PLANCS: Classes for Programming Adaptive Behaviour Based Robots

T. S. Dahl and C. Giraud-Carrier
Machine Learning Research Group,
Department of Computer Science,
University of Bristol,
tdahl@cs.bris.ac.uk, cgc@cs.bris.ac.uk

Abstract

This paper presents a set of classes we have developed in order to implement adaptive animal behaviours in simulated robots. The programmable learning artificial neural circuits (PLANCS) merge an emulation of the classic behaviour based subsumption architecture with a neuron based circuit interface for cognitive modeling. A number of experiments are presented to exemplify the use of PLANCS in implementation and an analysis of animal conditioning is included as an example of PLANCS based, class level solution modeling.

1 Introduction

Animal and human learning is successful only because it is strongly biased by a vast behavioural context. We have abandoned traditional top-down machine learning (ML) methods for behaviour learning because of their inherent difficulties with expressing learning biases and background knowledge. In order to provide robots with high level learning abilities that correspond to those of animals and humans, it is necessary to implement equivalent amounts of context dependence in these artificial systems.

Recent work has argued that different forms of human and animal learning are not a different manifestations of a single adaptive mechanism, neither are they a collection of independent mechanisms. Instead they form an evolutionary hierarchy from habituation to language acquisition, Moore (1996). In the spirit of that work, we have taken a bottom-up approach to behaviour learning by implementing increasingly complex adaptive behaviours through small increases to less adaptive behavioural foundations, Dahl and Giraud-Carrier (2001b). This is done in a behaviour based (BB) robotics framework, Arkin (1998) using the Webots Khepera robot simulator [1].

In Section 2 we present PLANCS in detail. Section 3 presents our experiments and their use of PLANCS. Section 4 presents a PLANCS based analysis of conditioning. Section 5 draws conclusions from the experiments and the analysis and Section 6 places PLANCS in a formal ML theory framework. Finally, in Section 7 we discuss future work.

[1] A trial version of the simulator is available from the Cyberbotics homepage at: `http://www.cyberbotics.com`

2 The PLANCS Classes

2.1 Origin

During our work on a BB controller for the second online ALife Creators Contest, Christensen (2000), we developed a set of classes that expanded a subsumption like BB architecture called Edmund, Bryson and McGonigle (1997). The Edmund architecture is based on a hierarchical control system of drives, competences and fixed action patterns, all implemented as separate objects. The PLANCS classes extend this architecture by adding concurrent execution for all the objects and by widening the scope for cognitive modeling to include a neural level.

While studying adaptive animal behaviours and the evolution of learning, it became clear that there is currently a plethora of theories about the mind that are ready to be tested in an AI setting. Many of these concern different areas of the brain and their functionality. In order to take inspiration from these theories, we needed to do cognitive modeling on a neural level but also on a superneural or neural circuit level.

2.2 The Layered Structure of PLANCS

The PLANCS architecture models neural circuitry on a number of increasingly structured levels. It is visualised in figure 1 and explained in the paragraphs below.

2.3 The Network Layer

The bottom layer of the PLANCS architecture, the network layer, deals with thread initiation and synchronisation. This layer abstracts away the underlying processor structure and facilitates porting programs written using

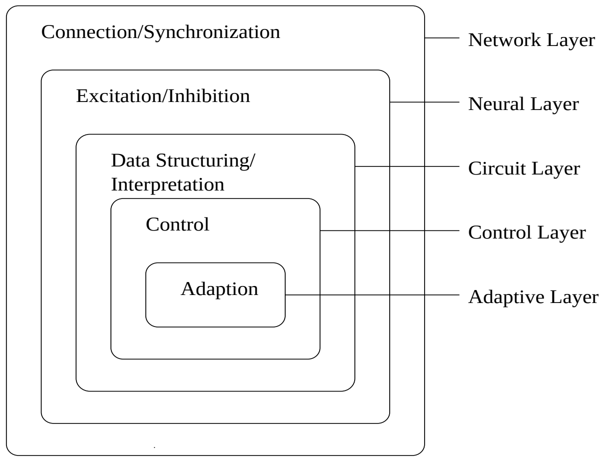

Figure 1: The Layers of PLANCS Classes

PLANCS to the distributed architectures that are common in robotics.

The original version of the subsumption architecture was implemented in lisp on a network of finite state machines running on a set of off-board processors that communicated with a uni-processor robot over a radio link, Brooks (1995). One of the main motivations behind it was the additivity of processing power that was achieved through concurrent behavioural layers with low bandwith communication.

Our neural circuit threads correspond to the off-board processors of the original subsumption architecture, but to keep the additivity properties, we needed to restrict the thread communication. How we do this is further discussed in section 2.5.

2.4 The Neural Layer

On top of the network layer sits the neural layer. The task of this layer is to handle information about the neuron level connections between circuits and the circuit's exitatory state. Every circuit has a set of inputs and a set of outputs and continuously keeps track of how many of the input circuits are currently firing. It also allows the circuits connected to the outputs access to the exitatory state so that they can do the same.

By adding a minimal control layer on top of this layer, a circuit can implement a sum-threshold neuron which fires whenever a minimal number of input neurons fire. It is possible to model natural brain processes using a collection of these basic circuits, but neuron level knowledge of natural algorithms is rare. In addition, a large number of threads would be required, something that would add a large computational thread management overhead and make such implementations inefficient. Our classes are meant for circuit level cognitive modeling, as described in section 2.5.

2.5 The Circuit Layer

Whereas the neural layer is an abstraction of a single neuron, the circuit layer is an abstraction of a collection of neurons. This is reflected in a more complex interconnection structure. Neural circuits are allowed to pass objects between each other as long as these objects are seen as an abstraction of the axons that interconnect the circuits.

The motivation for this layer is that we try to model natural algorithms on a circuit level. There are many theories about the functionality of different areas of the brain that do not include neuron level detail, Carlson (2000). We wanted to model these theories in a BB framework and a concurrent neural circuit framework was the natural synthesis.

The guidelines for BB robotics, state that communication between computational nodes should be specified down to the wire that interconnects them, Brooks (1991). The neural circuit abstraction sticks to this rule in a cognitive modeling context, but in a programming context allows structured data to be passed between nodes along unspecified communication channels. The type of the data passed between circuits is an abstraction of the set of interconnections between the neurons in the circuits and the way the receiving circuit interprets these, e.g. an approach feeder drive circuit takes as its input a feeder percept object, containing information about a feeders horizontal placement in the visual field and its proximity, calculated from its relative size. This information could be transmitted by connecting all the binary green sensitive pixels in a vision circuit (retina) to the approach feeder sense and let the approach feeder sense calculate the data. In nature however, a lot of processing is done in the visual cortex, Bruce et al. (1997). We reflect this in our implementation by having a feeder sense circuit that constructs a feeder percept object from the raw data. The passing of a percept object between the feeder sense and the approach drive reflects an underlying collection of axons from neurons in the feeder sense that reflect the results of the calculations and to the neurons in the approach drive that interpret these connections.

Connection Classes A circuit inherits from a number of connection classes according to what kind of connections it accepts and provides. The approach drive is a member of a receive feeder percept class, indicating that in a neural model, it should receive a collection of input axons, that it will interpret as a feeder. The class that provides the feeder percept, the feeder sense, inherits from a feeder percept provider class.

The ability to accept and provide percept objects and the corresponding neural connection model are part of a circuit's class and hence fixed. The choice to make the communication structure a class property reflects the fixed nature of the underlying neural connections and our commitment to the principles of BB programming.

Another common pair of connection classes is the integer receiver and provider. In a neural model, integers simply model an aggregate of an array of binary neural connections.

2.6 The Control Layer

The control layer does not deal with interactions between circuits, but with the production of outputs from the inputs.

Our classes provide a number of default algorithms for the control layer, but also the opportunity for new specialised algorithms to be programmed in Java or C++.

When developing control algorithms for specialised circuits, we kept the analogy to the massively parallel algorithms of the brain as close as possible. Thinking about the input, output and control algorithm as an abstraction of a collection of neurons helps produce a natural division of tasks between circuits as well as practical objects for inter-circuit communication.

One of the most interesting default control algorithms is the one used in our associative memory circuits, this specialised behaviour is the basis for the learning layer.

2.7 The Learning Layer

Our use of explicit memory circuits stems from a view of memories as traces or echoes of underlying neural activity, Johnson and Hirst (1993). We wanted to implement memory as a layer of neurons interconnecting different underlying circuitry and allowing associations to be made between different elements of different behaviours. There is nothing stopping anyone from using normal variables to store information in between excitations rather than using explicit memory circuits, and there are good arguments for doing so on different occasions. Our reason for not doing so was that we wanted to study all uses of memory in order to explore their functional and evolutionary relationships.

The different memory circuits implemented to support the different forms of learning are presented together with the experiments in section 3.

Mixing pre-programmed and adaptive behaviours or reactiveness and pro-activeness, is a complex topic and has been described as one of the main properties of agenthood Wooldridge and Jennings (1995). The use of explicit memory circuits clearly divides the reactive elements of behaviours from the adaptive elements, but also emphasises and clarifies their interactions.

3 Experiments on Adaption

3.1 A Reactive Foundation

We started our exploration by implementing a reactive controller which consisted of three simple layers, one for moving forward, one for approaching food and one for avoiding near by obstacles. The circuits of the initial reactive foundation are presented in Figure 3.

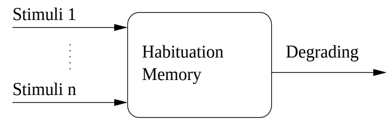

Figure 4: A Habituation Type Memory Circuit

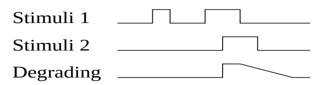

Figure 5: The Firing Pattern of Habituation Memory

The dashed arrow from the **TouchFullFeederDrive** to the **AvoidProximityDrive** indicates inhibition. The urge to veer away from things that are almost touching the robot must be inhibited to allow the robot to get in physical contact with food.

3.2 Habituation Learning

Habituation type learning takes place when repeated application of a stimulus leads to temporarily decreased responsiveness. For example, the escape response of the guppy to a shadow passing overhead decreases when such stimulus is repeatedly presented, McFarland (1999). The opposite, sensitisation, is when the responsiveness is temporarily increased as a result of the presentation of a stimuli, for example, a common octopus is increasingly likely to emerge from its home to attack a neutral stimuli after it has been fed, Macintosh (1983).

We have generalised habituation type learning to include all forms of learning that depend on a simple but quickly degrading memory of an event. We have demonstrated this kind of learning in a robot that remembers if it has just changed course in order to avoid an obstacle that stands between it and a source of food. Using the terminology above, a compensation behaviour is sensitised by avoiding an obstacle in the context of food approach.

The different neural circuits involved in our habituation learning experiment are presented in Figure 3.

Habituation Type Memory The octagonal in Figure 3 denotes a habituation memory circuit which takes inputs from a drive to approach full feeders and a drive to avoid obstacles. When these two drives fire simultaneously a continuously firing, but rapidly fading memory is established and used to activate a drive that compensates the feeder approach. The inputs and outputs of a habituation type learning circuit is presented in figure 4. Its firing patterns are presented in Figure 5.

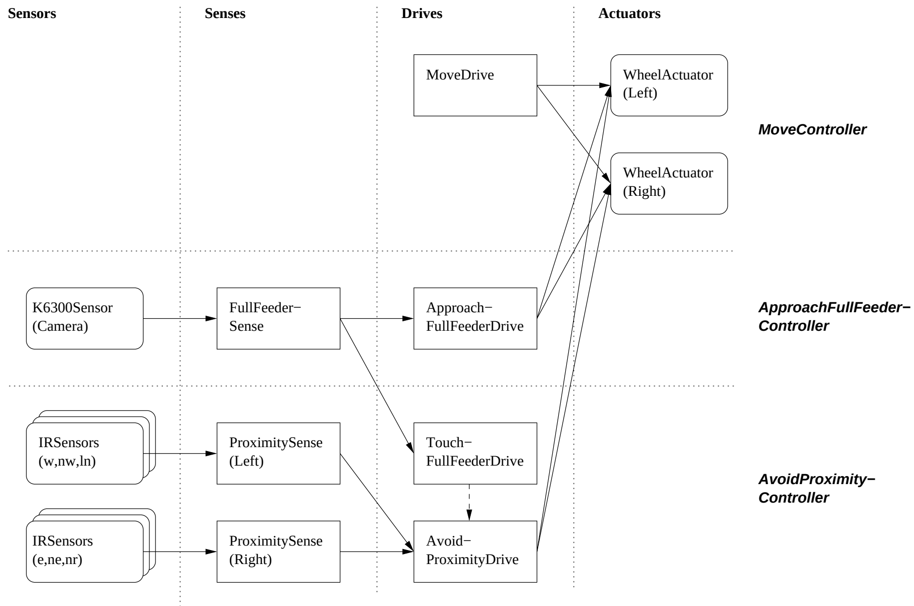

Figure 2: The Reactive Foundation

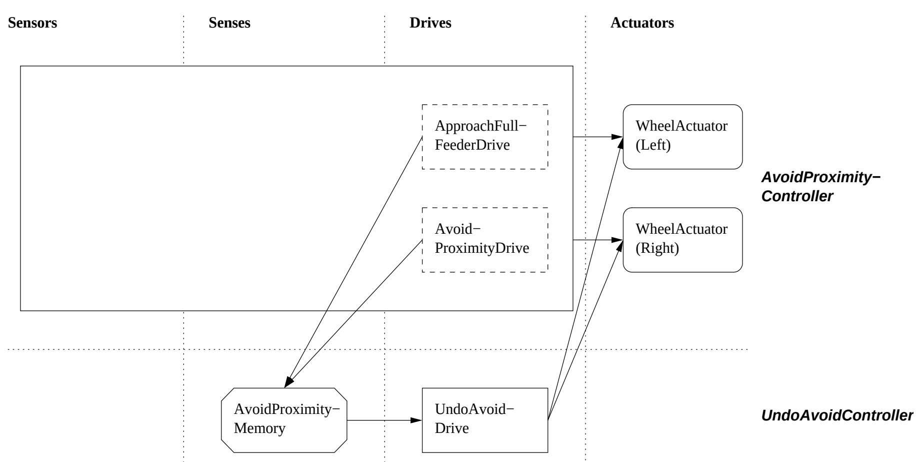

Figure 3: Habituation Type Learning

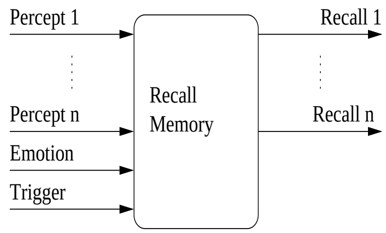

Figure 8: A Recall Memory Circuit

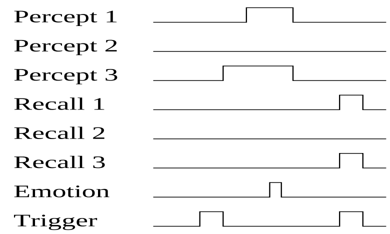

Figure 9: The Firing Pattern of Recall Memory

3.3 Navigation and Mapping

The discovery of *place cells* in rats, Muller et al. (1991), gave birth to a novel solution to the problem of spatial learning, Fuhs et al. (1998). We used this idea to present a model of spatial learning as simple association.

This model needed the presence of a spatial sense. For this, we implemented a foraging drive which explored an area by following a certain pattern, turning and accelerating according to its location. A position sense and an orientation sense provided the necessary sense of space with the position sense circuit corresponding to a set of place cells. The circuits making up the foraging behaviour is presented in Figure 6.

On top of the exploring controller, we added an associative memory circuit which remembered the state of the spatial sense every time a pleasure emotion circuit fired. The pleasure circuit fired whenever food was consumed.

We also added a new memory dependent drive which steered the robot toward the position associated with the pleasure of eating whenever the robot got hungry. This was done by using a recall memory circuit to remember the position of the feeders. The memory circuit was subsequently excited by the hunger sense and the recalled place was approached by the new drive. The circuits used to add spatial learning to the exploring controller is presented in Figure 7. The emotion circuit is indicated by an ellipse.

Recall Memory The octagonal circuit in Figure 7 denotes an recall memory circuit. This circuits stores the state of a number of percept inputs whenever the emotion input fires. It subsequently recalls the stored state every time the trigger circuit fires.

A recall memory circuit is presented in Figure 8. Its firing pattern is presented in Figure 9.

In our implementation, the percept is the place sense, the emotion is pleasure, the trigger is hunger, and the recall is the position where food was consumed.

3.4 Behaviour Recognition

Our first attempt to model a more complex form of learning with two interacting adaptive layers, was a courtship display experiment. In this experiment, two robots used a display behaviour to avoid the injuries of physical fighting. These kind of displays are common in animals and are one of the simplest forms of animal communication, Hauser (1996).

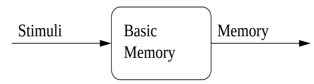

Figure 12: A Basic Memory Circuit

Reactive Interaction As a basis for more complex interactions, we implemented a reactive behaviour where a robot always tries to get in physical contact with its opponent, i.e. fight it, when is sees it close by. The circuits that implement the reactive interaction is presented in Figure 10.

Learning from Fighting The first adaptive layer was a physical fighting behaviour where the robots interact and work out which robot is the strongest by the amount of damage they take. A memory circuit is used in this behaviour to remember the pain of being the weakest robot. A memory dependent avoidance behaviour then ensures that the weakest robot avoids its opponent in the future. The circuits involved in the fighting behaviour are presented in Figure 11.

Basic Memory The memory circuit used in this layer, was named a basic memory circuit. It continuously fires after a certain combination of stimuli has been presented. A basic memory circuit is presented in Figure 12 and its firing pattern is presented in Figure 13.

In our implementation, the stimuli are the fear brought on by the pain together with the sense of another robot close by.

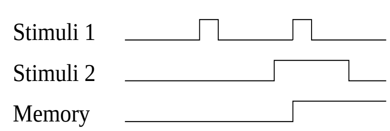

Figure 13: The Firing Pattern of Basic Memory

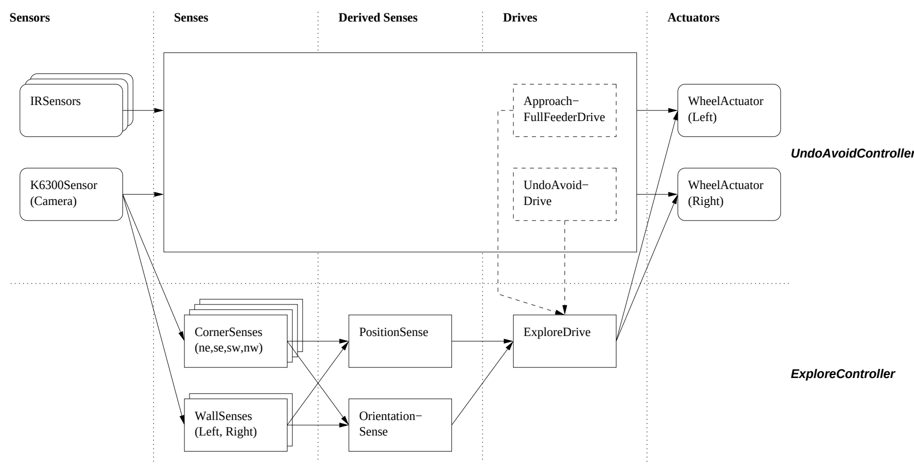

Figure 6: The Exploring Controller

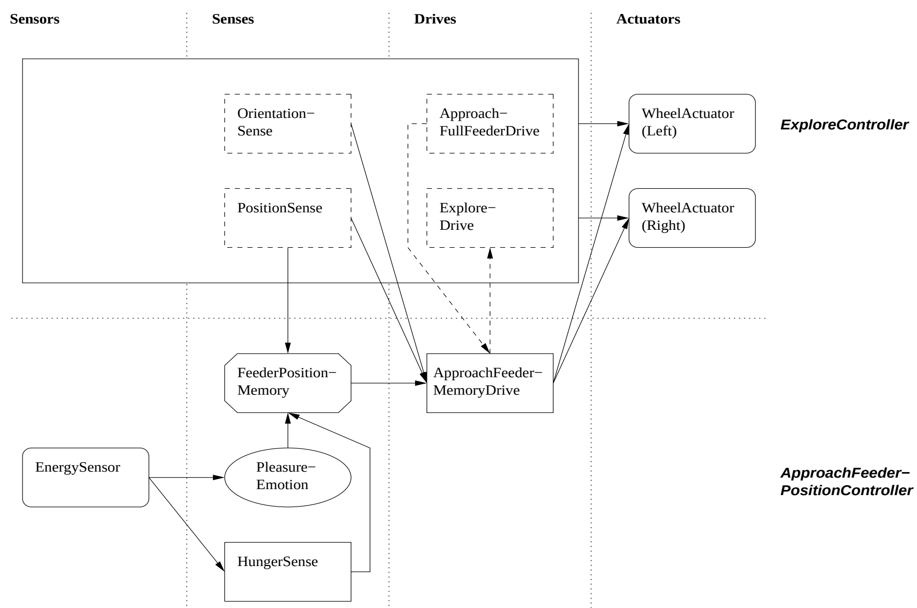

Figure 7: Learning Food Locations

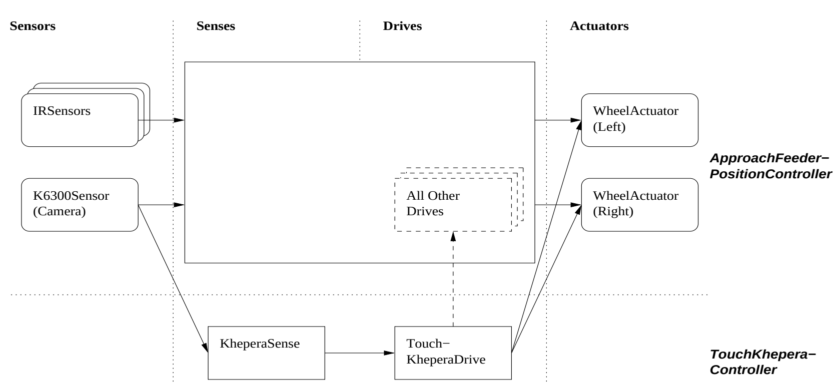

Figure 10: Reactive Interaction

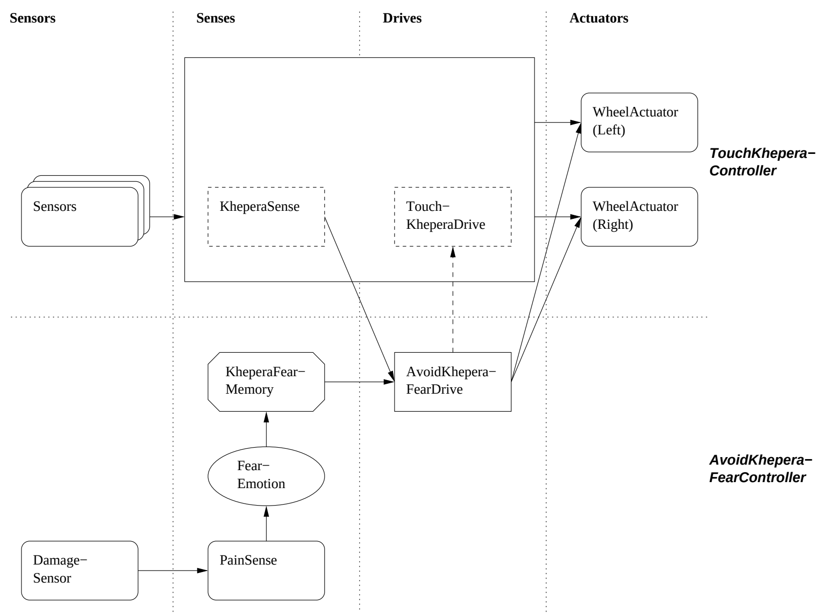

Figure 11: Learning from Fighting

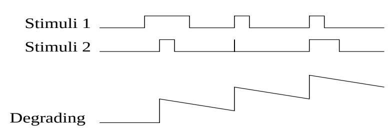

Figure 15: The Firing Pattern of Accumulating Memory

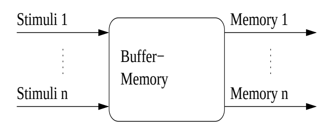

Figure 17: A Buffer Memory Circuit

Learning from Display On top of the fighting layer, we implemented a courtship display layer which took the form of a stand-off initiated by seeing the front of the opponent close by. In a stand off, the robots remain motionless for an amount of time corresponding to their strength. This behaviour also needed a memory circuit to keep track of how long the robot had been displaying. This use of memory can be described in the habituation type learning framework as increased sensitisation of a yielding behaviour.

The stand-off was over when one robot yielded and recognised the opponent as stronger. This recognition fired the fear emotion and a basic memory was created using the same circuit that was used in the physical fighting layer. The circuits used to implement the display behaviour on top of the fighting behaviour are displayed in Figure 14.

Accumulating Habituation Memory The memory circuit in Figure 14 is an accumulating habituation memory circuit. It produces an integer that increases every time the stimulus is active, and slowly degrades afterward. The accumulating habituation circuit looks exactly like the habituation memory circuit, but has a different firing pattern, presented in Figure 15.

3.5 Basic Association

What we call basic association learning is a collection of types of learning where sensory triggers are directly associated with specific motor patterns. This kind of learning was shown to be highly biased by a famous experiment by John Garcia, published in 1966 where rats would learn to avoid poison from taste but not from shock, Garcia and Koelling (1966). This kind of bias has later been referred to as the Garcia Effect or Garcia Conditioning, Moore (1996).

We show simple associative learning through a behaviour that learns to recognise food as edible or poisonous according to its colour.

Buffer Memory A problem with associating the color of a feeder with the pain of being poisoned is that at the time the pain sets in, the color of the feeder is not a part of the sensory context. In order to be able to make the association, we need to remember the color of the last seen feeder.

To do this, we use a sensory buffer memory which stores the latest value of a sense and reproduces it continuously until a new non zero value is sensed. A buffer circuit is presented in Figure 17 and its firing pattern is presented in Figure 18.

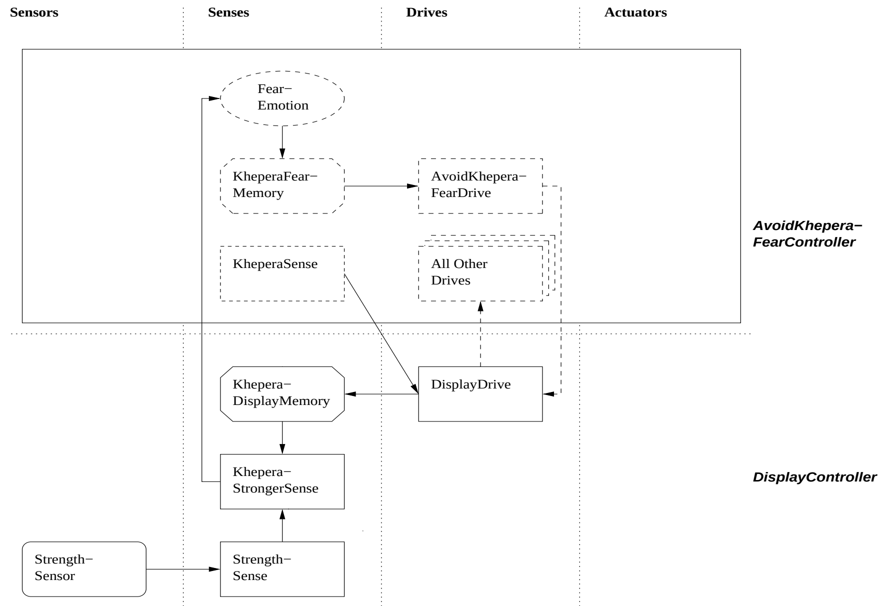

Figure 14: Learning from Display

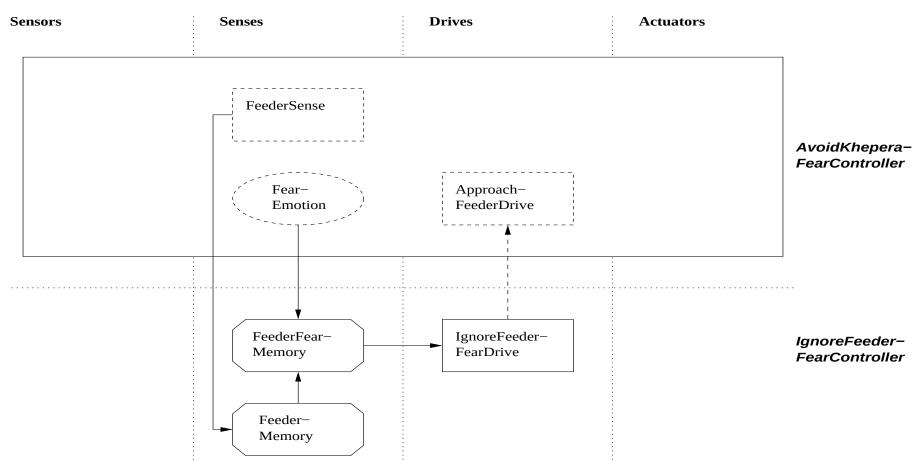

Figure 16: Learning to Avoid Poison

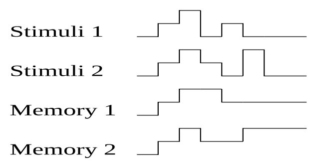

Figure 18: The Firing Pattern of Buffer Memory

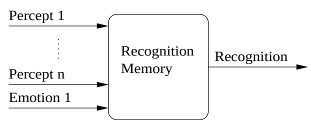

Figure 19: A Recognition Memory Circuit

Recognition Memory Circuits In order to associate the colour of the feeder with the pain of being poisoned we needed a memory circuit that takes a percept object and a binary emotion connection as inputs and produces a binary output. Whenever the emotion input fires, the corresponding percept object is stored. On subsequent presentations of an identical percept, the memory circuit will fire.

This kind of circuit is presented in Figure 19. Its firing pattern is presented in Figure 20.

4 Analysing Operant Conditioning

The experiments on spatial learning and poison discrimination were simple examples of conditioning. A type of conditioning that borders on operant conditioning is autoshaping. In the paragraphs below we look at how we can use PLANCS to support autoshaping and operant conditioning in the future.

From Conditioning to Autoshaping: The most basic form of operant conditioning is what is called autoshaping, Pearce (1997). Autoshaping is a form of conditioning where an unconditioned response (UR) is associated with a conditioned stimulus (CS) through repeatedly presenting it together with an unconditioned stimulus (US), e.g. a pigeon will start pecking at a key if the key is repeatedly presented together with food.

An experiment that would demonstrate this kind of learning is a world where food would regularly appear next to a unique object such as a cube. A robot could now demonstrate autoshaping by learning to approach cubes in the hope that some food would be close by.

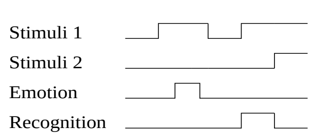

Figure 20: The Firing Pattern of Recognition Memory

In order to do this it is necessary to generalise the trigger of the approach behaviour to include a cube. The trigger for the approach behaviour was originally a full feeder percept. We suggest to generalise this old and soon to be evolutionary outdated trigger by connecting a cube sense to the approach behaviour circuitry via a recognition circuit like the one we used for poison recognition. We then connect the recognition circuit to the pleasure circuit. Another change we have to make is to let the sight of food cause the pleasure circuit to fire, not only the eating of it. This has a knock-on effect on the spatial learning behaviour, but we ignore that here. With this new setup, the robot would be able to learn to approach cubes if one was ever perceived together with a feeder.

The circuit design of our autoshaping solution is presented in Figure 21.

From Autoshaping to Operant Conditioning: Operant conditioning, also called instrumental conditioning is when a randomly produced behaviour also called operant behaviour, is related to a stimuli by an US. The typical example is a rat that is taught to push a lever by being rewarded every time its frantic behaviour leads to the level being pushed. This type of learning requires the display of *random* or *operant* behaviour. It also requires associations to be made between the randomly produced UR and the CS, the lever, thus making the successful behaviour, the lever pushing, a conditioned response (CR).

For this, buffer circuits are needed for both the stimulus and the response and the robot needs to be able to associate these through associative memory circuits. This would create a completely learned behaviour chain that is only grounded in the sensors and the actuators. This behaviour would also need to be able to inhibit the random behaviour that originally dominated this scenario.

Chaining of Conditioned Responses Experiments on operant conditioning often create complex chains of CRs that have the previous CR as their CS. This demands a double role from the action buffer memory. It has to support association from senses to actions as well as to other actions. This would take multiple response memory circuits with complex interconnections. We have not yet explored fully the implications of using PLANCS to support this kind of operant conditioning.

5 Conclusions

There are two vague evolutionary trends in the memory types we have used throughout our experiments and our analysis.

On the side of memory establishment, we can split our learning circuits into three kinds: memories that are always established, such as buffer memory, memories that

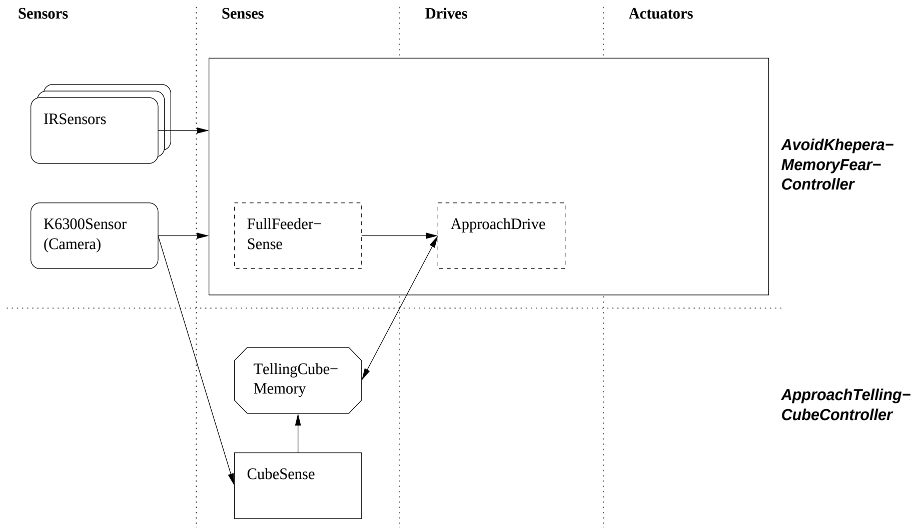

Figure 21: Associating Cubes with Approach Behaviour

are only established when the right stimuli are present but which don't need an emotion circuit, these include habituation memory and accumulating habituation memory, and finally those that need an emotion circuit to fire, which is basic memory, recall memory and recognition memory.

On the firing side, there are the continuously firing and degrading kind, habituation memory and accumulating habituation memory. There are the continuously firing, non-degrading kind, basic memory and buffer memory, the binary triggered kind, recall memory, and there is recognition memory which has its own internal concept of identity that decides whether it fires or not.

Looking at these two attributes suggests the following hierarchy according to increasing sophistication in establishment and reactivation:

1. Buffer memory
2. Habituation memory
3. Accumulating habituation memory
4. Basic memory
5. Recall memory
6. Recognition memory

This is a rough ordering based on unfounded assumptions about what is sophisticated behaviour for memory. There are other more important issues in memory circuit complexity, such as the number of inputs and outputs it allows, i.e. decreasing bias, and the complexity of the internal learning algorithm.

What seems like a fair conclusion to draw is that there are a number of very specified types of memory that support very limited forms of learning, and a number of more general types that allow more general associations. It is likely, and it seems to be supported by our analysis, that a certain level of generality in the memory circuits is needed to support the generality of adaptive behaviours such as conditioning.

It also seems safe to predict that more general kinds of learning will use even less biased memory circuits.

6 PLANCS and ML Theory

In the recognition memory circuits, the function that decides if subsequent percepts are identical to the stored percept is what is classically discussed in ML theory, Anthony and Biggs (1997).

The percepts that are presented when the emotion circuit fires constitute the set of examples and the consequent firing pattern of the memory circuit constitutes the theory produced from the examples. The other types of memory circuits are trivial in an ML sense.

The alphabet available to describe the examples in a recognition circuit are limited by the attributes of the percepts. This is a learning bias in that it is a restriction on the complete set of inputs that is available to the robot. Correspondingly, the hypothesis language is limited by the type of the output. There are however, no limitations on the algorithms that can be used to produce the outputs from the inputs, and hence no biases apart from the language biases.

Our poison recognition example creates a binary hypothesis, but it is also possible to produce theories that are numerical values or objects. Our goal is not to study the effects of different learning methods within a circuit, but to study the biases on the circuits.

The traditional top-down approach to learning minimises the language biases and tries to place all the biases in the learning algorithm. This is equivalent to having a single learning circuit that takes all the sensor values as inputs and produces outputs for all the actuators.

Our bottom-up approach tries to solve the same problem by restricting the inputs and outputs of the learning circuits to create many minimal search spaces. This means minimising the learning task in all behaviours.

On one hand, this is a sensible approach, as there is an evolutionary cost on adaption and whenever adaption is introduced in evolution it should be maximally biased. On the other hand, humans seem to have the mental machinery to drastically change their behaviour according to reasoning and social adjustment. It is not clear whether this is really true and what evolutionary advantages it might have had if it is. We believe that only by studying increasingly adaptive behaviour will we gain insight into these questions.

7 Future Work

7.1 Future Experiments

It is clear from our experiments that a number of different learning circuits are necessary to support the forms of learning we have demonstrated. There is not much scope for generalising the functionality of the different learning circuits though it might be argued that they can all be implemented on a neural level.

There appears to be more of a uniformity to the learning circuits used as we start exploring conditioning learning with its extensive need for recognition circuits.

Having had fairly diverse result in implementing low level adaptive animal behaviours, it would be very interesting to see if the apparent increase in memory circuit uniformity on the conditioning level and above is real or just perceived. Further experiments on conditioning would provide more definite answers than the analysis we presented in section 4.

An increasingly menacing problem is the poverty of our behavioural repertoire. We believe that in order to produce useful results, we need our behaviours to be as natural as possible. As we study increasingly complex adaption mechanisms, we become increasingly dependent on natural supporting behaviours to provide the necessary senses and adaptive biases. The use of a pink box sense in section 4 is an example of an artificial support behaviour. If a greater number of basic behaviours had been present, we would have been able to use a pre-existing percept as a basis for our association learning.

Having to let the sight of a feeder induce the pleasure emotion had an unexpected side-effect on the spatial learning behaviour. These kind of interaction issues though not discussed in this paper, usually provide a lot of information about the quality of the structure of the layers and also teach us a lot about behaviour interaction in general. To have used a pre-existing and more natural percept rather than the ad-hoc pink cube sense would have mirrored more closely the way the associative learning is likely to have evolved in nature and would also have been likely to have thrown up more interaction issues.

The pink cube experience and other arguments from cognitive robotics, Brooks (1997) and from the theory of evolution, Allman (1999) have lead us to formulate an approach to studying adaptive behaviour that emphasises the need to include as many different behaviours as possible on different levels of complexity Dahl and Giraud-Carrier (2001a). We have also done an initial analysis of human evolution to identify what types of behaviour are likely to have coexisted in our pre-history. We use this analysis as a guideline to how far we should develop the complexity of one dimension of behaviour before we need to include other behaviours in order to avoid artificial support circuits.

7.2 Top-Down Effects

Top-down cognitive effects are described in psychology as high level processes influencing low level ones, e.g. when word and sentence context affecting letter recognition, Johnson and Hirst (1993).

PLANCS seem to be well suited for exploring such effects, in that it allows downward connections to be made from high level layers. The top down question becomes important in a learning context when it comes to letting high level adaptive behaviours override the more basic instincts of the lower levels. We already touched on this issue in section 4 when we said that conditioned responses has to be able to inhibit the operant behaviours.

Acknowledgments

This work was supported by Hewlett-Packard, The University of Bristol, and The Norwegian Educational Load Fund. Thanks also to Cyberbotics for a free student license to use the Webots Khepera robot simulator.

References

J. M. Allman. *Evolving Brains*. Scientific American Library, 1999.

M. Anthony and N. Biggs. *Computational Learning Theory*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1997.

R. C. Arkin. *Behaviour Based Robotics*. MIT Press, 1998.

R. A. Brooks. Intelligence without reason. In *Proceedings of IJCAI 91*, pages 569–595. Morgan Kaufmann, 1991.

R. A. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA-2:14–23, 1995.

R. A. Brooks. From Earwigs to Humans. *Robotics and Autonomous Systems*, 20(2–4):291–304, 1997.

V. Bruce, P. R. Green, and M. A. Georgeson. *VISUAL PERCEPTION Physiology, Psychology, and Ecology*. Psychology Press, Third edition, 1997.

J. J. Bryson and B. McGonigle. Agent Architecture as Object Oriented Design. In *Intelligent Agents IV, Proceedings of the Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL'97), LNAI 1365*, pages 15–30. Springer Verlag, 1997.

N. R. Carlson. *Physiology of Behaviour*. Allyn and Bacon, seventh edition, 2000.

H. I. Christensen. The WEBOTS Competition. *Robots and Autonomous Systems Journal*, 31:351–353, 2000.

T. S. Dahl and C. Giraud-Carrier. Evolution, Adaption and Behavioural Holism in Artificial Intelligence. Submitted to the 6th European Conference on Artificial Life (ECAL'01), 2001a.

T. S. Dahl and C. Giraud-Carrier. High Level Learning through Incremental Adaptivity. Submitted to the 3rd International Conference on Modeling and Using Context (CONTEXT'01), 2001b.

M. C. Fuhs, A. D. Redish, and D. S. Touretzky. A Visually Driven Hippocampal Place Cell Model. In J. Bower, editor, *Computational Neuroscience: Trends in Research*, pages 101–106. Plenum Publishing, 1998.

J. Garcia and R. A. Koelling. Relation of cue to consequence in avoidance learning. *Psychonomic Science*, 4:123–124, 1966.

M. D. Hauser. *The Evolution of Communication*. MIT Press, 1996.

M. K. Johnson and W. Hirst. MEM: Memory Subsystems as Processes. In A. F. Collins, S. E. Gathercole, M. E. Conway, and P. E. Morris, editors, *Theories of Memory*, pages 241–286. Lawrence Erlbaum Associates, 1993.

N. J. Macintosh. General Principles of Learning. In T. R. Haliday and P. J. B. Slater, editors, *3, Genes, Development and Learning*, Animal Behavior. Blackwell Scientific Publications, 1983.

D. McFarland. *Animal Behaviour, Psychobiology, ethology and evolution*. Addison Wesley Longman Ltd, fourth edition, 1999.

B. R. Moore. The Evolution of Imitative Learning. In C. M. Heyes and B. G. Galef, editors, *Social Learning in Animals: The Roots of Culture*, pages 245–265. Academic Press, 1996.

R. U. Muller, J. L. Kubie, E. M. Bostock, J. S. Taube, and G. J. Quirk. Spatial firing correlates of neurons in the hippocampal formation of moving rats. In J. Paillard, editor, *Brain and Space*, chapter 17, pages 296–333. Ocford University Press, 1991.

J. M. Pearce. *Animal Learning and Cognition*. Psychology Press, 2nd edition, 1997.

M. Wooldridge and N. R. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.