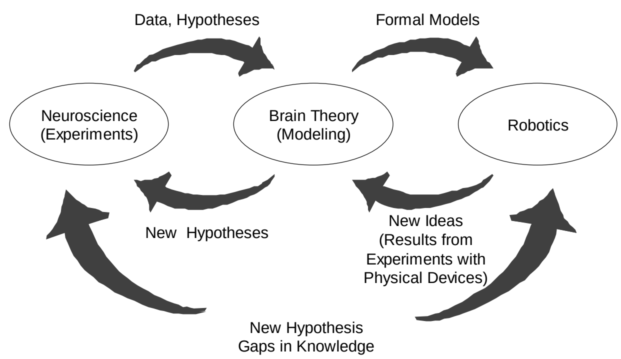

Figure 1. Framework for the study of living organisms through cycles of biological experimentation, computational modeling, and robotics experimentation.

To address the underlying complexity in building biologically inspired robotic systems we have developed a multi-level analysis approach integrating across different modeling and simulation levels studied primarily with respect to four different ones: (1) autonomous robotic agents, (2) behavior, (3) neural networks, and (4) detailed neurons.

1. At the highest level, autonomous robotic agents are designed to interact with the world via sensors and actuators. These agents are simulated in virtual autonomous agents and implemented in real robots. Autonomous robotic agents are exemplified by biologically inspired systems, such as the computational frog (*rana computatrix*) [1], the computational praying mantis [5], the computational cockroach [6], and the computational hoverfly [11].

[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.".

2. At the behavior level, neuroethological data from living animals is gathered to generate single and multi-agent systems to study the relationship between an agent and its environment, giving emphasis to aspects such as cooperation and competition between agents. We describe agent behavior in terms of perceptual and motor *schemas* [3] decomposed and refined in a recursive fashion. Behaviors, and their corresponding schemas, are simulated via the Abstract Simulation Language ASL [22]. Examples of behavioral models include the praying mantis *Chantlitlaxia* ("search for a proper habitat") [9] and the frog and toad prey acquisition and predator avoidance models [12].
3. At the neural network level, neuroanatomical and neuronphysiological data are used to generate perceptual and motor neural network models corresponding to schemas developed at the behavioral level. These models try to explain the underlying mechanisms for sensorimotor integration. Neural networks are simulated via the Neural Simulation Language NSL [24][25]. Neural network models are exemplified by the prey acquisition and predator avoidance neural models [10] and the toad prey acquisition with detour behavior model involving adaptation and learning [13].
4. At the detailed neural level, electrochemical neural mechanisms are studied to understand different neural phenomena such as synaptic plasticity and presynaptic inhibition. A number of models are used depending of the mechanisms simulated, such as the compartmental model, where a single axon is divided in compartments [19], and the ion kinetics model, where chemical concentration responsible for electric current is simulated [18]. These models are simulated with systems such as GENESIS [7] and NEURON [17].

2 Modeling Levels

In the following sections we overview the different modeling levels using as an example *rana computatrix* [Arbib 1987] behaviors inspired on biological studies of frogs and its application to different robotic experiments.

2.1 Autonomous Robotic Agents

Autonomous robotic agents can be either simulated in a virtual world or executed in the real world. In particular, frogs (and toads) and the corresponding *rana computatrix* use vision and tact as their primary sensors with legs and tongue as their primary actuators, both virtual and real. In Figure 2 we show an illustration of a frog in a setup involving a prey (worm) interposed by a fencepost.

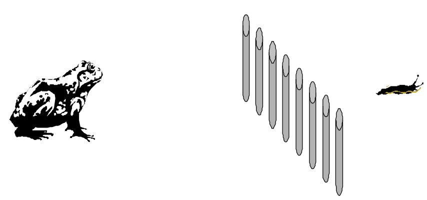

Figure 2. Computational frog in a prey and barrier setup.

2.2 Behaviors

Behaviors are described by ethograms, as the one shown in Figure 3 defining *rana computatrix* behaviors.

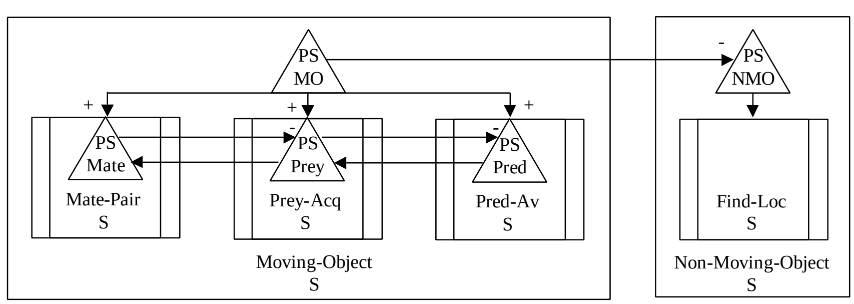

Figure 3. *Rana Computatrix* ethogram: Mating, Prey Acquisition and Predator Avoidance schemas (moving and non-moving objects) [8]. The diagram shows feedback between perceptual schemas (triangles) and regular schemas (rectangles). Note the hierarchical schema organization. (Acronyms are as follows: PS - Perceptual Schema, MO - Moving Object, NMO - Non-Moving Object, S -Schemas)

In Figure 4 we show in more detail a typical prey acquisition behavior for the frog.

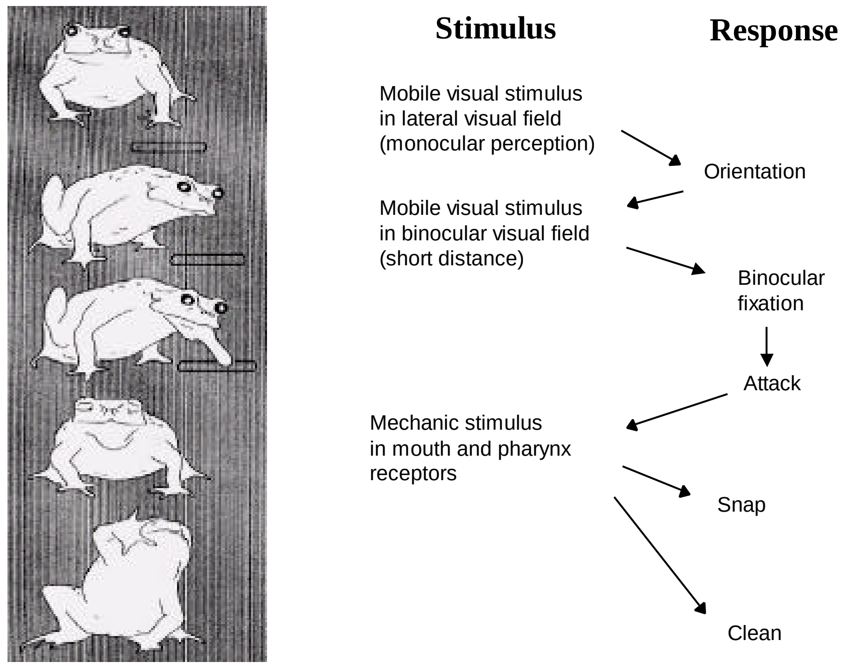

Figure 4. Frog's prey acquisition behavior involving a worm as shown on the left-hand side. The right-hand side describes the frog's response in relation to the stimulus [16].

The particular behavior we will describe in more detail is the frog's prey acquisition with detour as introduced previously in Figure 2. The setup involves a frog and a barrier in front of a prey, where fencepost gaps have the same width, with the following experiments carried out [13] and shown in Figure 5.

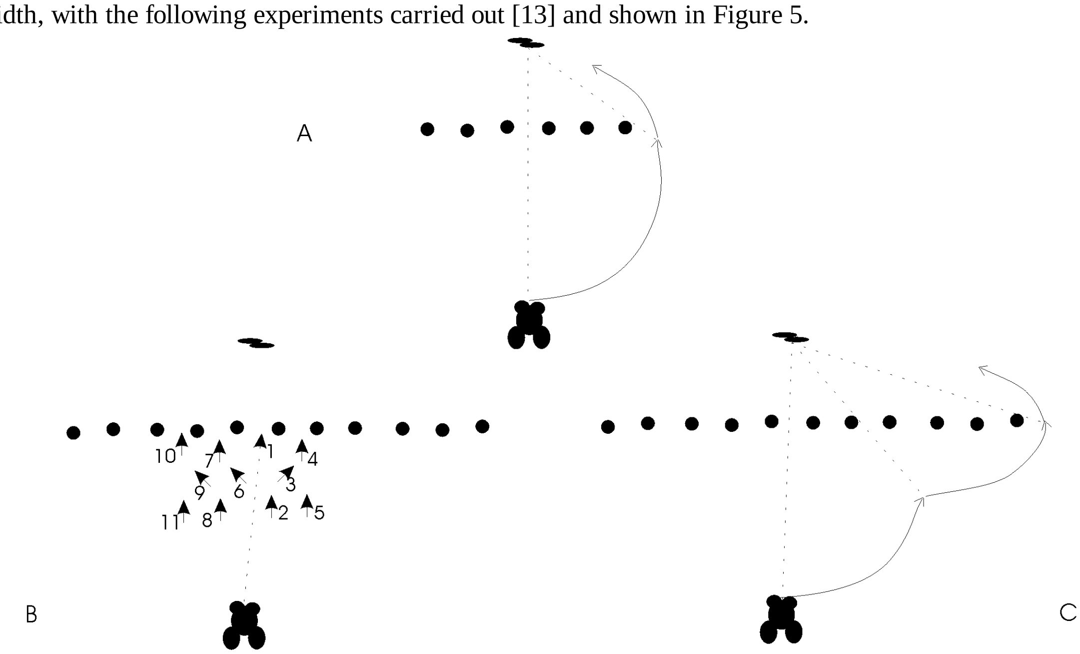

Figure 5. **A.** Approach to prey with single 10cm barrier with immediate detour. **B.** Approach to prey with single 20 cm barrier: first trial with frog in front of 20cm barrier (numbers indicate the succession of the movements). The toad directly approaches de center of the barrier requiring successive trials to manage the detour around it. **C.** Approach to prey with single 20cm barrier. After 3 trials the frog detours directly around the 20cm barrier. Arrowheads indicate the position and orientation of the frog following a single continuous movement after which the frog pauses.

- **Experiment I**: Barrier 10cm Wide**.** Frogs that started from a long enough distance (15-25cm) in front of a 10cm wide barrier (and with the worm 10cm behind the barrier) showed (in 95% of the trials) reliable detour behaviors from the first interaction with the 10cm barrier. They produced an immediate approach movement towards one of the edges of the barrier.
- **Experiment II**: Barrier 20 cm wid**e.** The "naïve" frog (a frog that has not been yet exposed to the barrier) tends to go for a fencepost gap in the direction of the prey (this was the case for 88% of the trials). The frog starts out approaching the fence trying to make its way through the gaps. During the first trials the frog goes straight towards the prey thus bumping into the barrier. Since the frog is not able to go through a gap it backs-up about 2cm and then reorients towards one of the neighboring gaps. After 2 (43%) or 3 (57%) trials, the "trained" frog is already detouring around the barrier without bumping into the barrier. The behavior involves a synergy of both forward and lateral body (sidestep) movements in a very smooth and continuous single movement.

In order to model such behaviors we introduce the schema computational model. Schemas define a hierarchical distributed model for action-perception control, where each schema incorporates its own structure and control mechanisms. At the higher abstraction levels, the detailed schema implementation is left unspecified, only specifying what is to be achieved. At a lower level, schemas are implemented with neural networks or other processes. The schema computational model follows a tree-like structure as shown in Figure 6 (schemas may also be shared making the structure a directed graph). At the top, a high level schema is decomposed into two lower level schemas where the three schemas together are known as schema *aggregates*, or *assemblages*. When at the same level, schemas are interconnected (solid arrows), or when at different levels, schemas are relabeled having their task delegated (dashed arrows).

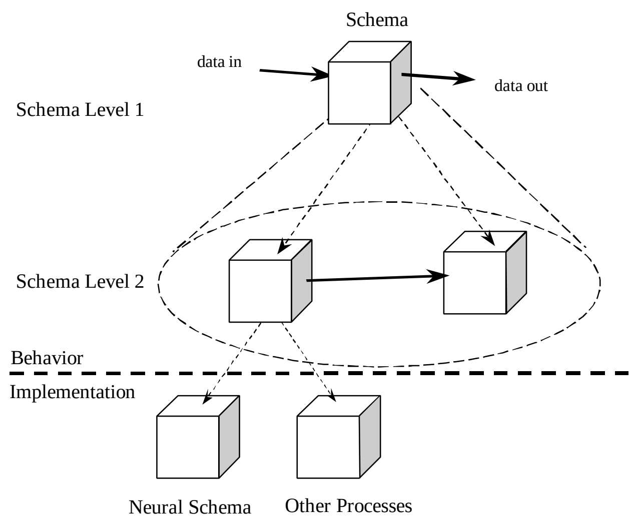

Figure 6. The ASL/NSL computational model is based on hierarchical interconnected schemas. A schema at a higher level (level 1) is decomposed (dashed lines) into additional interconnected (solid arrow) subschemas (level 2). At the lowest level schemas are implemented by neural networks or other processes.

The schema interface consists of multiple unidirectional control/data, input and output ports having a body where schema behavior is specified, as shown in Figure 7. Communication is in the form of asynchronous message passing, hierarchically managed, internally, through anonymous port reading and writing, and externally, through dynamic port *connections* and *relabelings.*

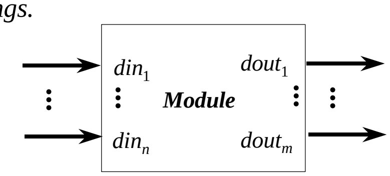

Figure 7. Each schema may contain multiple input, $din_1,...,din_n$, and output, $dout_1,...,dout_m$, ports for unidirectional communication.

When doing connections, output ports from one schema are connected to input ports from other schemas, and when doing relabelings, ports of similar type (input or output) belonging to schemas at different levels in the hierarchy are linked to each other. The hierarchical port management methodology enables the development of distributed architectures where schemas may be designed in a top-down and bottom-up fashion implemented independently and without prior knowledge of the complete model or their final execution environment, encouraging component reusability.

Figure 8 shows the schema model hierarchy corresponding to the toad's prey acquisitions with detour model [14]. We show a single schema level (level 1) describing the different behaviors being modeled, primarily *prey approach* and *static object avoid.* Additional schemas include visual and tactile input, moving stimulus selector (when more than one prey exists), prey and static object recognizers together with the four types of motor actions: forward, orient, sidestep and backward. Tasks at this level are delegated to the next level down, the neural level, where schemas perform more refined tasks. In this model, both the prey approach and the static object avoid schemas are implemented by neural schemas: a *Retina* [21], *Maximum Selector* [15], *Tectum* and *PreTectum-Thalamus* [8], together with neural motor heading maps.

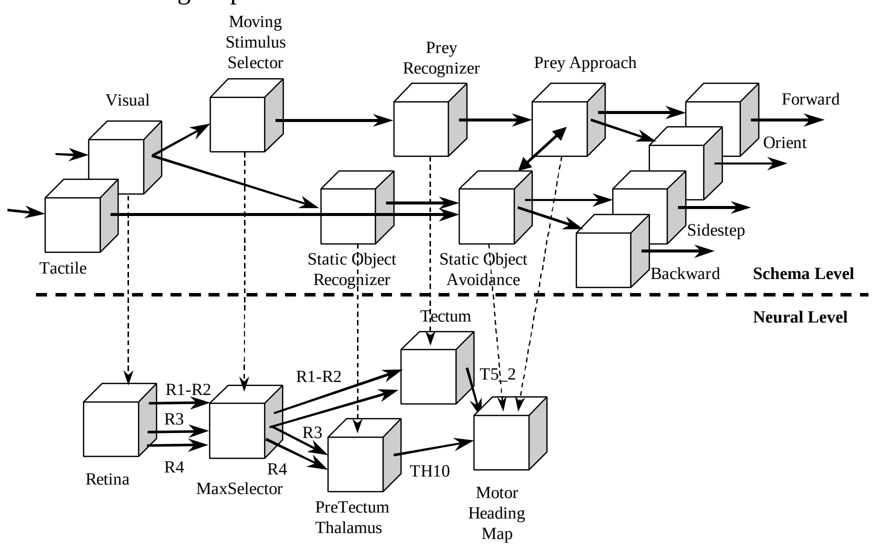

Figure 8. Schema model hierarchy for the toad's prey acquisition and static object avoidance model previously described.

Complexity is much more significant when considering more behaviors and other brain regions [4].

2.3 Neural Networks

Biologically inspired neural networks are based on physiological and anatomical neural mappings. For example, Figure 9 shows a diagram of different neural areas involved in the frog's prey acquisition and detour model.

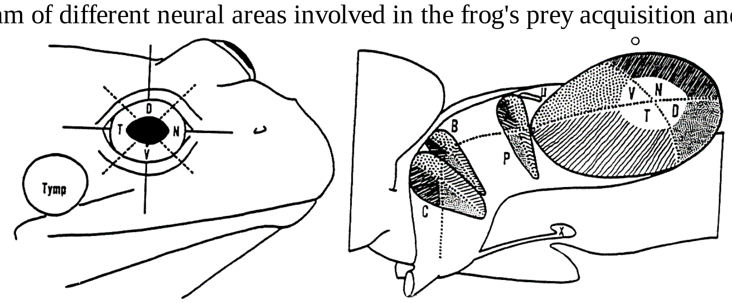

Figure 9. The two illustrations show the most important areas in the frog's prey acquisition model. These are the Optic Tectum (O) (divided in four regions: Temporal (T), Dorsal (D), Nasal (N) and Ventral (V)), the Thalamic Pretectal Neuropil (P), together with other regions: Nucleus of Belonci (B), Lateral Geniculate Nucleus (C) and Basal Optic Root (X) [20].

Neural schemas provide their implementation in terms of neural networks processing, as shown in Figure 10.

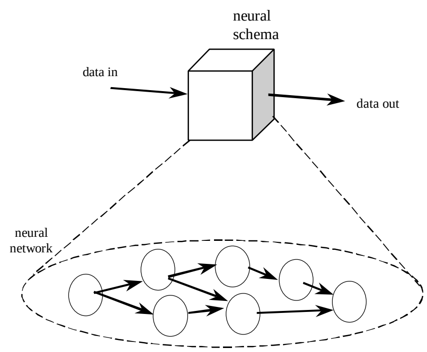

Figure 10. Neural schema hierarchy showing task delegation to neural networks processing.

At this level, neural networks are simple processing units interconnected among each other to provide large-scale computation. Each neuron is defined by its membrane potential value *m* depending on its previous history and current input s_m while its output value *M* is defined by a non-linear threshold function over its membrane potential, as shown in Figure 11. For example, the leaky integrator model [2] is used to simulate such neurons.

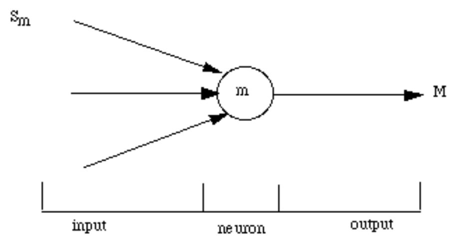

Figure 11. Simple neural element as basic component at the neural network level.

For example, at this level of granularity the MaxSelector [15] neural schema is implemented by the neural network shown in Figure 12.

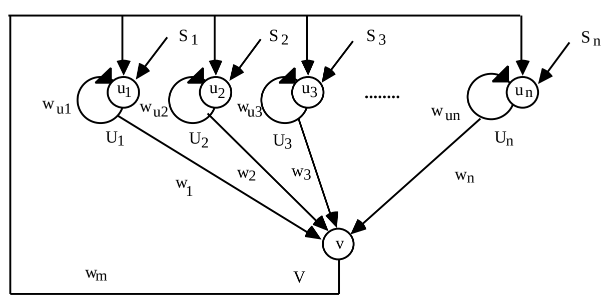

Figure 12. The neural network shown corresponds to the architecture of the Maximum Selector model, where u_i and v represent neural membrane potentials, U_i and V represent neural firing rates, S_i represent inputs to the network, and w_i represent connection weights. The network is initialized with a number of positive inputs assigned to different cells. After many iterations the network stabilizes producing a single "winner", i.e. a single active cell.

The neural schema model also provides an extended model where neurons themselves may have their task delegated by neural implementations of different levels of detail, from the very simple neuron models to very detailed ones [23].

2.4 Neurons

Neuron models vary in their detail, depending on the particular simulated mechanisms, involving at the top level of a soma (nucleus of the neuron), an axon (output of the neuron), and dendrites (input to the neuron). Connections between neurons take place in the synapses at the axon terminals of one neuron connected to the dendrites of another neuron. Synapses are the main mechanism for plasticity in neurons and can be further refined into much more detail, as shown in Figure 13.

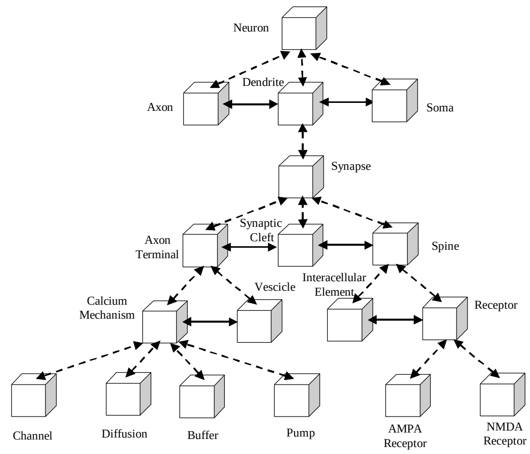

Figure 13. Neural modeling at different levels of details.

3 Simulation Results

Due to space limitations in this paper we only show the resulting path motion seen at the top level for the previous basic experiments, as shown in Figure 14. Additional graphs (not shown here) display neural network states for the different neural schemas.

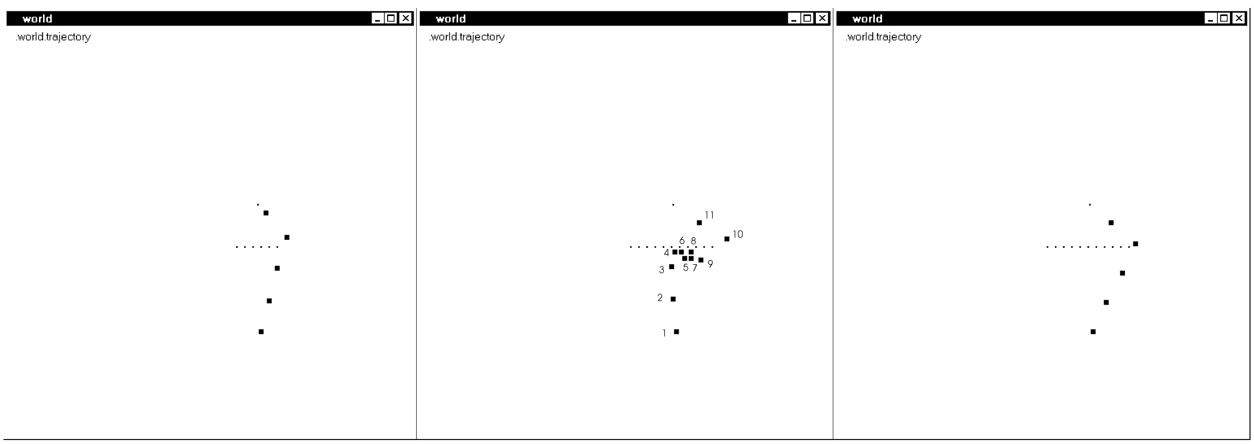

Figure 14. The above diagrams display the *Rana Computatrix* basic experiments for the prey acquisition and detour model. The different dots correspond to the frog's trajectory from its initial location as it finally reaches the prey. The left-hand side shows the resulting motion path for the 10cm barrier. Note how the frog heads directly towards the side of the barrier. The middle diagram displays the resulting motion path for the 20cm barrier experiment before learning. We have added numbers corresponding to the frog's position in time. In this particular experiment the frog hits the barrier three times before perceiving the side of the barrier and moving towards the prey. The right-hand side diagram shows the resulting motion path for the 20cm wide barrier after learning.

4 Discussion

The work presented here overviews the inherent complexity in modeling biological systems. This complexity can be managed by taking a multi-level approach emphasizing both top-town and bottom-up designs through different granularity levels. At the top level agents are defined in terms of sensors and actuators and may involve interaction with other agents, such as in competition and cooperation. Next level down, each agent is described in terms of its behaviors such as in the frog's prey acquisitions with detour model. Once basic behaviors are defined additional ones may be added taking advantage of the underlying schema architecture. Next level down, behaviors are implemented by different (or common) neural schemas representing neural network processing. The detailed neuron bottom level is required only when simple neural models do not provide sufficient processing capabilities such as those requiring synaptic plasticity or presynaptic inhibition. Current work involves experimentation with these and other models and applying them to robots to provide the feedback in experimentation as described in Figure 1.

References

[1] Arbib, M.A., Levels of Modelling of Mechanisms of Visually Guided Behavior, *Behavior Brain Science* 10:407-465, 1987.

[2] Arbib, M.A., *The Metaphorical Brain 2*, Wiley, 1989.

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

[4] Arbib, M.A., Erdi, P. and Szentagothai, J., *Neural Organization: Structure, Function and Dynamics*, MIT Press, 1998.

[5] Arkin, R.C., Ali, K., Weitzenfeld, A., and Cervantes-Perez, F., Behavior Models of the Praying Matis as a Basis for Robotic Behavior, *in Jouranl of Robotics and Autonomous Systems*, 2000 (to be published).

[6] Beer, R. D., *Intelligence as Adaptive Behavior: An Eperiment in Computational Neuroethology*, San Diego, Academic Press, 1990.

[7] Bower, J.M., and Beeman, D., *The Book of GENESIS, Exploring Realistic Neural Models with the GEneral NEural SImulation System*, Telos, Springer-Verlag, 2nd Edition, 1998.

[8] Cervantes-Perez, F., Lara, R., and Arbib, M.A., A neural model of interactions subserving prey-predator discrimination and size preference in anuran amphibia, Journal of Theoretical Biology, 113, 117-152, 1985.

[9] Cervantes-Perez, F., Franco, A., Velazquez, S., Lara, N., 1993, A Schema Theoretic Approach to Study the 'Chantitlaxia' Behavior in the Praying Mantis, *Proceeding of the First Workshop on Neural Architectures and Distributed AI: From Schema Assemblages to Neural Networks*, USC, October 19-20, 1993.

[10] Cervantes-Perez, F., Herrera, A., and García, M., Modulatory effects on prey-recognition in amphibia: a theoretical 'experimental study', in *Neuroscience: from neural networks to artificial intelligence*, Editors P. Rudoman, M.A. Arbib, F. Cervantes-Perez, and R. Romo, Springer Verlag Research Notes in Neural Computing vol 4, pp. 426-449, 1993.

[11] Cliff, D., Neural Networks for Visual Tracking in an Artificial Fly, in *Towards a Practice of Autonomous Systems: Proc. of the First European Conference on Artifical Life (ECAL 91)*, Editors, F.J., Varela and P. Bourgine, MIT Press, pp 78-87, 1992.

[12] 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.

[13] Corbacho, F., and Arbib M. Learning to Detour, *Adaptive Behavior*, Volume 3, Number 4, pp 419-468, 1995.

[14] Corbacho, F., and Weitzenfeld, Learning to Detour, in *The Neural Simulation Language NSL, System and Applications*, MIT Press, 2000 (to be published).

[15] Didday, R.L., A model of visuomotor mechanisms in the frog optic tectum, *Math. Biosci.* 30:169-180, 1976.

[16] Ewert, J.P, *Neuroethology, an introduction to the neurophysiological fundamentals of behavior*, Springer-Verlag, 1980.

[17] Hines, M., and Carnevale, T., The NEURON Simulation Environment, *Neural Computation*, 9:1179-1209, 1997.

[18] Hodgkin, A.L. and Huxley, A.F., A quantitative description of membrane current and its application to conduction and excitation in nerve, *Journal of Physiology*, 117, 500-544, 1952.

[19] Rall, W., Branching dendritic trees and motoneuron membrane resistivity, *Exp. Neurol*., 2:503-532, 1959.

[20] Scalia, F., and Fite., K.V., A retinotopic analysis of the central connections of the optic nerve in the frog, J. Comp. Neurol., 158:455-478.

[21] Teeters, J.L., and Arbib, M.A., A model of the anuran retina relating interneurons to ganglion cell responses, *Biological Cybernetics*, 64, 197-207, 1991.

[22] 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.

[23] Weitzenfeld, A., Arbib, M., A Concurrent Object-Oriented Framework for the Simulation of Neural Networks, *Proceedings of ECOOP/OOPSLA '90 Workshop on Object-Based Concurrent Programming, OOPS Messenger*, 2(2):120-124, April 1991.

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

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