

	<p style="text-align: center;"><b>ADAPT</b>  <i>IST-2001-37173</i>  <i>Artificial Development Approach to Presence Technologies</i></p>
---	---

## Deliverable Item 5.1 System's architecture

**Delivery Date:** January 06, 2004


**Classification:** Public

**Responsible Person:** Giorgio Metta

**Partners Contributed:** ALL

**Short Description:** The main objective of this document is to provide a set of design principles that allow our research into developmental robotics, developmental psychology, and cognitive science to coalesce into the design of a complete system's architecture that can be embodied into a physical system (a robotic platform) capable of interacting with the external environment. The system's architecture should provide a medium for an intentional process (as extensively defined in D2.1) to 'emerge' from a loop between motor actions, epigenetic development of multi-modal representations of visual, auditory, and haptic sensations, and the use of these representations to derive new motivations.

This deliverable is intended to be the link between the theory of intentionality as outlined in D2.1 and the actual implementation into the robotic platform. In practice, the forthcoming D5.2 and D5.3 will include an ever growing amount of details on the architecture as experimentation progresses.

	<p style="text-align: center;"><b>Project funded by the European Community under the "Information Society Technologies" Programme (1998-2002)</b></p>
---	---

## Contents

Types of learning .....	3
Self-supervised learning.....	4
Role of reinforcement learning .....	6
Pattern acquisition and feature maps .....	7
Summary .....	9
References.....	9

## Types of learning

In analyzing the type of learning involved in designing a complex architecture, it seems useful to discuss various techniques that are part of the common practice in machine learning. We can, to a certain extent, distinguish three separate parts corresponding to three different types or modes of learning:

Type of learning	Machine learning correspondence	Brain areas/mechanism
Classical conditioning	Self-supervised learning	Cerebellar system
Value learning (motivational system)	Reinforcement learning	Dopamine (and other neuromodulators) system, basal ganglia
Feature extraction	Unsupervised learning	Cerebral cortex

Although, certainly this is very gross subdivision, especially when identifying the areas of the brain involved, it offers a useful schematization and ground for discussion. Also, as introduced in D2.1, there's a distinction to be made between what is innate (phylogenetic) and acquired (epigenetic). This is summarized in the following table:

Complete Agent			
	Self-Supervised Learning	Reinforcement Learning	Unsupervised Learning
Phylogenetic modules	Innate motor programs	Innate values	Tabula Rasa
Epigenetic modules	Extended motor programs	New motor programs	Feature Extraction

where the three different modes of learning are emphasized. Supervised learning as such doesn't have a role into the design, the reason being that there's no place for a learning mode where the exact "solution" has to be provided (we should be asking who provided the solution then). In a slight variation supervised learning is still viable (and effective). This variation is called self-supervised learning (SSL). The way it works is by directly acting on the process that generates the training data samples and thus it closes the loop between data collection, learning, and further data collection (after learning). The learning part is strictly speaking supervised, and any algorithm that solves the problem (function approximation) can be employed. An example of this approach is feedback-error learning. Self-supervised learning is extremely effective for learning sensori-motor coordination tasks, and it doesn't require anything so esoteric to become implausible as a mechanism for acquiring sensori-motor coordination in the brain. Self-supervised learning is clearly bound only to learn what has been designed for.

On the other hand, reinforcement learning (RL) can be more flexible in determining such goals (and pass them to the faster SSL). Thus, in our architecture RL is considered as a mechanism serving two goals:

- Using innate values (a set of) to determine the goals for SSL and consequently learn new motor programs.
- Learning new values (motivations) to expand the robot (or biological agent) “skills”: i.e. expanding the range of experiences and behaviors the robot “likes” (is motivated) to repeat.

The unsupervised learning process is used to serve the self-supervised and reinforcement learning processes via incrementally learning an appropriate set of features that can be used to augment the innate set of motor synergies. Specifically, unsupervised learning is used to “quantize” the robot’s world into fragments of some meaningful use (a vector quantization procedure on a vast space). This includes determining visual features or motor variables (and chunks of the big state space where the robot lives) that are used in learning behaviors and/or in learning new motivations. The quantization procedure determines the relative importance of regions of the spectrum of the signals that are used in building for instance sensori-motor coordination (e.g. ICA or other subspace methods). These vectors are in practice used in mapping sensory data into motor responses; they simply represent an efficient coding of sensory and motor spaces for the purpose of building functional mapping between them. This component can be seen as learning from *tabula rasa* and is akin to the cerebral cortex.

## Self-supervised learning

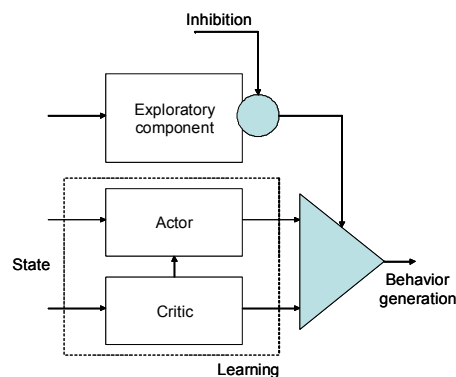
Grossly speaking, autonomous learning requires a slightly different approach from classical supervised paradigms where data is pre-segmented by hand and simply fed into a function approximator. Autonomous learning is perhaps closer to reinforcement learning in that it requires action and proper behaviors (exploratory) to gather the training set. Necessarily our architecture will require bootstrapping behaviors supporting building the training set. The question of how much explore and how to get quickly to a solution is an open one in reinforcement learning and unfortunately reinforcement learning itself tend to be difficult, requiring a very large number of samples (as we will see though RL has a role in the architecture). In addition, in the case of a real robot we shouldn’t allow “spurious” or random control values to get to the low-level controllers; at the basis of any control strategy we should probably have a reasonable “safe” explorative procedure and certainly not a complete random one. Self-supervised procedures can be identified (similar in spirit to feedback error learning) and given the appropriate amount of exploration they can quickly approximate the desired sensorimotor coordination pattern.

When data samples are available in sufficient number with respect to the size of the parameter space of the function approximator of choice the system can start learning and using what has been learnt up to date; necessarily in the long run the influence of explorative behaviors should be reduced. At least two possibilities exist here: learning could be implemented either in batches or fully online. The specific strategy is mostly a function of the algorithm and specific implementation of the function approximation. Inhibition or a functional equivalent should take care of reducing or mixing up exploration with actual “exploitation” of the acquired behavior.

Our discussion is only focused here on the function approximation problem since a good part of the sensorimotor behaviors can be actually well implemented by mapping sensory values onto motor commands or the opposite or even by a combination of the two (e.g. feedback error learning or distal learning).

Another constraint on the design of explorative behaviors is that they should mostly “explore” the space that will be used in the future. Needless to say that failure to do so might result in very poor performance.

The learning algorithm can be conceptually divided in two parts: the one providing the “learning signals” sometimes called the “critic”<sup>1</sup>, and the one doing the behavior called the “actor”. This distinction is important in motor control problems since the actor must be extremely fast and should work in a small delay regime. On the other hand, the critic could take even seconds or minutes to process the training data and provide less frequent adjustments to the actor’s parameters. We maintained as much as possible (apart from trivial cases) this distinction within our system. This division is to some extent compatible with biological mechanisms of learning being these for example the rates at which synaptic changes and growth processes develops in the brain compared to actual spikes’ travel times.



**Figure 1: A module for learning sensorimotor coordination.**

Figure 1 sketches the modules required for each actual behavior acquisition. At the moment of writing we have only conducted a few experiments with the combination and definition of modules presented here. Examples of explorative components are (at the moment) bounded random behaviors (used when training the hand localization map) or early muscular synergies (simulated muscles of course) connecting and generating activations of muscles spanning different joints and even different limbs. In learning reaching, these synergies can be exploited to bias the exploration space and avoid random movements. Whenever learning relies on multiple cues, such as visual and motor, having an initial coordination (although imprecise) can be advantageous. One net effect would be the reduction of the learning space to be explored before getting to a reasonable behavior. This strategy was used in our previous work (see G.Metta, G.Sandini and J.Konczak. *A Developmental Approach to Visually-Guided Reaching in Artificial Systems*. Neural Networks Vol 12 No 10 pp. 1413-1427 (1999)).

The actor and critic modules in our experiment consisted of a simple batch learning back-propagation neural network. Although, not the best, it proved to be very reliable so far. Back-propagation has been extensively tested and its behavior very well characterized in literature. Consequently, it is much easier to understand especially when things do not go as expected. The implementation maintains the separation of actor and critic to the point of having a slow

<sup>1</sup> The use here of “actor” and “critic” is slightly different than in the RL literature. The actor is the part of the module that actually computes the control values given some input (i.e. it’s a function mapping inputs into outputs possibly in a complicate way). The critic is whatever machinery observes the actions taken by the actor (for an arbitrary period of time) and judges the quality of the actor’s parameters in achieving a certain control goal. An explicit value function (or Q-function) is not necessarily represented in this view.

batch learning method as critic, and a distinct process providing the behavior. Naturally, given the overall robot architecture, the two modules can be even running on two different machines. Inhibition and the control of activation and coordination of many behaviors is still argument of further research and no definite implementation has been reached yet. Figure 2 shows the combination of many blocks of this type. In this case too, the realization is completely hypothetical since testing has not been performed yet.

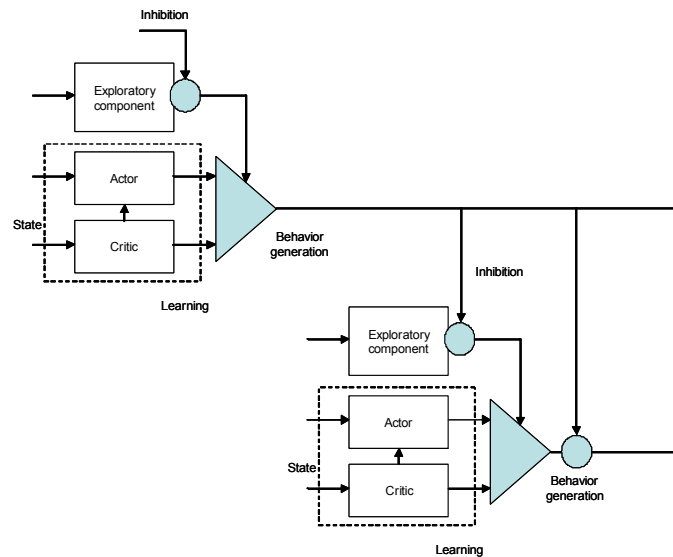


Figure 2: The combination of learning modules in a hypothetical subsumption arrangement.

## Role of reinforcement learning

The self-supervised modules are fine as long as a “target” behavior is somehow provided to the system. This target could be loosely specified as for example “to zero the retinal slip in stabilizing vision” (e.g. vestibulo-ocular reflex and similar) or “to zero the distance between hand and object in reaching”. Also, the exploratory component provides the basis for gathering training data online (as discussed above). All this is fine but unfortunately too specific and not extremely flexible. To some extent, these specifications can be seen as part of the phylogenetic inheritance of the individual and it is possible (and plausible) to engrave the acquisition of certain behaviors at this level. Examples are learning to coordinate eye movements with those of the head and trunk, learning to attend to objects, and learning to coordinate with the movement of the arm and hand.

For more complicated behaviors it could be impractical to specify by hand their “working set-point” especially if the number of behaviors is supposed to grow during ontogenesis. Also, it would make sense (and it’s more along the lines laid on D2.1) to have the agent/robot discover what is important on the basis of a value and/or motivational system. This can be achieved exactly by reinforcement learning.

Our idea here is to have the value system discover what is “pleasurable” (or bear a certain value) for the robot and derive the set-point (akin to “zero the retinal slip” kind of specification) for a self-supervised module.

In addition RL at this level should not deal with the details of the specific controller (which are taken care of by the self-supervised module) and it can be rather tuned to explore the complex space of possible set-points (it should answer to questions like: is it better to have zero retinal slip or rather 1.5deg/sec retinal slip?).

One of the values that we are going to consider is novelty. Our purpose is to design a curious robot that actively seeks new experiences and tries new things. Curiosity is important in guiding exploration in a direction which maximizes the potential for learning. Curiosity can be accomplished by measuring the novelty of the sensory inputs and using the measured novelty to compute a value signal which is used by reinforcement learning to steer behavior.

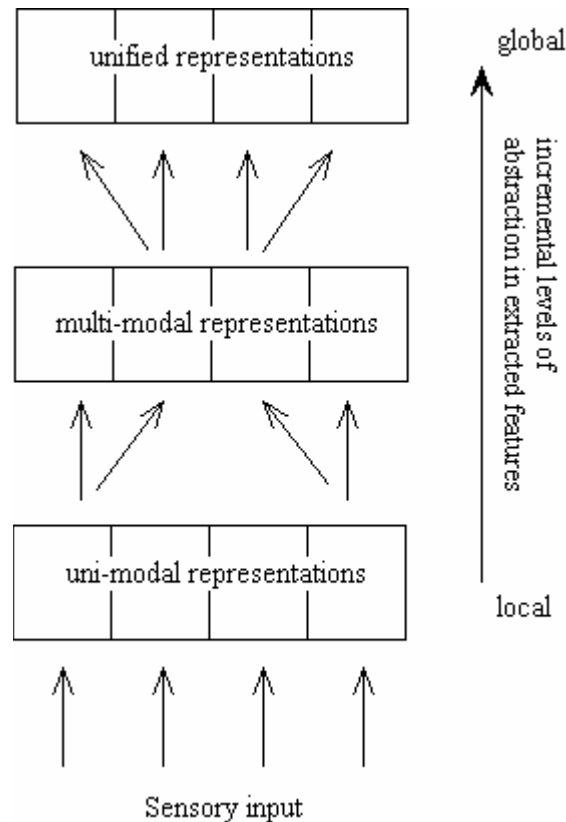
A well-known problem of reinforcement learning is credit assignment. The delay between an action and the consequent reward may be long and several other actions may have been taken in the meanwhile. Temporal difference (TD) learning solves this problem by replacing the immediate reward signal by an internally generated value signal which reflects the system's own assessment of the quality of the situation. More precisely, the value signal is the temporal difference of the future expected reward. During learning, the robot is able to find new values which are good predictors of future rewards or previously learned values.

TD learning is supported by biological evidence stating that the activity of midbrain dopamine neurons is very similar to the reward prediction error of TD reinforcement learning models (Sutton, 1988), (Sutton and Barto, 1998; 1990), (Bertsekas and Tsitsiklis, 1996). Experimental evidence and simulation studies suggest that dopamine neuron activity serves as an effective reinforcement signal for learning of sensorimotor associations (Montague et al. 1996), (Schultz et al. 1997), (Suri, 2002), (Daw, 2003), (Daw et al. 2002).

## **Pattern acquisition and feature maps**

This section outlines the unsupervised learning component of the system's architecture, which will be termed a "feature extractor". The feature extractor is an organized hierarchy of feature maps used for the derivation of new ontogenetic representations. Such representations are composed of invariances extracted at increasing levels of abstraction from multi-modal sensory input data. The acquisition of patterns refers to the learning of features at various levels of abstraction. The self-supervised learning and reinforcement learning components of the system architecture are concerned with the learning of motor synergies. It is the task of the unsupervised learning component to serve the self-supervised and reinforcement learning processes via incrementally learning an appropriate set of features that can be used to augment the set of motor synergies that the system architecture is initialized with.

The structure of the feature extractor is to be a hierarchy of feature maps where each map contains an organized structure of ontogenetic representations that develops over the course of learning. The mechanisms for how the hierarchical structure of maps and the ontogenetic representations within them will be developed, will extend previous research in hierarchically structured learning models such as neocognitron (Fukushima and Wake, 1991), (Fukushima, 1980; 1988), (Fukushima and Takayuki, 1983).



**Figure 3: The purpose of the feature-map hierarchy is to build new ontogenetic representations at increasing levels of abstraction. It is important to note that there are no clearly defined boundaries between the uni-modal, multi-modal, and unified representations depicted in this schema. These types of representations will exist as part of a continuum of representations and will be defined according to their level of abstraction in the hierarchy. In terms of schematically describing the hierarchy of representations, a single sensory modality is directly extracted from sensory input, a multi-modal representation consists of features extracted and combined from many different modalities, and a unified representation denotes the highest level of abstraction in that many multi-modal representations are combined.**

The goal of the feature extractor is to learn new ontogenetic representations based upon regularities in features learnt from sensory input data. As illustrated in Figure 3, such representations will be contained within a feature map and the level of abstraction ranges from uni-modal to unified representations. Also illustrated in Figure 3, the lowest level of feature maps in the hierarchy accepts and processes a multitude of sensory inputs. As invariant features are extracted from the input, data connections will be made to higher level feature maps in the hierarchy, in order that features extracted at lower levels can contribute to higher levels of abstraction in the representation. In Figure 3, the attainment of unified representations is described as an incremental process where learnt invariant features are used at different levels of abstraction in the formation of higher-level representations. Note that the term ‘local’ denotes the lowest level representation, that represents only a single sensory modality, whereas, the term ‘global’ denotes the highest-level representation, that represents a unification of many sensory modalities. Also, note from the same figure, that in the construction of higher-level representations, extracted features from a lower level representation converge; whereas, from the perspective of a single high-level representation the extracted features used in its construction diverge in their correspondence to multiple lower level representations.



## Summary

Learning in our architecture relies on three complementary, relatively separate modules, SSL, RL, and UL that operate concurrently. Self-supervised learning is the main mode of motor learning. Reinforcement learning augments its capabilities by assembling new motor programs which lead to reward. Unsupervised learning serves the needs of the other modules by extracting features which are needed by the controllers and value predictors learned by the other modules.

A long-term driving force of learning is the interplay between novelty-seeking motor system and the unsupervised learning module which learns expectations. Curiosity is accomplished by using novelty as a value for reinforcement learning. Once the unsupervised learning module is able to predict the sensory inputs, the same input is no longer considered novel and the robot will find something else to study.

## References

Arkin, R. (1999). Behavior-Based Robotics. MIT Press, Cambridge, USA.

Andry, P., Moga, S., Gaussier, P., Revel, A., and Nadel, J. (2000). Imitation: learning and communication. In, Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior, editors: Meyer, J., Berthoz, A., Floreano, D., Roitblat, H., Wilson, S., pages: 353-362. MIT Press, Cambridge, USA.

Becker, S. (1993). Learning to Categorize Objects Using Temporal Coherence. In, Advances in Neural Information Processing Systems (volume 5), editors: Hanson, S., Cowan, J and Giles, L., pages: 361-368. Morgan Kaufmann Publishers, San Mateo, USA.

Bertsekas, D., and Tsitsiklis, J. (1996). Neural Dynamic Programming. Athena Scientific Publishers, Belmont, USA.

Daw, N. (2003). Reinforcement learning models of the dopamine system and their behavioral implications. PhD Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA.

Daw, N., Courville, A., and Touretzky, D. (2002). Dopamine and Inference About Timing. In, Proceedings of the Second International Conference on Development and Learning, editors: McClelland, J., and Pentland, A, pages: 271-276, IEEE Computer Society Press.

Fukushima, K. (1980). Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position, Biological Cybernetics, vol. 36(1): 193-202. Springer-Verlag Heidelberg, Germany.

Fukushima, K., and Takayuki, I. (1983). Neocognition: A Neural Network Model for a Mechanism of Visual Pattern Recognition, IEEE Transactions on Systems, Man, and

Cybernetics, vol. 13(5): 826-34. IEEE Systems, Man, and Cybernetics Society Press, New York, USA.

Fukushima, K. (1988). Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition, *Neural Networks*, vol. 1(1): 119-130. Elsevier Science Press, Amsterdam, Holland.

Fukushima, K., and Wake, N. (1991). Handwritten alphanumeric character recognition by the neocognitron. *IEEE Transactions on Neural Networks*, 2(3): 355-365. IEEE Neural Networks Council Press, Louisville, USA.

Kohonen, T. (1995). Self-organizing maps. Springer Series in Information Sciences. Springer-Verlag, Berlin.

Kohonen, T., Kaski, S., and Lappalainen, H. (1997a). Self-organized formation of various invariant-feature filters in the Adaptive-Subspace SOM. *Neural Computation*, vol. 9(6): 1321-1344. MIT Press, Cambridge, USA.

Kohonen, T., Kaski, S., Lappalainen, H., and Salojärvi, J. (1997b). The Adaptive-Subspace Self-Organizing Map (ASSOM). In, *Proceedings of the Workshop on Self-Organizing Maps*, editors: Kohonen, T., and Erkki, O., pages: 191-196. Helsinki University of Technology, Helsinki, Finland.

Kohonen, T., Bry, K., Jalanko, M., Riittinen, H., and Németh, G. (1997c). Spectral classification of phonemes by learning subspaces. In, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, editor: Olson, R., pages 97-100. IEEE Press, Piscataway, USA.

Lungarella, M. and Pfeifer, R. (2001). Robots as cognitive tools: Information-theoretic analysis of sensory-motor data. In, *Proceedings of the Second International Conference on Humanoid Robotics*, editors: Lungarella, M. and Pfeifer, R., pages: 1-10. Tokyo, Japan.

Lungarella, M. and Berthouze, L. (2002). Adaptivity via alternate freeing and freezing of degrees of freedom. In, *Proceedings of the Ninth International Conference on Neural Information Processing*, editor: Wang, L., pages: 1-10. IEEE Press, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

Lungarella, M. and Berthouze, L. (2002). Adaptivity through physical immaturity. In *Proceedings of the Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development*, editors: Steels, L., Trevarthen, C., and Weng, J. pages 79-86. MIT Press, Cambridge, USA.

Manzotti, R. (2000). Intentionalizing nature. In, *Proceedings of Tucson 2000: Toward a Science of Consciousness*, editor: Hammeroff, S., pages 34-35. Imprint Academic, Tucson, USA.

Manzotti, R. (2001). Intentional robots. PhD Thesis. Lira-Lab, University of Genova, Genova, Italy.

- Metta, G. (1999). *Babybot: A study on sensory-motor development*. PhD Thesis, Lira-Lab, University of Genova, Genova, Italy.
- Metta, G., Sandini, G., and Konczak, J. (1999). A Developmental Approach to Visually Guided Reaching in Artificial Systems. *Neural Networks*, vol. 12(10): 1413-1427. Elsevier Science Press, Amsterdam, Holland.
- Metta, G. Sandini, G., Natale, L. Panerai, F. (2001). Development and Robotics. In, *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, editors: Hashimoto, S., and Takanishi, A., pages: 33-42. IEEE Press, Tokyo, Japan.
- Montague, P., Dayan, P., Sejnowski, T. (1996). A Framework for Mesencephalic Dopamine Systems based on Predictive Hebbian Learning. *Journal of Neuroscience*, vol. 16(1): 1936-1947. Society for Neuroscience, Washington DC, USA.
- Muir, D. and Nadel, J. (1998). Infant social perception. In, *Perceptual development*, pages: 247-285, Editor: Slater, A. Psychology Press, Sussex, England.
- Nadel, J. and Butterworth, G. (1999). *Imitation in infancy*. Cambridge University Press, Cambridge, England.
- Nadel, J. and Tremblay-Leveau, H. (1999). Early interpersonal timing and the perception of social contingencies. In, *Early social cognition*, editor: Rochat, P., pages: 189-212, Lawrence Erlbaum Publishers, Mahwah, USA.
- Nadel, J., Carchon, I., and Kervella, C. (1999a). Expectancies for social contingency in 2-month-olds. *Developmental Science*, vol. 2(1): 164-174. Blackwell Publishing, Oxford, England.
- Nadel, J., Guérini, C., Rivet, C., and Pezé, A. (1999b). The evolving nature of imitation as a communicative format. In, *Imitation in infancy*, editors: Nadel, J., and Butterworth, G., pages: 209-234. Cambridge University Press, Cambridge, England.
- Nadel, J., Croué, S., and Mattlinger, M. (2000). Do children with autism have ontological expectancies concerning human behaviour? *Autism*, vol. 2(1): 133-145. SAGE publications, London, England.
- Nadel, J., and Melot, A. (2001). How clear is the four-year-olds' "clear-cut change" in understanding mind? *Cognitive Development*, vol. 15(1): 153-168. Elsevier Science Press, Amsterdam, Holland.
- Natale, L., Metta, G., and Sandini, G. (2002). Development of Auditory-evoked Reflexes: Visuo-acoustic Cues Integration in a Binocular Head. *Robotics and Autonomous Systems*, vol. 39(2): 87-106. Elsevier Science Press, Amsterdam, Holland.

Pfeifer, R. and C. Scheier (1997). Sensory-motor coordination: The metaphor and beyond. *Robotics and Autonomous Systems*, vol. 20(1): 157-178. Elsevier Science Press, Amsterdam, Holland.

Pfeifer, R. and C. Scheier. (1998). Representation in Natural and Artificial Agents: an Embodied Cognitive Science Perspective. In, *Zeitschrift für Naturforschung C: A Journal of Biosciences*, Special Issue: Natural Organisms, Artificial Organisms, and Their Brains, vol. 53(1): 550-559. Springer-Verlag, Bielefeld, Germany.

Pfeifer, R., and Scheier, C. (1999). *Understanding intelligence*. MIT Press, Cambridge, USA.

Pfeifer, R., and Hara, F. (2001). *Morpho-functional machines: the new species*. Springer-Verlag, Berlin, Germany.

Sandini, G., Metta, G., and Konczak, J. (1997). Human Sensorimotor Development and Artificial Systems. In, *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Intellectual Human Activity Support for Nuclear Applications*. Editor: Kitamura, M., pages: 303-317. The Institute of Physical and Chemical Research, Wako-shi, Saitama, Japan.

Sandini, G. Metta, G. Natale, L., and Panerai, F. (2001). Sensorimotor interaction in a developing robot. In, *Proceedings of the First International Workshop on Epigenetic Robotics*, editor: Balkenius, C., Zlatev, J., Kozima, H., Dautenhahn, K., Breazeal, C., pages: 53-60. Lund University Press, Lund, Sweden.

Scheier, C., Pfeifer, R., and Kuniyoschi, Y. (1998). Embedded neural networks: exploiting constraints. *Neural Networks*, vol. 11(1): 1551-1569. Elsevier Science Press, Amsterdam, Holland.

Schultz, W., Dayan, P., Montague, P. (1997). A Neural Substrate of Prediction and Reward. *Science*, vol. 275(1): 1593-1599. Stanford University Press, Stanford, USA.

Suri, R. (2002). TD Models of Reward Predictive Responses in Dopamine. *Neural Networks: Special Issue on Computational Models of Neuro-modulation*, vol. 15(4): 523-533. Elsevier Science Press, Amsterdam, Holland.

Sutton, R. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, vol. 3(1): 9-44. Kluwer Academic Publishers, London, England.

Sutton, R., and Barto, A. (1990). Time-Derivative Models of Pavlovian Reinforcement. In, *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, editors: Gabriel, M., and Moore, J., pages: 497-537. MIT Press, Cambridge, USA.

Sutton, R., and Barto A., (1998). *An Introduction to Reinforcement Learning*, MIT Press, Cambridge, USA.