A Neural Schema System Architecture for Autonomous Robots

Alfredo Weitzenfeld, Ronald Arkin, Francisco Cervantes, Roberto Olivares and Fernando Corbacho

Abstract —**As autonomous robots become more complex in their behavior, more sophisticated software architectures are required to support the ever more sophisticated robotics software. These software architectures must support complex behaviors involving adaptation and learning, implemented, in particular, by neural networks. We present in this paper a neural based *schema* [2] system architecture for the development and execution of autonomous robots in both simulated and real worlds. This architecture has been developed in the context of adaptive robotic agents, *ecological robots* [6], cooperating and competing with each other in adapting to their environment. The architecture is the result of integrating a number of development and execution systems: NSL, a neural simulation language; ASL, an abstract schema language; and *MissionLab*, a schema-based mission-oriented simulation and robot system. This work contributes to modeling in Brain Theory (BT) and Cognitive Psychology, with applications in Distributed Artificial Intelligence (DAI), Autonomous Agents and Robotics.**

Index terms —**Autonomous Robots; Autonomous Agents; Schemas; Neural Networks; Architecture.**

I. Introduction

To enable the development and execution of complex behaviors in autonomous robots involving adaptation and learning, sophisticated software architectures are required. The neural schema architecture provides such a system, supporting the development and execution of complex behaviors, or *schemas* [3][2], in a hierarchical and layered fashion [9] integrating with neural network processing.

In general, *schema theory* helps define brain functionality in terms of concurrent activity of interacting behavioral units called *schemas*. Schema-based modeling may be specified purely on behavioral data (*ethology*), while becoming part of a neural based approach to adaptive behavior when constrained by data provided by, e.g., the effects of brain lesions upon animal behavior (*neuroethology*). Schema modeling provides a framework for modeling at the purely behavioral level, at the neural network level or even below [28]. In terms of neural networks, *neural schema theory* provides a functional/structural decomposition, in strong contrast with models which employ learning rules to train a single, otherwise undifferentiated, neural network to respond as specified by some training set. Neural schema-based modeling proceeds at two levels: (1) model behavior in terms of schemas, interacting functional units; (2) implementation of schemas as neural networks based on neuroanatomical and neurophysiological studies. What makes the linking of structure and function so challenging is that, in general, a functional analysis proceeding "top-down" from some overall behavior need not map directly into a "bottom up" analysis proceeding upwards from the neural circuitry.

The work described in this paper is the product of a collaborative research depicted in Fig. 1.

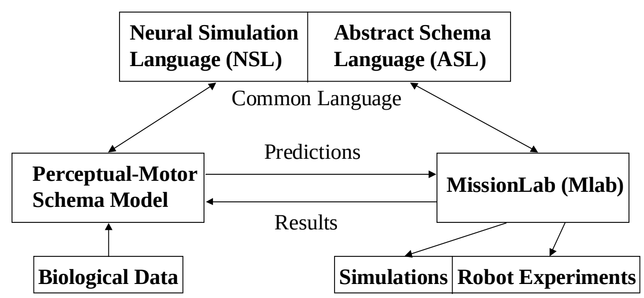

Fig. 1. The diagram describes the collaboration research as the integration of different research components. Biological data from experiments serves as input to perceptual and motor schema modeling. These models generate predictions on robotic behavior carried out in MissionLab simulated or executed in real world robots. The results serve as feedback to calibrate the theoretical models. These two components are integrated via the Neural Simulation and Abstract Schema languages

Biological data from behavioral studies in the praying mantis "Chantitlaxia" [11] and the frog and toad prey acquisition and predator avoidance behaviors [12][14], are used to generate neural schema models: *perceptual* schemas, dealing with sensory input or perceptions; *motor* schemas, dealing with motor action; and *sensorimotor* schemas, integrating between sensory input and motor action. These studies are modeled in terms of computational schemas in the Abstract Schema Language (ASL) [25], implemented as neural networks in the Neural Simulation Language (NSL) [27], and simulated in a virtual world or executed in the real world with the *MissionLab* (Mlab) robotic system [23].

Alfredo Weitzenfeld and Francisco Cervantes are faculty members at the Departamento Académico de Computación in the Instituto Tecnológico Autónomo de México. Ronald Arkin is a faculty member at the College of Computing in Georgia Institute of Technology. Roberto Olivares is a graduate student at the College of Computing, Georgia Institute of Technology. Fernando Corbacho is a faculty member at the Departamento de Ingeniería Informática in the Universidad Autónoma de Madrid. For further information please contact Alfredo Weitzenfeld at alfredo@lamport.rhon.itam.mx.

II. SCHEMAS, NEURAL NETWORKS AND AUTONOMOUS ROBOTS

The neural schema system architecture for autonomous robots comprises the integration of three separately developed architectures, each built to support a different aspect of schema modeling.

A. *Schemas*

As a computational model, schemas define a hierarchical and distributed architecture for the development of complex adaptive systems. A number of schema-based architectures have been developed for different application domains, e.g. VISIONS [18], in vision; RS (Robot Schemas) [22] and MissionLab [3], in robotics. Based on these domain specific architectures, a unified schema computational model, ASL (Abstract Schema Language) [25], was designed with the ability to integrate with neural networks processing across different domains as well. Schemas in ASL are hierarchical and distributed autonomous agents, where ASL integrates concurrent object-oriented programming methodologies [29] with agent modeling [8]. As a language ASL corresponds more to a specification language rather than to an explicit programming language. The detailed implementation is left unspecified, only specifying what is to be achieved. Different implementations may correspond to a single schema, where implementation is in terms of neural networks or other schema process. The ASL computational model is shown in Fig. 2.

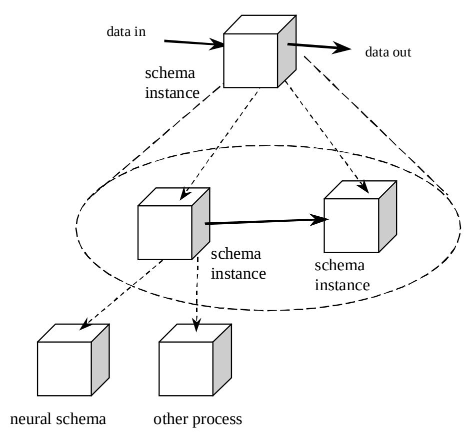

Fig. 2. A schema is shown decomposed into other schemas. This decomposition gives rise to schema aggregation, or schema *assemblages*. Schemas are specified and implemented either through *wrapping*, which enables static integration of external programs, or through *task delegation*, which enables dynamic integration of schemas as separate specification and implementation tasks. (Solid arrows between boxes represent connections between objects, while dashed arrows represent task delegation.)

Schema interfaces consists of multiple unidirectional control or data, input and output *ports*, and a method section where schema behavior is specified. 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*. When doing connections, output ports from one schema are connected to input ports from other schemas, and ports from schemas at different hierarchies are linked to each other when doing relabelings. The hierarchical port management methodology enables the development of distributed systems where objects may be designed and implemented independently and without prior knowledge of their final execution environment, encouraging model reusability. This supports both top-down and bottom-up system designs as required by neural schema modeling.

In order to support complex schema modeling, ASL is design as a distributed multithreaded system architecture, executing on different platforms [10], as shown in Fig. 3.

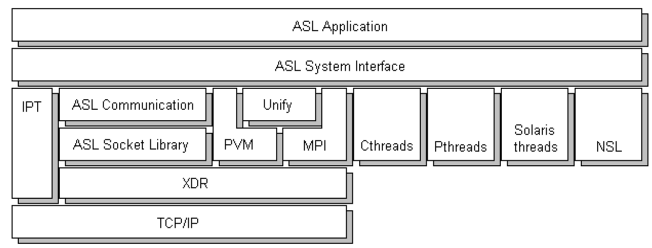

Fig. 3. The Abstract Schema Language (ASL) System Architecture consists of a top application layer where schema models are described. The next layer down is the system interface integrating with the underlying distributed architecture and libraries. In this diagram we show a number of libraries supported by ASL, including its own communication and socket library.

B. *Neural Networks*

Neural networks serve as the underlying implementation for neural schemas. Lower level neural network components integrate with higher level schemas, as shown in Fig. 4.

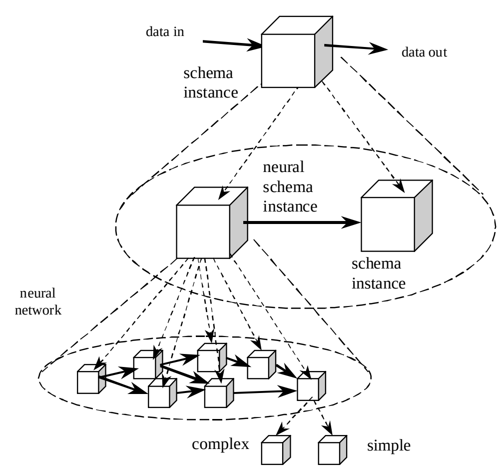

Fig. 4. The neural schema hierarchy extends the basic schema hierarchy with neural processes. A schema is mapped to a full neural network made of any number of interconnected neurons. Each neuron may be described as simple or complex neural processes following in itself the schema hierarchy.

The Neural Schema Language (NSL) [27] provides the linkage to ASL, enabling the integration of neural networks as schema implementations. The ability to implement schemas through different neural networks results in the added benefit of enabling the construction of distributed neural networks. Mapping between schemas and neural networks may not only be 1 to 1, but also many to many. The neural schema model not only enables the encapsulation of neural networks into schemas, but 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 the very complex ones [26].

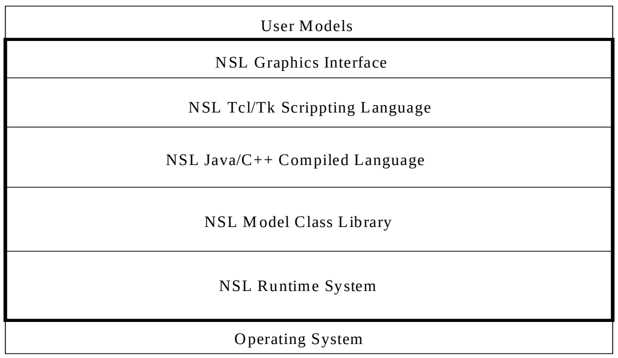

Fig. 5. The NSL System Architecture consists of a number of layers. From the top, user models are controlled via the graphics interface integrating with the Tcl/Tk scripting language. This layer is built on top of the Java/C++ modeling language. The modeling language integrates with the model class library. Model executing is carried by the NSL runtime system.

The NSL system architecture is shown in Fig. 5. Models are described via a compiled language, where graphics displays and a scripting language provide the interfacing mechanisms between the model and the user. Two implementations of the system currently exist: NSLC in C++ and NSLJ in Java.

C. Schema-based control for autonomous robots

In robotics, schemas have been used to provide the underlying software control mechanisms for a number of systems, e.g. MissionLab [3] and RS [22]. In particular, in the control of autonomous robots, such as with MissionLab, motor schemas have been encoded as a variant of the potential field methodology [21]. In this context, schemas have the following characteristics:

1. Each is an independent asynchronous computational agent executing in parallel with other schemas.
2. Sensing is directly tied to motor control following the action-oriented perception paradigm, where information is obtained via sensing on a need-to-know basis [4].
3. Each active schema produces a vector that encodes the behavioral response for a given stimulus.
4. Coordination of schemas is typically conducted via behavioral fusion: vector summation and normalization of the individual schemas outputs.
5. Schemas can be aggregated into assemblages, which provide a higher level of abstraction.
6. Their use is rooted in neuroscientific and psychological theory.

Fig. 6. The picture shows a collection of schema-based robots at the Mobile Robot Laboratory at Georgia Tech. All robots are autonomous varying in their complexity from different sized wheeled robots to legged ones.

This particular form of behavioral control has been tested on a wide range of robotic systems: from teams of small robots used for competitions to military sized vehicles [5], as shown in the Fig. 6.

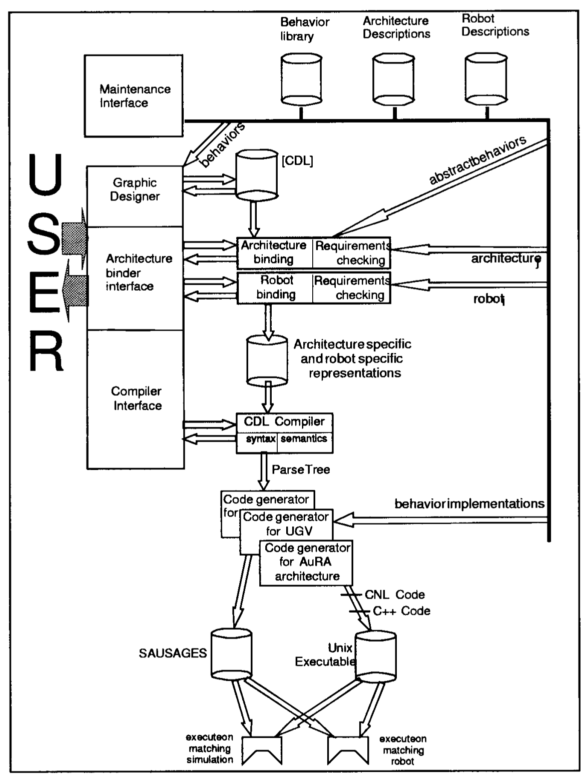

Fig. 7. The MissionLab System Architecture consists of a number of libraries and layers supporting a number of execution environments.

MissionLab [23] is a tool that has been recently developed for the testing and deployment of schema-based reactive controllers for autonomous robots. It incorporates a graphical user interface, reusable software libraries, a simulation facility, and the capability to download executable robot code for a range of real mobile platforms. MissionLab serves as the testbed for the results in this project. The architecture of MissionLab is shown in Fig. 7.

D. Integrated Architecture

In order to enable the described schema modeling, the three architectures: ASL, NSL and Missionlab, were integrated under a single system environment. ASL was first integrated to NSL [10], and then the ASL/NSL system to MissionLab [24]. The integrated architecture is shown in Fig. 8.

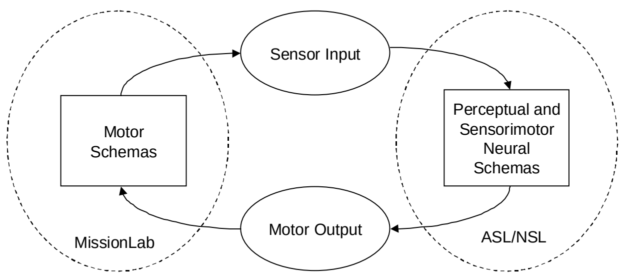

Fig. 8. The ASL/NSL/MissionLab Architecture is integrated through sensor input and motor output. Sensory information is passed from the MissionLab robotic agent to the ASL/NSL perceptual and sensorimotor neural schemas, generating motor output passed in return to MissionLab motor schemas in order for the robotic agent to generate action.

Integration is carried out through binding points between ASL/NSL and MisionLab. Sensor input from MissionLab, simulated data or real world data from actual robots, is read by the perceptual neural schemas in the ASL/NSL system. Sensorimotor neural schemas in ASL/NSL generate output to the motor schemas executing in MissionLab, either in the simulated or real world.

III. COMPUTATIONAL NEUROETHOLOGY

Neuroethology, the study of the nervous system and animal behavior, has inspired a number of computational models, such as Rana Computatrix, the computational frog [1], the computational cockroach [7], and the computational hoverfly [13]. Such computational models involve a rich number of neural based behaviors, such as the Chantlitaxia, searching from a proper habitat, taken from the Praying Mantis behavior [11], as described in the ethogram in Fig. 9.

Different models are currently being developed under the ASL/NSL/MissionLab neural schema architecture. Besides the Chantlitlaxia behavior [6], we have prototyped the adaptive toad's prey acquisition behavior due to a static barrier [14], and developing a prey acquisition and predator avoidance behavior modulated by learning processes in neural networks [20].

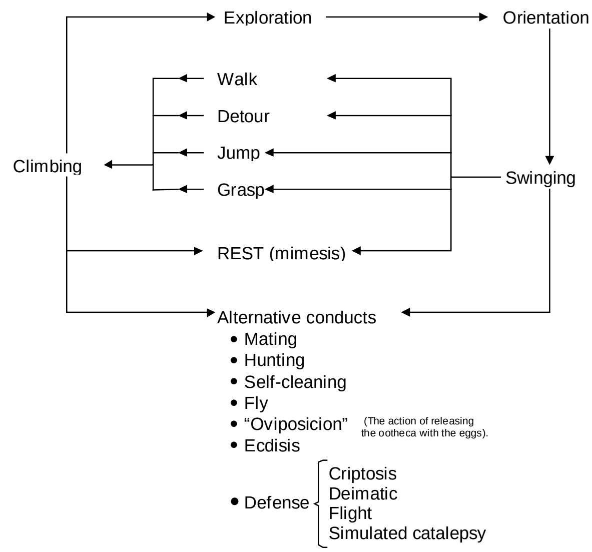

Fig. 9. The diagram shows a Praying Mantis' *Chantlitlaxia* (serach for a habitat) ethogram. This conduct is described in terms of a set of a underlying behaviors: exploration, orientation, swinging, climbing, walk, detour, jump, grasp and rest. Alternative conducts in the praying mantis are also shown: mating, hunting, etc. (See [6] for additional information.)

A. Prey Acquisition with Detour Behavior

As an example of a model developed under the neural based schema architecture we describe the toad's detour behavior due to stationary objects on its way to a prey [14]. The experiment being modeled consists of a barrier placed between a prey and a toad, as shown in Fig. 10.

Two different barrier sizes were tried, 10 and 20 cm. Both barriers are made of fenceposts, where each fencepost has a very small width, but tall enough not to have the toad jump over it. The fence posts are distanced 2 cm from each other. The toad is 20 cm away from the barrier, and the prey is 10 cm away opposite the barrier.

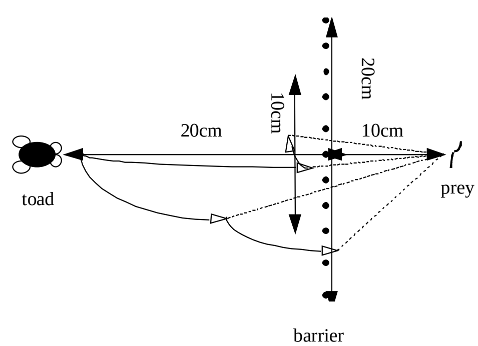

Fig. 10. Toad's prey acquisition with detour behavior experiment with a 10cm and a 20cm wide barrier. The toad is located 20cm in front of the barrier while the prey is 10cm away on the opposite side. The toad immediately detours the 10cm barrier. On the other hand the toad approaches directly the center of the barrier, requiring successive trials to manage the detour around it.

When the barrier is 10 cm wide the toad approaches directly to the barrier edges and from there continues to the prey, as shown in Fig. 10. When the barrier is 20 cm wide, the toad advances to the middle of the barrier, more precisely to the closest gap between the fenceposts. Not being able to go through the gap, the robot backs up, reorients and tries again. This adaptive process continues until the edge of the barrier is in sight. Fig. 11 shows the toad's behavior with a 20 cm barrier without and with learning. These experiments are further described in [17][15].

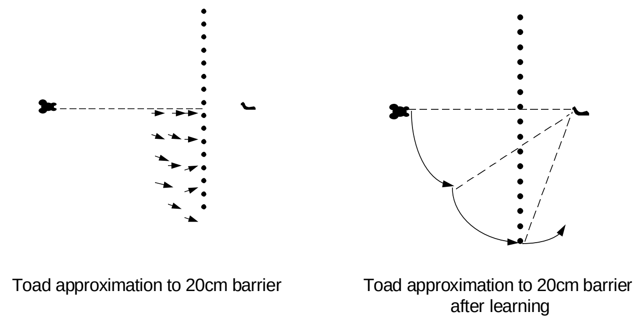

Fig. 11. The diagram shows the toad's prey acquisition model for a 20 cm barrier. On the left, before learning, the toad tries repetitively to go across the barrier, succeeding only when approaching the barrier's edge. On the right, after learning, the toad immediately approaches the barrier's edge and continues across it.

1) Schemas

In order to reproduce these experiments we developed a schema based model with a robotic agent taking the place of the toad. At the highest level, model behavior is described by means of schema specifications. At this level, the complete model is described by a network of interconnected schemas as shown in Fig. 12.

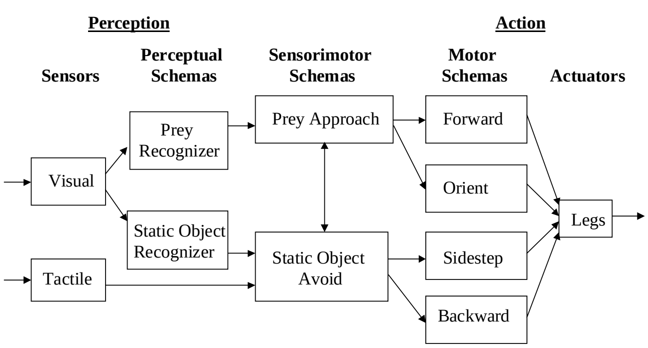

Fig. 12. The diagram shows the toad's prey acquisition with detour behavior component schemas. Schemas are organized in terms of perception and action. Perception comprises sensors (visual and tactile) and perceptual schemas (prey recognizer and static object recognizer). Action comprises actuators (legs) and motor schemas (forward, orient, sidestep and backward). The two aspects are integrated by sensorimotor schemas (prey approach and static object avoid).

The model consists of visual and tactile sensory input, perceptual schemas for recognizing stationary objects and prey moving objects, sensorimotor schemas for prey approach and static object avoidance, and motor schemas for performing forward, backward, sidestep and orient motions. Visual input is used to recognize both the static barrier and the moving prey, while tactile input is triggered when the robotic agent bumps into the barrier (not being able to go through the gap).

Rather than processing input symbols to yield output symbols, the individual schemas have *activation levels* which measure their degree of confidence. In response to the perceptual schemas input, the *more active* of the two sensorimotor schemas will *trigger* the appropriate motor schema to yield the appropriate response.

In other words, the sensorimotor schemas *compete* to control the behavior of the animal. This is a very simple example of the type of mechanisms of *competition and cooperation* that can be exhibited by a network of schemas. In particular multiple motor schemas may be coactivated to control subtle behaviors. The perceptual schemas are not simply *yes-no* recognizes, being equipped with a confidence level to provide a *parametric description* which can be used in tuning motor behavior appropriately. When the toad recognizes the prey, the animal does not respond by moving in a standard or random direction, bur rather it snaps at the position in space where the prey is located as indicated by the "prey-recognizer" schema.

2) Neural Networks

Some of the schemas in the toad's prey acquisition model are implemented all the way down to neural networks. Other schemas, for which the detailed neural circuitry is not known or involves unnecessary computation for the range of phenomena under study, are modeled in a simpler manner. For example, motor schemas in this model were not implemented through neural circuitry for simplification reasons. The neural network level implementing higher level schemas is shown in Fig. 13.

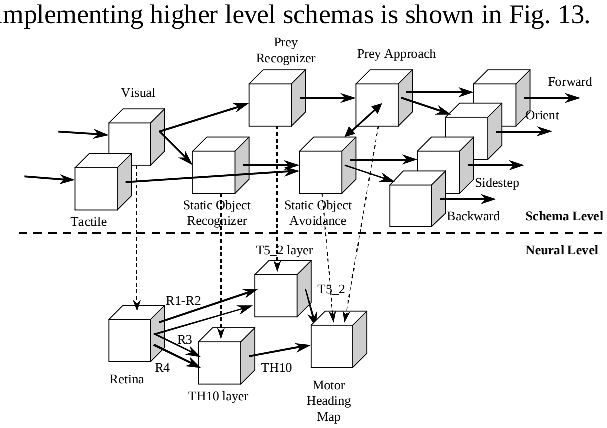

Fig. 13. The diagram shows the toad's prey acquisition with detour behavior component neural schemas. The top level consists of the original schemas. The next level down describes the mapping to neural schemas. This includes the retina, implementing the visual sensory schema; the T5_2 layer implementing the prey approach perceptual schema; the TH10 layer implementing the static object recognizer perceptual schema and the motor heading map implementing both the prey approach and static object avoidance sensorimotor schemas. Other schemas are left as general processes not implemented through neural networks.

The neural level consists of a Retina corresponding to the Visual input, T5_2 and TH10 neural layers corresponding the moving prey and static object recognizer, and a motor heading map where the static object and prey maps integrate. The motor heading map produces a target heading angle corresponding to the strongest map activity; providing inhibition between the prey approach and static object avoidance. This inhibition is important to avoid activating antagonist motor schemas simultaneously. A tactile modulation component provides adaptation to the model by increasing the inhibition repetitively, every time the robot hits the barrier. (The detailed model description can be found in [14].)

3) Autonomous Robots

The complete autonomous robotic agent is built by integrating the perceptual and sensorimotor schemas in ASL/NSL with the motor schemas in MissionLab, as shown in Fig. 14.

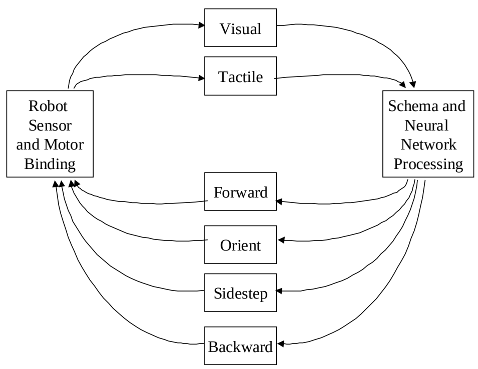

Fig. 14. At the schema level in the toad's prey acquisition model, ASL/NSL is linked to MissionLab via visual and tactile perceptual schemas and forward, orient, sidestep and backward motor schemas. Actual sensors and actuators are held in the robotic agent in MissionLab with bindings to the simulated or real robot.

The robot provides visual and tactile input to the neural schema process. These respond by producing appropriate forward, orient, sidestep and backward activations, generating robot movement. The cycle continues indefinitely, terminating only when reaching the prey. When executed in a real robot, only sensory and actuator binding is modified in MissionLab without the need to change any of the actual model details.

IV. Results

A. Prey Acquisition with Detour Behavior

The the robot (SP Frog) 20cm experiment before learning, as seen from Missionlab's simulation console, is shown in Fig. 15.

Individual schema behaviors are shown in the following figures. Fig. 16, shows the visual sensor (retina) view of the barrier as initially seen. Fig. 17, shows the resulting attraction field integrating the prey approach sensorimotor schema corresponding to prey attraction and the static object avoidance sensorimotor schema corresponding to barrier repulsion

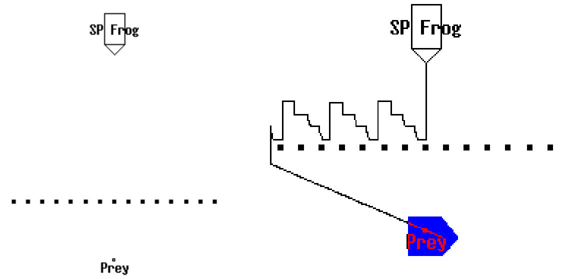

Fig. 15. The figure shows MissionLab's simulation console view of the agent's response to the 20cm wide barrier before learning. The left side shows the frog as initially positioned in front of a barrier with the prey away from it. The right side shows the resulting trajectory generated by the agent until reaching the prey.

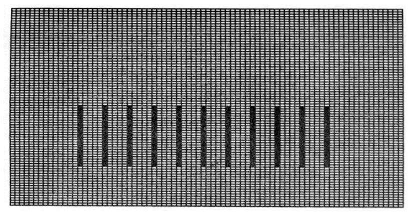

Fig. 16. Toad's visual view (retina) of the barrier. The number of elements in the figure corresponds to the number of sensor elements in the retina.

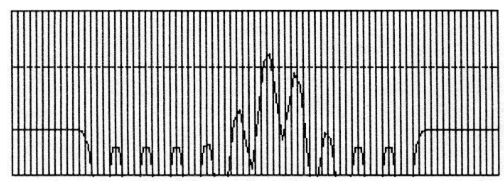

Fig. 17. Resulting attractant field from integrating the prey attraction of the prey approach sensorimotor schema and barrier repulsion of the static object avoidance sensorimotor schema. The highest value activity in the figure corresponds to the robot's preferred orientation (which initially corresponds to the prey's direction).

As the robot bumps into the barrier, the barrier's gap inhibition gets incremented resulting in a new attraction field in the motor heading map producing reorientation, as shown in Fig. 18. Every time the frog hits the barrier, it backs down and sidesteps.

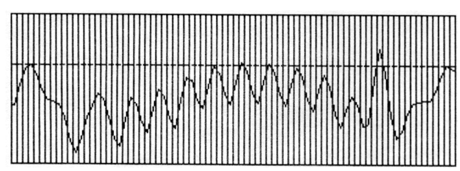

Fig. 18. Resulting attractant field from integrating the prey attraction of the prey approach sensorimotor schema and barrier repulsion of the static object avoidance sensorimotor schema after the toad bumps with the barrier. Again, the highest value activity in the figure corresponds to the robot's preferred orientation.

As the frog again tries to approach the prey, it gets attracted again and once more hits the barrier, this time on a different gap. This process continues until the edge of the barrier is in sight, as shown in Fig. 19.

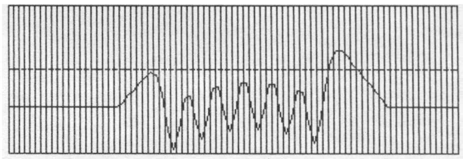

Fig. 19. Resulting attractant field from integrating the prey attraction of the prey approach sensorimotor schema and barrier repulsion of the static object avoidance sensorimotor schema after the toad bumps with the barrier. Again, the highest value activity in the figure corresponds to the robot's preferred orientation. This time the barrier gap is in sight.

From there on, the toad generates a direct path to the prey. This specific trajectory was generated due to the model reorientation specifics. Other simulated results, and more detailed results, can be found in [14].

V. Conclusions and Future Work

This paper has shown the fundamentals of the ASL/NSL/MissionLab neural schema system architecture for autonomous robots. (A previous architecture is described in [19].)

An important aspect of this architecture is the ability to incorporate adaptation and learning through neural network processes in developing new behavioral architectures for autonomous agents [16] as well as robots. The work presented here, while still in progress, goes beyond architectures where behaviors are described in terms of global states or single levels, limited in terms of neural schema integrated adaptation and learning mechanisms. As models become more complex in their nature, the distributed and concurrent nature of the integrated ASL/NSL/MissonLab architecture would become of even greater importance.

The prey acquisition model presented in this paper reproduces one of a number of behavioral experiments with toads. Other experiments are currently being tested under this architecture, in particular, extensions to the toad's and praying mantis prey acquisition and predator avoidance models as they are modulated by learning processes [17]. Furthermore, we are also in the process of experimenting with these models with actual robots in the real world [6].

VI. Acknowledgments

This research is supported by the National Science Foundation in the U.S. (Grant #IRI-9505864) and CONACyT in Mexico (Grant #546500-5-C006-A).

VII. 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., "Schema Theory", in the Encyclopedia of Artificial Intelligence, 2nd Edition, Editor Stuart Shapiro, 2:1427-1443, Wiley, 1992.

[3] Arkin, R.C., "Motor Schema-Based Mobile Robot Navigation", International Journal of Robotics Research, Vol. 8, No. 4, August 1989, pp. 92-112, 1989.

[4] Arkin, R.C., "The Impact of Cybernetics on the Design of a Mobile Robot System: A Case Study", IEEE Transactions on Systems, Man, and Cybernetics, 20:6, pp. 1245-1257, Nov/Dec. 1990.

[5] Arkin, R.C. and Balch, T., "AuRA: Principles and Practice in Review", Journal of Experimental and Theoretical Artificial Intelligence, 9:2-3, pp. 175-189, April-Sept. 1997.

[6] Arkin, R.C., Cervantes-Perez, F., and Weitzenfeld, A., "Ecological Robotics: A Schema-Theoretic Approach", to appear in "Intelligent Robots: Sensing, Modelling and Planning", Editors R.C.Bolles, H.Bunke, and H.Noltemeier, World Scientific, 1997.

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

[8] Briot, J.-P., Gasser, L., "From Objects to Agents: Connections between Object-Based Concurrent Programming and Distributed Artificial Intelligence", IJCAI '91 Workshop on Objects and AI, 1991.

[9] Brooks, R., "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, 2:14-23, March 1986.

[10] Calderas, C., Mármol, S., "Diseño e Implementación de un Sistema Distribuido Heterogéneo para la Extensión del Lenguaje de Modelos Teórico Esquemáticos ASL", ITAM Undergraduate Thesis, CANNES-TR-96-01, ITAM, México, Octubre 1996.

[11] 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.

[12] 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. Rudomian, M.A. Arbib, F. Cervantes-Perez, and R. Romo, Springer Verlag Research Notes in Neural Computing vol 4, pp. 426-449, 1993.

[13] 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.

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

[15] Corbacho, F. Arbib, M. A., "Learning Internal Models to Detour" *Society for Neuroscience*. Abs. 624.7, 1997.

[16] Corbacho, F. Arbib, M. A. "Schema-Based Learning: Towards a Theory of Organization for Adaptive Autonomous Agents" *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, California, 1997.

[17] Corbacho, F., Khort, B., Lin, B. Nothis, A. & Arbib, M. A. "Learning to Detour: Behavioral Experiments with frogs" *Proceedings of the Workshop on Sensorimotor Coordination: Amphibians, Models, and Comparative Studies*. Sedona, Arizona, 1996.

[18] Draper, B., Collins, R., Brolio, J., Hanson, A., Riseman, E., "The Schema System", Int Journal of Computer Vision, 2:209-250, 1989.

[19] Fagg, A., King, I., Lewis, A., Liaw, J., Weitzenfeld, A., "A Testbed for Sensorimotor Integration", Proceedings of IJCNN '92, Baltimore, MD, 1:86-91, 1992.

[20] Flores, L.R., "Modulación de Tiempo Variable de la Elicitación de la Conducta de Captura de Presas en los Anfibios Anuros", UNAM Master Thesis, CANNES-TR-97-01, ITAM, Mexico, DF, Agosto 1997.

[21] Khatib, O., "Real-time Obstacle Avoidance for Manipulators and Mobile Robots", Proc. IEEE International Conference Robotics and Automation, pp. 500-505, St. Louis, 1985.

[22] Lyons, D.M., Arbib, M.A, "A Formal Model of Computation for Sensory-Based Robotics, IEEE Trans. on Robotics and Automation", 5:280-293, June, 1989.
[23] MacKenzie, D., Arkin, R.C., and Cameron, R., "Multiagent Mission Specification and Execution", in Autonomous Robots, 4:1, pp.29-52, Jan.1997. (Also appears in Robot Colonies, ed.R.Arkin and G.Bekey, Kluwer Academic Publishers, 1997.)
[24] Olivares, R., "Integración del Ambiente de Simulación de Robótica Neuroetológica: NSL/ASL/MissionLab", ITAM Undergraduate Thesis, CANNES-TR-97-02, ITAM, México, DF, Septiembre 1997.
[25] Weitzenfeld, A., "ASL: Hierarchy, Composition, Heterogeneity, and Multi-Granularity in Concurrent Object-Oriented Programming", Proceeding of the Workshop on Neural Architectures and Distributed AI: From Schema Assemblages to Neural Networks, Oct 19-20, USC, October 19-20, 1993.
[26] 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.
[27] Weitzenfeld, A., Arbib, M.A., "NSL - Neural Simulation Language", in Neural Networks Simulation Environments, Editor J. Skrzypek, Kluwer, 1994.
[28] Weitzenfeld, A., Arbib, M.A., Cervantes-Perez, F., Rudomin, P., and Alexander, A., 1998, "Multi-level Simulation Methodology: A Computational and Experimental Approach to Neural Systems", in Proc. of 1998 NSF Design and Manufacturing Grantees Conference, pp 651A-652A, Jan. 5-8, Monterrey, Mexico.
[29] Yonezawa, A., Tokoro, M., Eds., 1987, Object-oriented concurrent programming, MIT Press.