Hierarchy and Sequence vs. Full Parallelism in Action Selection

Joanna Bryson (joannab@cogsci.ed.ac.uk)
Department of Psychology; 7 George Square
Edinburgh, EH8 9NG UK

Abstract

Hierarchical organization has become an unfashionable model of intelligent control within some communities of both natural and artificial intelligence. What has replaced it are models based on parallel distributed processes, both neural and behavior based, or dynamic systems theory that defies modularity, let alone rigorous structure.

In this paper we present experimental results demonstrating an artificial reactive hierarchy-based system that outperforms fully parallel systems in a highly dynamic environment with a large number of conflicting goals. This work is conducted in Tyrrell's (1993) Simulated Environment and can be seen as an extension of his work on comparing action selection mechanisms. We observe that the hierarchical strategy has also been well demonstrated in nature. We argue that preserving full reactivity for a complex intelligence may not be worth the cost in terms of complexity.

Introduction

The use of both hierarchical and fixed sequential orderings of behavior for action selection has been postulated since the time of the early ethologists (Lorenz, 1981). The advantages of such systems are clear: they reduce the combinatorial complexity of control. The set of actions which may be selected from is determined by context, including state internal to the agent. Once an agent is attending to a particular strategy, actions associated with alternative strategies are unlikely to interfere with behavior production.

Early artificial intelligence research followed these hierarchical models, but ran into great difficulty in coping with dynamic environments in real time. Action selection was based on constructive planning, in which was in turn based on heuristic search (Newell and Simon, 1981), a system both demonstrated and theoretically proven intractable without major alteration (Nilsson, 1984; Chapman, 1987). These results lead researchers both inside and outside of artificial intelligence towards a new paradigm of *reactive intelligence* (Agre and Chapman, 1990; Maes, 1991; Goldfield, 1995; Rosenschein and Kaelbling, 1995; Hendriks-Jansen, 1996). A reactive system is designed from the beginning to be situated in a complex, dynamic environment, which it must constantly monitor and to which it must instantly react.

A basic premise for many of these researchers is that a truly reactive system must have all aspects of its intelligence constantly active and sampling the environment (Hendriks-Jansen, 1996; Tyrrell, 1993; Brooks, 1991; Maes, 1991). This is associated with the *behavior based* approach to reactive intelligence, in which intelligence is composed of relatively simple modules: tightly coupled units of sensing and action. These behaviors are the perceptual system of the agent. If they are inactive, relevant information in the environment will be ignored. Some authors also reject hierarchy in a controller altogether, on the basis that it results in bottlenecks, staged responses, and governed, unreactive behavior (Hendriks-Jansen, 1996; Maes, 1991). Others design systems that exploit hierarchical order, but still maintain constant parallel processing (Tyrrell, 1993; Blumberg, 1996).

Such approaches still neglect many of the advantages of hierarchical organization described above. This paper demonstrates that an agent may neglect otherwise significant portions of its environment in pursuit of more important goals and still be very successful in a highly complex, dynamic and hostile environment. We use an established test base for reactive control, Tyrrell's Simulated Environment, and the architecture previously tested as best in these conditions, the Extended Rosenblatt and Payton Architecture (Tyrrell, 1993). We then draw on biological evidence that animals (including humans) are not fully reactive, and argue that the complexity of the intelligent system necessary to handle full environmental information makes a selective hierarchy preferential to a fully parallel organization.

Approach

This section describes the environment used in the experiments, and the architectures representing the two forms of cognitive organization for control.

The Simulated Environment

Tyrrell's Simulated Environment, the SE, specifies an action-selection task which requires an agent to manage a large number of conflicting goals. He defines an environment in which a small omnivorous animal is required to

survive and breed. He defines six subproblems for the animal to solve.

1. Finding sustenance. In addition to water, there are three forms of nutrition, satisfied in varying degrees by three different types of food.

2. Escaping predators. There are feline and avian predators, which have different perceptual and motion capabilities.

3. Avoiding hazards. Benign dangers in the environment include wandering herds of ungulates, cliffs, poisonous food and water, temperature extremes and darkness. The environment also provides various forms of shelter including trees, grass, and a den.

4. Grooming. Grooming is necessary for homeostatic temperature control and general health.

5. Sleeping at home. The animal is blind at night; its den provides shelter from predators and other hazards, and helps the animal maintain body temperature while conserving energy.

6. Reproduction. The animal is male, thus its reproductive task is reduced to finding, courting and inseminating mates. Attempting to inseminate unreceptive mates is hazardous.

The success of the animal is considered to be the number of times it mates in a lifetime. This is highly correlated with life length, but long life does not guarantee reproductive opportunities.

These problems vary along several axes: homeostatic vs. non-homeostatic, dependency on external vs. internal stimuli, periodicity, continual vs. occasional expression, degree of urgency and finally, whether prescriptive or proscriptive with regard to particular actions. In addition to these problems, the environment is highly dynamic. Food and water quantities, temperature and light vary, and animals move. Sensing and action are uncertain. Perception in particular is extremely limited and severely corrupted with noise; the animal usually misperceives anything not immediately next to it, unless it chooses to spend time and expose itself by rearing up and "looking around" in an uncovered area.

Tyrrell separates the problems of learning and navigation from the problem of action selection by providing these elements in his simulation. Thus the animal has available as primitives a direction in which it thinks it remembers its home or recently observed food and water. The animal's sense of location with respect to its den decays over time and distance, thus keeping track of its bearing is a part of the "sleeping at home" sub-problem, even though the precise mechanism of navigation is not explicit.

The results of the animal's performance is heavily dependent on chance implementation details of the environment. In his thesis, Tyrrell attempts to check the independence of his results from these details by running tests in four different "worlds". Besides the standard model he first designed, there are three variants. As documented (p. 162) these vary in the following ways.

1. Perception is altered by affecting the animal's visibility according to the time of day, the amount of vegetation, and the animal's own activity.

2. The noise variance for navigational information is tripled, and the size of the remembered map is halved.

3. Motor control is compromised by making it more probable that incorrect actions are be taken, and by changing the conspicuousness and energy consumption of various actions.

The Extended Rosenblatt and Payton Action Selection Mechanism

Tyrrell used this environment to test five different architectures from ethology and artificial intelligence: the first four were drive theory (Hull, 1943), the psycho-hydraulic system of Lorenz (1950, 1981), spreading activation networks of Maes (1991), and the connectionist, hierarchical, feedforward networks of Rosenblatt and Payton (1989). The fifth, which he recommends not only as best but as nearly optimal, is his own extension of the latter system, the Extended Rosenblatt and Payton architecture (ERP). In this model, all the constituent elements operate continuously in parallel. These elements constantly evaluate their own relevance with respect to the current environment and the animal's needs. This relevance is used to weight the element's recommendation for a next action, which is also constantly in computation. Ultimately a winner-take-all algorithm is used to determine the expressed action, which may be a combination of the desires of several elements.

Tyrrell rejects the subsumption architecture of Brooks (1986) as being too ill-defined to be considered a true architecture for action selection. Regardless of the accuracy of this criticism, the ERP shares many features in common with subsumption, such as full concurrency. The primary difference is the system of prioritization between behaviors: the ERP's is more orderly with a clear directionality between any two behaviors. Subsumption allows for cycles within a single layer, though in practice these seldom exist. Also, the somewhat complicated system of inhibition and suppression in subsumption is replaced by a uniform pipeline metaphor in the ERP, with no obvious loss of expressiveness.

Tyrrell's work and the ERP architecture in particular have been very influential. For example, Humphrys (1996) evaluates a number of strategies for final action selection, favoring a selection on the basis of maximizing the least unhappiness of the various units. Blumberg (1996) has divided his behavior elements into classes that do not need to compete for physical resources on the agent, and may therefor express themselves simultaneously, as in the joyous or woeful gaits for humans and dogs. However, the experiments in this paper compare directly to to the Extended Rosenblatt and Payton architecture. This is for two reasons: first,

because it had already been stringently tested against four other popular models of actions selection, and second, because a solution for the SE implemented by its original author already exists, removing several of the sources of bias possible in such a comparison exercise.

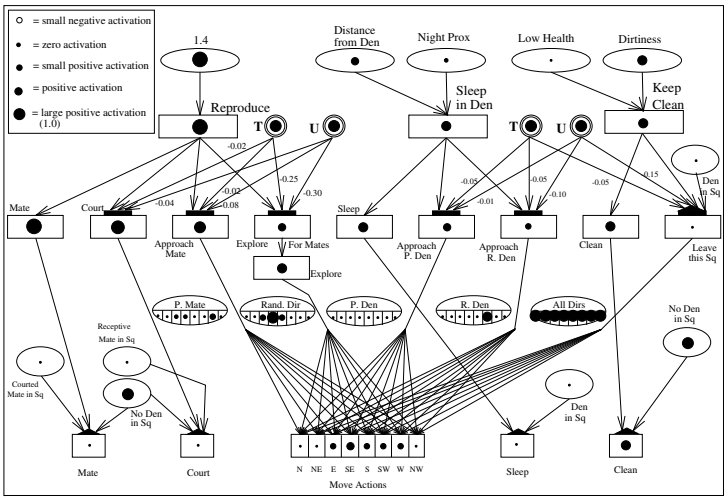


Figure 1: A fraction of the Extended Rosenblatt & Payton action selection network for controlling an animal in Tyrrell's Simulated Environment. **T** and **U** are temporal and uncertainty penalties for actions that take time or pursue a goal uncertain to succeed. P stands for perceived, R for remembered. (Tyrrell, 1993, page 247).

The Rosenblatt & Payton model was developed and first tested as a robot architecture (Rosenblatt and Payton, 1989). The main feature of this architecture is what Tyrrell terms a *free-flow hierarchy*. Nodes in the hierarchy receive activation from internal and external stimuli and higher elements of the hierarchy. They pass energy to their children. What differentiates this model from the drive model or other hierarchies is that no decisions are made until the leaf or action nodes. This allows for the selection of *compromise candidate* behaviors: behaviors that satisfy or express multiple drives. See for example Figure 1.

Tyrrell's extensions to Rosenblatt & Payton's original system are as follows:

- The addition of penalties for temporal delays and uncertainty of rewards for items that are distant or based on poor memory or perception. This allows the animal to chose a small but apparent near food source rather than a large but distant, poorly remembered food source.

- The creation of explicit penalties for appetitive vs. consummatory actions[1].

- The addition of a new rule for combining the inputs to a single node that explicitly distinguished the maximum of inputs adding positive energy to a particular node and the minimum of inputs opposing the node. This was significantly more successful than simple summation of the activations (See further Humphrys, above).

Edmund

The ERP is able to model a hierarchical structure of data flow, but it does not represent decisions or a focus of attention — normally considered basic elements of a cognitive model. This reflects one of the strongest beliefs in the agent community, that traditional hierarchical structure is overly rigid and leads to neglect, and must be replaced by more dynamic, fully parallel architectures (Maes, 1991; Brooks, 1991; Hendriks-Jansen, 1996). The reason Tyrrell argues that the ERP is a near optimal architecture is due to its ability to monitor all goals simultaneously, thus enabling it to select more optimal actions in particularly contentious situations. He emphasizes the idea of *compromise candidate* actions, which satisfy more than one set of goals.

The argument of this paper is that an adequately reactive hierarchy can surpass the performance of a fully reactive system, despite the loss of information and compromise candidates. The simplification in control structure that results from taking advantage of selective attention makes an intelligent approach easier to develop, whether by natural evolution or by human design.

The experiments in this paper use an agent architecture called Edmund (Bryson and McGonigle, 1998)[2]. Edmund is a behavior-based architecture with parallel ordinal slip-stack hierarchical (POSH) action selection. Under Edmund, the behavior modules represent a higher level of abstraction than the simple control primitives they have come to represent in architectures such as the ERP, RAPS (Firby, 1987) or PRS (Georgeff and Lansky, 1987). A behavior in Edmund is a group of related functions for sensing and acting upon the

[1] Failing to favor consumatory actions is also a problem for Maes' architecture, which performs particularly poorly in the SE.

[2] Edmund is available on line from http://www.ed.ac.uk/~joannab/edmund.html

world, and the state they need in order to make intelligent discriminations. So, for example, an animated agent may have a behavior for navigating through a world, which is dependent on a stored set of experiences pertaining the the location and accessibility of various landmarks or salient items. Or, it might have a greeting behavior which selects from a variety of possible expressions, which in turn reflect the agent's mood or familiarity. The behaviors of the agent are responsible for monitoring the parts of the environment that are salient to them, for remembering salient experiences such as the agent's own current motivation or previous social encounters, and for controlling expression of their associated actions. However, a behavior does not on its own determine *when* an action is expressed; that is left to the action selection mechanism.

The POSH action selection mechanism is intended to provide reactivity while at the same time provide for exploiting the advantages of hierarchical and sequential control for combating the combinatorial complexity of the action selection problem. The basic element of control is a hierarchical reactive plan similar to those used in RAPs (Firby, 1987), teleo-reactive programs (Nilsson, 1994) and PRS (Georgeff and Lansky, 1987). Complexity is controlled by the presence of hierarchical and even sequential elements in the reactive plan — at any one point of execution of a particular plan, only a limited number of options is being selected between. Reactiveness comes from two sources. First, there are several different hierarchies, or active plans, operating in parallel. These hierarchies are still prioritized: a higher priority plans can interrupt any lower priority ones if it becomes apparent that the agent should attend to a more urgent matter. The second source of reactiveness in POSH is that no stack is saved within the hierarchies, only the root and the current location. Consequently, when a sub-section of a plan completes (either by succeeding or failing) the entire situation within the plan is re-evaluated, thus ensuring that the current activity is still appropriate. This slip-stack approach supports chains of arbitrary length, including loops, in a control structure, which allows for cycles of arbitrary length to be described. An example would be controlling the alternate paces of two-legged walking until a destination is reached or some other goal interferes.

Further details of the architecture The most fundamental level of control under POSH are primitive actions and sense predicates. Both of these are simple interfaces to the behaviors which control their expression[3]. These primitives are in turn organized into two fundamental control structures, the *action pattern*, which executes a set sequence of behavior, and the *competence*, which is a single-tier reactive plan. The elements of an action pattern are the sensory or action primitives. A competence is a single tier of a conventional reactive plan, somewhat like a triangle table or TR (Nilsson, 1994). A competence consists of a set of elements which may be either action patterns or other competencies. The elements of a competence are prioritized and may have triggers. The highest priority element that triggers successfully is allowed to execute. The very highest priority element of a competence is a goal trigger, which on successful execution (a recognition that the competence's goal is achieved) terminates the competence. Competencies also terminate if none of the elements can fire, or if an element that is a competence is selected, in which case it replaces its parent on the slip-stack. Examples of Edmund control hierarchies are included in the Results section, below, which will make this organization clearer.

The following aspects of Edmund facilitate its reactive nature:

- Each element of a competence is prompted to perform by its own perceptual requirements. There is no error passing or assumption of success for an operation. An action sequence will complete regardless of the success of its elements. Competence execution time can be limited by the habituation of elements.
- The top level of the hierarchy consists of special competencies called *drives*. Drives execute all their elements in course-grained pseudo-parallel with best-attempt scheduling for the frequency of attention to the varying elements. This allows for the constant monitoring of important sense modalities or internal states and the possibility of interruption for urgent or opportunistic conditions. Being a competence, the drive's elements are still prioritized. This is the parallel ordinal rooting of the hierarchical plans.
- No stack is maintained as control traverses through a hierarchy of competencies. If a competence activates an element which is also a competence, the new competence takes its parent's place in receiving the attention from the root drive. If a competence chain completes, the drive's attention returns to its initial element for that chain. This is the slip-stack hierarchy.

Edmund thus has the potential for expressing fairly dynamic behavior, but it does not have the complete opportunism or full democracy of Maes' and Tyrrell's architectures. On the other hand, Edmund allows for the expression of both sequences and prioritization. It allows for the utilization of selective attention to ensure particular tasks are completed fluently. Because Tyrrell's simulated environment provides all of the memory, action and sensory primitives required for its task, the experiments below only make use of Edmund's POSH action selection, the behavior-based element of the architecture is not really exploited.

Experiments

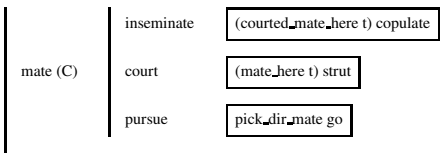
Procedure

The code for controlling an agent using the Edmund architecture within the SE was written over a three week period.

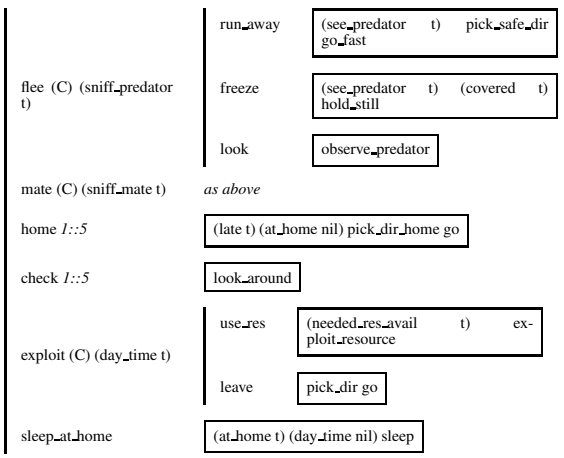
[3]Notice, as is standard for behavior-based control, the primitive "actions" are themselves closely tied to perception. The sense primitives are only used for informing the decisions of the action selection mechanism.

Total programming time was approximately six days. Considerable time had to be devoted to evaluation; the highly dynamic nature of the SE lead to very volatile results, so 6000 simulations (representing no more than 10 days of life each) had to be run for statistically significant results.

The first POSH plan was the most simple one that could generate a genetic fitness measure – one that only pursued mating. Thus the root drive "life" had a single competence "mate", which consisted of:



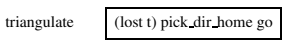
This simple program performed fairly well. Though on average it lived under half a day, it had a genetic fitness of slightly more than 2.0, immediately outperforming Tyrrell's best implementation of Maes (1991). Other obvious behaviors were added gradually, testing the new additions and ordering over approximately 800 runs of the simulator each. Simplicity was favored unless the program improved very significantly. After this phase of development the components of life in order of priority were:



Homing in the evening and looking for predators take priority intermittently, but habituate to allow other, lower priority behaviors to express themselves immediately after they have fired.

At this point, the genetic fitness of the Edmund animal was approximately 65% of the ERP's. The next phase of development was to examine the animal's performance step-by-step in the SE graphical user interface. This resulted in the following observations and changes:

- The animal was often killed by feline predators while freezing. Fixed by prohibition to the freezing action pattern in the presence of felines.
- The animal seemed too concerned with staying in shelter and failed to explore much unless chasing mates. Altered by reducing the desire for shelter in pick_dir.
- Less concerned with shelter, the animal now sometimes explored so far afield it lost bearing and could no longer find its den at nightfall. Fixed with the addition of the following action pattern with, the same priority as homing, but active all day and allowing no habituation:



- The animal often had unforced accidents with irrelevant animals and cliffs, particularly while it was pursuing mates. Fixed by altering pick_mate_dir to take these into account.

Another class of problems could not be fixed so programmatically, because they dealt with tradeoffs. For example, the animal's erratic perception frequently leads it to hallucinate a predator nearby. Too high of priority to perceived predators disrupts behavior coherence sufficiently to interfere with mating, while too little attention to predators reduces mating opportunities by reducing the expected lifespan. This class of problem was addressed by running a large number of simulations with randomly selected values for six critical thresholds: four for detecting predators (observing or fleeing each of the two types), one for seeking shelter and one for avoiding dangerous places.

Linear regression on this data was uninformative, because the variables relationships to the animals success were neither linear nor independent. Instead, we selected the variable sets for the top few performing individuals out of 4400 trials, and tested each over 800 additional runs. The best set improved the animal's mean performance by 25 percent.

At this point Edmund bested the ERP in all the test worlds except for world 3. Examining the simulator code showed that world 3's change primarily differed by substantially reducing the energy consumption for certain activities (such as mating and running) as well as significant changing the conspicuousness of some behaviors, both positive and negative. This lead to the postulate that Edmund's mouse was wasting too much time or conspicuousness on grazing. Consequently the **exploit** competence and its elements were divided into their constituent parts, eliminating the one case of "compromise candidate" determination that had made up Edmund's solution. Also, flight from predators was changed to be the same as normal cautious exploration, only faster.

Results

Edmund performed better (that is, mated more), but not significantly so (critical ratio (F) = 1.36) in Tyrrell's worlds. However, it was very significantly better than the ERP in every world except for variant world 3, and also than any other architecture reported in (Tyrrell, 1993). The final results (over 6600 trials) were:

World	Edmund	ERP	F
Standard	9.12 (0.19)	8.09 (0.17)	3.95^{**}
Var. 1	4.02 (0.09)	3.61 (0.09)	3.1^{**}
Var. 2	9.67 (0.2)	8.16 (0.16)	5.73^{**}
Var. 3	11.23 (0.23)	13.38 (0.23)	-6.58^{**}

where the parenthetical numbers indicate standard error, or the standard deviation of the mean scores.

We also tested both architectures in an additional set of worlds which were identical to Tyrrell's except that food was significantly scarcer. The range of possible initial food supplies was modified as follows:

Food	Original	Sparse
fruit	[50, 81]	[5, 35]
cereal	[45, 26]	[2, 27]
prey	[25, 36]	[2, 15]

which resulted in the following performances:

World	Edmund	ERP	F
Standard	8.17 (0.19)	4.77 (0.12)	15.01^{**}
Var. 1	3.56 (0.09)	2.46 (0.06)	9.59^{**}
Var. 2	10.79 (0.18)	4.56 (0.12)	27.6^{**}
Var. 3	10.74 (0.24)	12.53 (0.23)	-5.47^{**}

As can be seen, Edmund also coped substantially better than the ERP with this more difficult situation, though again with the exception of the third variant world, to which Tyrrell's implementation of the ERP seemed particularly well adapted. In this world, however, the overall performance of the Edmund animal was significantly better than the ERP.

Another important result pertaining to this paper's thesis is to evaluate the relative complexity of the two systems from a design stance. Edmund's agent required 20 action primitives and 22 dedicated sensing primitives. It also had 7 competencies and 23 action sequences defined in its final program script. The ERP's had 61 sensing nodes and 278 other nodes, of which 216 were intermediate points in its hierarchy. Edmund had 26 thresholds embedded in its primitives, of which 6 were not one of four standard values (0.2, 0.4, 0.6 or 0.8). The ERP had 264 weights, of which 189 were either 1.0 or -1.0. However, the other 75 weights took 37 separate values. In summary, the control of Edmund's agent was approximately an order of magnitude more simple from a design standpoint than that of the ERP's.

Analysis and Implications

Although there is an acknowledged advantage to following the original research and thus having a target to beat, both the relative lack of complexity of the POSH action selection and the significantly superior results are strong evidence that managing complexity through selective attention can be more important than having a fully reactive architecture.

This result is supported by the fact that natural evolution has also selected attention focussing strategies. Animals seem not only control for what information is worth attending to, but when. Ethology has long had examples such as the digger wasp, which treats caterpillars as food only when they are engaged in appetitive behavior (Lorenz, 1981). Rats also will not engage in any appetitive behavior (such as foraging or eating) without the appropriate triggering of their limbic system (Carlson, 1994). Recent work on decoding hippocampal firing patterns indicates that cells can have different roles and participate in different ensembles depending on the particular behavioral context the rat thinks itself to be in (Wiener, 1996). The human brain is extremely context sensitive — even the retina has more neurons feeding into it then taking information out of it, indicating that even at the very earliest levels of sensing, expectations grounded in current context significantly affects what is perceived.

The work described in this paper was originally conducted to address the criticisms leveled by researchers such as Maes, Brooks, Goldfield, and Hendriks-Jansen at not only artificial intelligence, but also psychology and philosophy. These scientists have claimed that the use of hierarchy for control is intractable for agents needing to survive in a dynamic world. They have neglected significant evidence for structured control in observed natural behavior (e.g. Dawkins, 1976), through reasoned argument (e.g. McGonigle and Chalmers, 1998), and in the experience artificial intelligence (e.g. Bonasso et al., 1997; Kortenkamp et al., 1998) in an over-reaction against the hopelessly unrealistic AI systems that preceded their work. While there can be no doubt that animals have dedicated parallel mechanisms for monitoring the occurrance of significant, plan altering events, there can also be no question of the importance of controlling the combinatorial complexity of action selection, nor of the critical nature of the design process. Whether the control for an agent is evolved, designed by hand, or some combination of the two, the agent's architecture must provide the right representation of control in order to make that development process likely to succeed.

Conclusion

This paper has presented work indicating that even (or perhaps especially) in an extremely dynamic and dangerous environment, an agent may be better off ignoring some of the information available to it. This is because finding the correct design, whether by hand-coding or through learning, is the key bottleneck to developing intelligence, and reducing information reduces the complexity of the task. The point is illustrated by the direct comparison in an experimental setting of two representative architectures: a fully informed architecture, Tyrrell's Extended Rosenblatt and Payton architecture, which had previously been evaluated against several of the other leading fully reactive architectures; and an architecture utilizing selective attention, called Edmund, which was shown to have significantly better performance and to allow for significantly simpler design.

Acknowledgments

I would like to thank Toby Tyrrell, The University of Edinburgh, and the CMU AI Archives for making the SE freely and publicly available; and Will Lowe for his technical assistance and proofreading.

References

Agre, P. E. and Chapman, D. (1990). What are plans for? In Maes, P., editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, pages 17–34. MIT Press, Cambridge, Massachusetts.

Blumberg, B. M. (1996). *Old Tricks, New Dogs: Ethology and Interactive Creatures*. PhD thesis, MIT. Media Laboratory, Learning and Common Sense Section.

Bonasso, R. P., Firby, R. J., Gat, E., Kortenkamp, D., Miller, D. P., and Slack, M. G. (1997). Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2/3):237–256.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2:14–23.

Brooks, R. A. (1991). Intelligence without reason. In *Proceedings of the 1991 International Joint Conference on Artificial Intelligence*, pages 569–595.

Bryson, J. and McGonigle, B. (1998). Agent architecture as object oriented design. In Singh, M. P., Rao, A. S., and Wooldridge, M. J., editors, *The Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL97)*. Springer-Verlag.

Carlson, N. R. (1994). *Physiology of Behavior*. Allyn and Bacon, Boston, 5 edition.

Chapman, D. (1987). Planning for conjunctive goals. *Artificial Intelligence*, 32:333–378.

Dawkins, R. (1976). Hierarchical organisation: A candidate principle for ethology.". In Bateson, P. P. G. and Hinde, R. A., editors, *Growing Points in Ethology*, pages 7–54. Cambridge University Press, Cambridge.

Firby, J. (1987). An investigation into reactive planning in complex domains. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 202–207.

Georgeff, M. P. and Lansky, A. L. (1987). Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 677–682, Seattle, WA.

Goldfield, E. C. (1995). *Emergent Forms: Origins and early development of human action and perception*. Oxford University Press.

Hendriks-Jansen, H. (1996). *Catching Ourselves in the Act: Situated Activity, Interactive Emergence, Evolution, and Human Thought*. MIT Press, Cambridge, MA.

Hull, C. (1943). *Principles of Behaviour: an Introduction to Behaviour Theory*. D. Appleton-Century Company.

Humphrys, M. (1996). Action selection methods using reinforcement learning. In Maes, P., Matarić, M. J., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, *From Animals to Animats 4 (SAB96)*, Cambridge, MA. MIT Press.

Kortenkamp, D., Bonasso, R. P., and Murphy, R., editors (1998). *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*. MIT Press, Cambridge, MA.

Lorenz, K. (1950). The comparative method in studying innate behaviour patterns. In *Symposia of the Society for Experimental Biology*, volume 4, pages 221–268.

Lorenz, K. (1981). *Foundations of Ethology*. Springer-Verlag.

Maes, P. (1991). A bottom-up mechanism for behavior selection in an artificial creature. In Meyer, J.-A. and Wilson, S., editors, *From Animals to Animats*, pages 478–485, Cambridge, MA. MIT Press.

McGonigle, B. and Chalmers, M. (1998). Rationality as optimised cognitive self regulation. In Oaksford, M. and Chater, N., editors, *Rational Models of Cognition*. Oxford University Press.

Newell, A. and Simon, H. A. (1981). Computer science as empirical inquiry: Symbols and search. In Haugeland, J., editor, *Mind Design*, chapter 1, pages 35–66. MIT Press, Cambridge, MA.

Nilsson, N. (1984). Shakey the robot. Technical note 323, SRI International, Menlo Park, California.

Nilsson, N. (1994). Teleo-reactive programs for agent control. *Journal of Artificial Intelligence Research*, 1:139–158.

Rosenblatt, K. and Payton, D. (1989). A fine-grained alternative to the subsumption architecture for mobile robot control. In *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*.

Rosenschein, S. J. and Kaelbling, L. P. (1995). A situated view of representation and control. *Artificial Intelligence*, 73.

Tyrrell, T. (1993). *Computational Mechanisms for Action Selection*. PhD thesis, University of Edinburgh. Centre for Cognitive Science.

Wiener, S. I. (1996). Spatial, behavioral and sensory correlates of hippocampal CA1 complex spike cell activity: Implications for information processing functions. *Progress in Neurobiology*, 49(4):335.