The Study of Sequential and Hierarchical Organisation of Behaviour via Artificial Mechanisms of Action Selection

Joanna Joy Bryson

M.Phil.
University of Edinburgh
2000

Abstract

One of the defining features of intelligent behaviour is the ordering of individual expressed actions into coherent, apparently rational patterns. Psychology has long assumed that hierarchical and sequential structures internal to the intelligent agent underlie this expression. Recently these assumptions have been challenged by claims that behaviour controlled by such structures is necessarily rigid, brittle, and incapable of reacting quickly and opportunistically to changes in the environment (Hendriks-Jansen 1996, Goldfield 1995, Brooks 1991*a*). This dissertation is intended to support the hypothesis that sequential and hierarchical structures are necessary to intelligent behaviour, and to refute the above claims of their impracticality. Three forms of supporting evidence are provided:

- a demonstration in the form of experimental results in two domains that structured intelligence can lead to robust and reactive behaviour,
- a review of recent research results and paradigmatic trends within artificial intelligence, and
- a similar examination of related research in natural intelligence.

The experimental domains are an autonomous mobile robot, and a simulated rodent situated in a simulated natural environment. Autonomous mobile robots are the standard platform for demonstrating the advantages of less structured reactive architectures, in this domain qualitatively similar results are shown to the reactive literature. In the simulated domain, quantitative comparisons were possible, and the structured approach taken in this dissertation shows significantly better results than the best fully parallel, reactive architecture previously reported (Tyrrell 1993).

The approach to synthetic intelligence used in these experiments exploits the advantages of hierarchy and sequence within a distributed cognitive architecture where at least partially modular subsystems execute in parallel. By successfully integrating these two strategies of control, it avoids the rigidity normally associated with hierarchical control, while retaining the advantages in combating combinatorial complexity of the action selection problem.

This dissertation also makes contributions by demonstrating the artificial intelligence architectures can be considered and tested as psychological hypotheses. It provides an explanatory chapter for enabling psychologists to examine this literature, and a review of recent agents architectures analysed as hypotheses and compared to the approach used in experiments. It also includes a discussion of various methodologies of artificial intelligence research, and their appropriateness for psychological laboratories.

Acknowledgements

First and foremost, I would like to thank my supervisor, Brendan McGonigle, for providing me with a stimulating research environment, a studentship, many interesting conversations and useful arguments, and a great deal of patience. Thanks also to my second supervisor, John Hallam, who provided useful comments, conversation and encouragement along the way.

Thanks to all the people and other primates of the Laboratory for Cognitive Neuroscience and Intelligent Systems. Particular thanks to Vikram Jaswal for conversations and pointing out Adams (1984) to me; Ben St. Johnston and Lawrence Warnett, fellow thralls to the robot; and Tony Dickinson, Della Purves, Mark Humphrys and John Ravenscroft for being around and telling me interesting things.

Thanks to all the people of the Edinburgh Mobile Robot Group, particularly Heba Al-Lakany, Jose Carmena, Nuno Chalmique Chagas, Will Chesters, John Demiris, Arturo Espinosa-Romero, Sandra Gadanho, Simon Perkins, Richard Reeve, Mike Spratling, Tim Taylor, Henrik Lund, Herbert Peremans, Chalita Hiransoog, Barbara Webb, Yuval Marom, Aude Billard, Mark Wright, Auke Jan Ijspeert, Ashley Walker, Jeremy Wyatt and Gill Hayes for companionship and useful conversation.

Ashley Walker, Jena Jordahl and Will Lowe helped video the robot. Lawrence Warnett provided Figures 4.3 and 4.2. Ian Horswill was supervising me during the development of Edmund at MIT in 1994-5 and made many helpful suggestions. Lynn Stein taught me a lot about dissertations. Dave Glasspool pointed out Norman & Shallice (1986) and it's similarities to Edmund to me. Kris Thórisson spent *lots* of time talking to me about Edmund and agent architectures during the period I was writing up this thesis; Victor Bonilla, Bruce Blumberg, Noel Sharkey, Bill Smart, Bob French, Mark Humphrys, Myra Wilson, Andreas Birk, Rene Schaad and Will Lowe all gave me substantial conversations on the topic as well.

Thanks to Viv Kendon, Gert Westermann, Sarah Gingell, and David Sterratt for putting up with me. Thanks to Johanna Moore for putting up with my dissertation. Special thanks to anyone I forgot to thank.

Thanks to Will Lowe for teaching me about science.

Declaration

I hereby declare that I composed this thesis entirely myself and that it describes my own research.

Joanna Joy Bryson
Edinburgh
27 January, 2000

Contents

List of Figures

Chapter 1

Introduction

1.1 Thesis

One of the defining features of intelligent behaviour is the ordering of individual expressed actions into coherent, apparently rational patterns. From approximately 1951 until the mid 1980s, the dominant theories in both psychology and artificial intelligence for explaining intelligent behaviour held that hierarchical and sequential structures internal to the agent or animal underlie this ordered expression (e.g. Lashley 1951, Tinbergen 1951, Piaget 1954, Hull 1943, Dawkins 1976, McGonigle & Chalmers 1996). However, the last two decades have seen an increase of support for a more dynamic theory of of intelligence (see Port & van Gelder 1995, for a review). This new theory holds that intelligence, like the brain itself, is actually composed of enormous numbers of small processes operating in parallel. Several researchers in this new paradigm have claimed that behaviour controlled by hierarchy is necessarily rigid, brittle, and incapable of reacting quickly and opportunistically to changes in the environment (Hendriks-Jansen 1996, Goldfield 1995, Maes 1991). They suggest that the apparent hierarchical organisation of behaviour is not the result of internal structured control, but it is rather only an inadequate model imposed on a far more complex dynamic process.

This dissertation presents a body of research examining the claim that hierarchy and sequence are not integral to intelligence, and concludes that this stance is not justified. This dissertation is primarily concerned with examining models of dynamic intelligence, both in the literature and the laboratory. A review of the last decade's artificial intelligence literature in this area indicates a need for both distributed and hierarchical control in the same system. This is followed by laboratory research demonstrating that such "hybrid" systems can meet and exceed the standards of fully distributed, dynamic control. Finally, this evidence that hierarchical control *can be* integrated into intelligent dynamic systems is then supported by further evidence that it *is* so integrated in mammalian intelligence. This final evidence is gathered from recent research in both psychology and neuroscience. On its own, the biological evidence has not been considered conclusive. The main contribution of this thesis is to strengthen the claim that hierarchy and sequence are integral to intelligent systems by synthesising these several sources of support.

This dissertation makes several additional contributions besides the direct support of its thesis. It provides a framework for considering evidence from artificial intelligence and psychology together, it examines methodological issues of performing artificial intelligence research in psychology laboratories, and it provides a detailed examination of one model from artificial intelligence in terms of psychological plausibility. These contributions support the main thesis indirectly by supporting the methodological approach of the dissertation.

1.2 Introduction to the Thesis Problem

1.2.1 Hierarchical Theories of the Organisation of Intelligent Behaviour

Initially it appears obvious that there must be some structure or plan to behaviour. Such structures radically simplify the problem of choosing a next behaviour by reducing the number of options that need to be evaluated in selecting the next act. The standard strategy for AI systems, which has been considered to recapitulate natural intelligence, is to convert perceptions and intentions into a plan — a sequence or partial ordering of behaviours which will attain the current goal. The term "plan" in this context does not necessarily indicate intentionality, nor even the conventional sense of "planning". Rather, it refers to a plan in the sense of something established, a sort of blueprint for action. Whether this blueprint is a result of instinct, past experience or an immediate creative process is a separate, though related question; assuming a hierarchical model, then some plans may be of different origins than others.

A plan is considered hierarchical if its elements might in turn be plans. For example, if a dog is hungry, it might go to the kitchen, then rattle its bowl. Going to the kitchen would itself entail finding a path through the house, which involves moving through a series of locations. Moving between locations itself requires a series of motor actions. The theory of hierarchical control supposes that the mechanisms responsible for the fine muscle control involved in the dog's walking are not the same as those responsible for choosing its path to the kitchen, and these in turn are not necessarily the concern of the system that determined to move to the kitchen in the first place. A plan is considered sequential to the extent that its elements deterministically follow each other in a fixed order, for example the order in which a dogs' feet are raised and advanced while it is moving in a particular gait.

Hendriks-Jansen traces the hierarchical theory of behaviour organisation in animals and man to the ethologist McDougall (1923), who presented a theory of the hierarchy of instincts. Ethological theory during this period, however, was dominated by Lorenz, who "denied the existence of superimposed mechanisms controlling the elements of groups" instead believing that "the occurrence of a particular activity was only dependent on the external stimulation and on the threshold for release of that activity." (Baerends 1976 p. 726 cited in Hendriks-Jansen 1996 pp. 233–234). This theory dominated until Lashley (1951) reintroduced the idea of hierarchical organisation of behaviour. Lashley supports hierarchy on the basis that there could be no other explanation for the speed of some action sequences, such as those involved in human speech or the motions of the fingers on a musical instrument. Neural processes are simply too slow to allow elements of such sequences to be independently triggered in response to one another. Lashley therefore proposes that all the elements of such a sequence must be simultaneously activated by a separate process — the definition of hierarchical organisation.

Lashley's argument was taken up and extended by Dawkins (1976), who further argued for hierarchical theories of control for reasons of parsimony. Dawkins argues that it is more likely that a complex action sequence useful in multiple situations should be evolved or learned a single time, and that it is also more efficient to store a single instance of such a skill. Dawkins' arguments and proposals anticipate many of the techniques developed later for robust control in artificial intelligence (e.g. Brooks 1991*b*).

From roughly the time of Lashley's analysis, hierarchical models have been dominant in attempts to model intelligence. Particularly notable are the models of Tinbergen (1951) and Hull (1943) in ethology, Chomsky (1957) in linguistics, and Newell & Simon (1972) in artificial intelligence and human problem solving. Mainstream psychology has been less concerned with creating specific models of behaviour control, but generally assumes hierarchical organisation as either an implicit or explicit consequence of goal directed or cognitive theories of behaviour

(Bruner 1982). Staged theories of development and learning are also hierarchical when they described complex skills being composed of simpler, previously-developed ones (Piaget 1954, Karmiloff-Smith 1992, Greenfield 1991).

1.2.2 Dynamic Theories and the Thesis Problem

The competing theory, that responsive animal intelligence cannot possibly be governed by hierarchical control, has emerged from some of the practitioners of the dynamic hypothesis of cognition (van Gelder 1998). As will be emphasised later in this dissertation when competing dynamic models are presented, not all researchers within this paradigm are in principle opposed to some hierarchical ordering, however an increasingly vocal subgroup are. The theory of dynamic action expression suggests that complex dynamic or chaotic systems operate within the brain producing the next behaviour not by selecting an element of a plan, but rather as an emergent consequence of many parallel processes (e.g. Minsky 1985, McClelland & Rumelhart 1988, Maes 1991, Brooks 1991*a*, Goldfield 1995, Kelso 1995, Hendriks-Jansen 1996, van Gelder 1998). Evidence supporting the older hypothesis of structured hierarchical behaviour is seen to have been biased by the hierarchical and sequential nature of human explicit thought and language. In particular, because much theoretical work in psychology is conducted using computer models, theories may be biased towards the workings and languages of the serial processors of the machines available to most psychologists (Brooks 1991*a*).

Appeals of dynamic theories of intelligence include that they better explain the fact that errors are made in sequencing of even familiar tasks (Norman 1998, Henson 1996). More importantly, dynamic theories allow for new information to constantly influence the course of actions. Thus if the dog described earlier on the way to its kitchen happens to pass a dropped sandwich, the action “eat what’s here” should suddenly overtake “visit the kitchen.” Systems with this capacity to be responsive and opportunistic are described in the artificial intelligence literature as being *reactive*.

A fundamental appeal of the dynamic hypothesis is that it is necessarily correct, at least to some level. Assuming a materialist stance, intelligence is known to be based in the parallel operation of the body’s neural and endocrine systems. It is nearly as well accepted that human and animal behaviour can be described as hierarchically ordered (Dawkins 1976, Greenfield 1991, Byrne & Russon 1998). The question is, how and when are these behaviours so organised?

Is the order purely apparent, or does it emerge in the brain prior to the external expression of behaviour? The research in this dissertation supports the hypothesis that at least some of the apparently hierarchically ordered behaviours observed in higher animals are in fact determined by hierarchical and sequential control structures organised within the animals' central nervous system.

The experimental section of this dissertation utilises a model which combines these two theories to a particular extent. It provides for a limited number of parallel processes, but also uses hierarchy to provide organised control within these processes. Such a hybrid model is not particularly novel within psychological theory, however such models are still subject to criticism and controversy. For example, the following is a quote from a recent book and dissertation[1] on the cognitive implications of research into dynamical systems. In this quote, the author is referring to the hybrid model of Hull (1943), but the description and critique could equally well apply to the model used in this dissertation.

> [In hybrid systems] Environmental contingencies play a part at the choice points and in the form of orienting feedback, but the part they play can be explained only in terms of the entities manipulated by the program, which of course takes the form of a temporal sequence of formally defined instructions.
>
> Nearly forty years of experience in AI have shown that such a control mechanism soon gets into trouble in the real world because of its lack of flexibility, the need to plan for all possible contingencies, the combinatorial explosion, the frame problem, and the problems of interfacing a formally defined planner, working with an internal representation of the world conceptualised as a task domain of objects, properties, and events, to effectors and receptors that need to deal with a noisy real world that clearly is not preregistered into objects, properties, and events. (Hendriks-Jansen 1996, page 245.)

Although Hendriks-Jansen is not a psychologist but a philosopher of mind, his argument represents a real challenge to the theory of hierarchical control. This challenge has recently been pursued most strenuously by those with alternative models to demonstrate: researchers using modular or neural theories of artificial intelligence such as Brooks (1991*b*), Maes (1991), and

[1] University of Sussex, School of Cognitive and Computing Sciences.

McClelland & Rumelhart (1988); and researchers within the mathematical and physical sciences with models of structure driving from chaos and complexity, such as Goldfield (1995), Kelso (1995) and Bohm (1980). Nevertheless, the problem of how hierarchically ordered behaviour emerges from the brain has also been an ongoing issue in psychology. The well-known defences of Lashley and Dawkins are still being revisited (Houghton & Hartley 1995, Nelson 1990). In response to a recent target article on imitation which assumed hierarchical structure to behaviour (Byrne & Russon 1998), nearly a fifth of the commentaries chosen to appear with the article questioned the existence of hierarchical control (Vereijken & Whiting 1998, Mac Aogáin 1998, Jorion 1998, Gardner & Heyes 1998, of 21 commentaries). As Gardner & Heyes (1998) point out, "The mere fact that [one] can describe behaviour in terms of goals and subgoals is not evidence that the behaviour was executed under hierarchical control." Clearly, more evidence in this debate is needed.

1.3 Introduction to the Dissertation Approach

1.3.1 Psychology and Artificial Intelligence

Psychology is the science of intelligent behaviour, that is, of behaviour resulting from human or animal intelligence. Within this dissertation, the term *psychology* will be applied broadly to include any of the disciplines dedicated to explaining natural intelligence, through observation of or experimentation on human or animal behaviour.

One of the ways to study natural intelligence is to create models of it. The advent of computers has allowed for the creation of functioning, animate models, which allow for particularly powerful explanations. Once an intelligent process has been fully described and expressed on a computer, it can be tested directly by observation, either in comparison to the predicted behaviour or to the behaviour of natural systems.

Simultaneous with the development of this capacity for animate modelling has been the emergence of the discipline of Artificial Intelligence. Artificial Intelligence (AI) is dedicated to the creation of intelligence in the artifact, and does not necessarily seek to directly replicate or model natural intelligence. Many practitioners of AI are more engineers than scientists. Among the scientists in the discipline, the research of many is more nearly related to mathematics or philosophy than psychology, as they explore the nature of information or the limits of computation. Some, however, are interested in natural intelligence.

Methodologically, this dissertation focuses on the practices of a particular AI research community, the agent-based community. Their approach involves a bottom-up, ethology-inspired approach to explaining intelligence that was first popularized in a series of thought experiments by Braitenberg (1984) (Walter 1950, though see also). Human intelligence is too complex an issue to be studied (or built) as a whole. As a consequence, AI has traditionally focussed on isolated human abilities that are considered exceptionally intelligent, such as playing chess, translating between languages and mathematical theorem proving. Unfortunately, successes in any one of these domains has failed to generalise into any of the others, or into the important tasks originally presumed to be more mundane, such as controlling motors for walking on two legs or recognising faces. Agent-based AI rejects this approach, instead favouring a recapitulation of evolution. Human intelligence is presumed to require primate intelligence, which in turn requires more basic mammal, reptile and even insect-level intelligence. Agent-based AI has taken a dynamic stance towards intelligence: its approach has been to demonstrate that the complex behaviour of a complete agent can arise from interaction of the number of simple units of intelligence running independently and concurrently. Work in this area is reviewed extensively in the following two chapters.

1.3.2 Establishing a Research Dialogue

To the extent that agent-based AI researchers are interested in replicating natural intelligence, there is a potential for fertile exchange between AI and psychology. Such a relationship exists between neuroscience and artificial neural networks, with several respected journals publishing papers from either field. The attempt to increase dialogue between psychology and agent-based AI has been an ongoing concern, particularly within the AI community. Three workshops on this topic of over 100 participants each were held in the UK alone in 1997, taking place in London under the auspices of the IEE, and in Oxford and Edinburgh, hosted in these latter two by their respective universities. All three were organised by AI practitioners with psychologists, biologists, specialists in evolution and neuroscientists invited as speakers and guests. In addition, this decade has seen an emphasis on inviting speakers from the natural intelligence sciences to the symposia of the Societies for the Simulation of Adaptive Behaviour (SAB), and for the Study of Artificial Intelligence and the Simulation of Behaviour (AISB).

The AI researchers seek both feedback on their own designs and new inspiration from current research in animals. Unfortunately, these efforts are still largely unsuccessful. The natural scientists, while sometimes finding the AI work intriguing, have seen little reflection of the incredibly complex systems with which they were experimenting in the necessarily simple and mechanistic artificial models. Yet psychologists often draw and describe in their own talks and papers far simpler models than any functional artificial agent embodies.

This dissertation is intended to demonstrate an improved method for establishing such a dialogue. Rather than attempting to teach psychologists to fully understand AI, the intent is to describe AI in psychological terms. Because the two fields share common problems, this redescription should not only communicate information about AI, but also provide an interesting new perspective on the problems of psychology. In particular, agent-based AI is characterised by a large number of approaches to developing agents; these approaches are referred to as *architectures*. This dissertation uses the following method for cross-disciplinary analysis of different agent architectures:

1. isolate the differences between one approach and its competitors,
2. express those differences as hypotheses in terms understandable by a practitioner of the alternate discipline (in this case psychology), and
3. propose these hypotheses as research questions.

This allows researchers from the first discipline to seek experimental evidence in the second to prove or disprove their hypotheses, thus validating or invalidating the original architecture or approach.

This approach is obviously a simple extension of standard experimental method, but it is a perspective that has been missing in the interactions between behaviour-based AI and psychology. Although a deeper understanding of both disciplines by researchers from both sides would undoubtedly increase mutual contributions; reducing communication to succinct, fundable research questions is both functional and expedient.

1.4 Summary and Outline of the Remaining Dissertation

This chapter has introduced the central thesis of the dissertation: that hierarchy and sequence can be integral parts of the control of intelligent behaviour. It has described the history of this theory of hierarchical control in psychology, and the contributions and challenges of the new, more dynamic hypotheses of intelligence. It has also provided a brief introduction to the methodological issues of the use of the cross disciplinary approach taken in establishing the thesis.

Despite the methodological argument of the previous section, in a cross-disciplinary dissertation there should be some attempt to establish a common vocabulary and an understanding of the differing viewpoints and frameworks of the disciplines. This dissertation is premised on the hypothesis that the methods and results of artificial intelligence and psychology can be mutually informing — that they can be used together to establish a thesis. The next chapter, Chapter 2, establishes this premise by defining the terms and problems addressed in agent-based AI, and relating them to the terms and concerns of psychology.

Chapter 3 begins the examination of the main thesis. It presents several alternative hypotheses on the organisation of intelligence, described as AI control architectures. This includes a hypothesis-level description of the architecture used in the laboratory experiments described in the following chapter, Chapter 4. The laboratory experiments and their analyses are followed by an analysis of the results of the previous two chapters in terms of the psychological and neurological literature in Chapter 5. These two chapters also contain more in-depth analyses of the AI methodologies used in examining the thesis, and suggestions for future work. Chapter 6 offers a summary and conclusion.

Chapter 2

Premises

2.1 Introduction

As was introduced in the previous chapter, the thesis of this dissertation will be established with evidence coming from research in both artificial and natural intelligence. Such an approach requires the premise that the methods and results of these two fields may be mutually informing. This chapter attempts to establish this premise by describing a relationship between psychology, particularly experimental and comparative psychology, and AI, particularly agent-based artificial intelligence.

The first step towards demonstrating this relationship is to create a mapping of the terminologies and concepts between the two fields. Perhaps unfortunately, this does not require the introduction of new words. By-and-large, AI has borrowed its vocabulary from psychology. However, there is a considerable shift in *meaning* of the terms as a consequence of a substantially different perspective and set of problems. The first part of this chapter attempts to describe this perspective shift and the some of the fundamental terms and concepts for agent-based AI.

After having introduced these differences, the question of the relevance of artificial intelligence to psychology and vice versa becomes more stark. The second part of this chapter argues that at least some of the concept shifts and distinctions introduced by artificial intelligence may be useful to psychology, and that it is consequently worth examining progress in this field from a psychological perspective. This perspective can in turn further AI by providing both knowledge and criteria for comparison on the metric of psychological validity.

2.2 Artificial Intelligence: Terminology and Perspective

As just stated in the introduction to this chapter, many of the key terms from artificial intelligence originate in psychology, but have significantly different connotations from those of their original source. This can be easily illustrated with more familiar terms from computer science, such as program, memory and disk. In each case, when the component of computer science was first conceived, it was labelled with an appropriate English word. However, the meanings of these terms have come to take on more connotations as the components and uses of the components developed. For example, there are now many more expectations for a computer program than that it be simply a schedule of events.

Because this dissertation straddles artificial intelligence and psychology, and some terms are used for different purposes in two fields, these terms will be necessarily ambiguous. This section is intended to reduce any potential confusion by introducing the primary terms from AI as they are used in the research reported. The primary differences in terminology and usage are consequents of the fundamentally different perspectives involved: psychology must rely on observation of intelligent behaviour, but AI is concerned with its construction. In general, within this dissertation the definitions of words will be consistent with common usage within the context of the current discussion (that is, either psychology or AI).

2.2.1 Agency

One term from artificial intelligence that has become almost as ubiquitous as "program" is **agent**. To a first approximation, the study of agents is the study of "complete" systems, in contrast to the specialised study of particular components of intelligence. An agent should have its own private knowledge base; it should have purpose or purposes; and it should operate in an environment that is rich, varied and complex (see Brooks 1991*b*, Wooldridge & Jennings 1995). This definition is deliberately broad enough to describe animals as well as some types of artificial intelligence.

Some definitions of "agent" require that one should be able to ascribe an agent goals or intentions. Because artificial intelligence constructs its agents, we have direct access to any information being used to determine an agent's behaviour. In other words, we can know precisely what is on an agent's mind at any given time. We also know what computations the artificial agent is capable of, and we may even construct the agent such that it constantly records or relates which computation it is currently engaged in. Expressing such types of knowledge in English is awkward, particularly when keeping to natural intelligence metaphors. Generally, the resulting language sounds intentional, such as "action selection," (defined below.) In most cases discussed within this dissertation, this intentional language does *not* imply a conscious decision on the part of the artificial agent, or any other humoncular awareness of the current goal. It is simply a description of some aspect of the agent's internal processing state, in contrast to its external expressed behaviour. There is also a separate literature discussing the need for the explicit representation of goals and intentions within an agent — this is discussed below in Section 2.3.1.

This dissertation identifies the subfield of artificial intelligence that studies the problems of creating autonomous agents as **agent-based artificial intelligence**. This is in contrast to to the dominant trend in artificial intelligence of studying a single aspect of intelligence, such as representation, reasoning, vision, kinematics, and so forth. The hypothesis underlying the agent-based approach is that each of these problems considered in isolation is under-constrained. Consequently, the solutions developed by these disciplines have been difficult to link together into larger, more complete systems. Attacking these problems simultaneously, within the constraint of a complete system, should necessarily result in elements that do interact. Satisfying the additional constraints of a complete agent may also lead to intelligence that is more similar to animal intelligence, since obviously animals must meet these same constraints [1] (see further McGonigle 1991).

One set of constraints on natural intelligence are the physical constraints of the "real world." In contrast, many AI systems operate in constructed worlds which may consist only of the numerical data set, a set of simple facts, or language without referents. One of the early emphases of agent-based AI has been the development of physical agents, or **robots**. "Robot" has been defined as any artifact that translates perceptual information received from the real world into action (Brady & Paul 1984). By this definition, an agent that speaks or creates

[1] Within the last five years, the term "agent" has been increasingly appropriated by computer scientists using the agent metaphor as a design strategy to create programs or applications with distributed units of responsibility. "Agent" in this sense is a much weaker usage of the term than that in this dissertation: it does not necessarily have any psychological or AI relevance. As of this writing, no standardised descriptor has yet been established to distinguish these two forms of agents, and in fact, several leading conferences still accept papers concerning both. The psychologist searching this literature must simply be aware of this distinction.

music is also a robot (Bryson 1992). However, most agent-oriented robotic research is on autonomous mobile robots: vehicles which operate and move around in the world without off-board or external direction. Autonomous mobile robots must cope with at least some of the same fundamental problems as animals. For example:

- they must balance multiple, possibly conflicting goals, such as food and safety,
- they must navigate their environment successfully,
- they must choose which behaviours to perform, and
- they must categorise perceptual information.

The use of robotics in research is discussed at length in Section 4.4.

Within this dissertation, the term **animal** is normally used in contrast to *robot*, not in contrast to *human*. There will be some discussion of uniquely human intelligence, but such intellectual attributes are largely outside the scope of the present dissertation. Consequently, unless otherwise stated, "animal" is meant to be inclusive of humans, though not necessarily of lower order invertebrates. More precise equivalent terms would be "intelligent biological agent" or "naturally intelligent agent", but these are avoided for obvious reasons.

2.2.2 Behaviour

This dissertation deals primarily with the problem of **action selection**, that is, the continuous problem of determining what to do next. An **act** here is defined in roughly the same way as it was by the experimental psychologists in the 1930's, as something gross and quantifiable, such as a lever press, rather than at the level of individual muscular or even limb movement (Adams 1984). The primary point of view for AI, however, is one of action *production*, not action result. The typical artificial intelligence researcher thinks as a programmer rather than as a psychologist. Consequently, the AI researcher tends to place more emphasis on what code is being executed by the robot, rather than what behaviour is observable.

This increased emphasis on cause or program relative to result may be the consequence of the researcher being more directly the author of the code than the behaviour. For example, in robotics, it is both easier and typically more precise to make records of code execution than of behaviour. Recording execution requires only slight modification to the readily-accessible program, and the recording is both completely reliable and arbitrarily precise. Recording external behaviour involves the same problems of subjective labelling and stop-watch precision faced by observational psychology.

The positive side of this, of course, is that roboticists can potentially measure both. Thus where a psychologist studying a rat at a lever has only one possible event, a lever press, a roboticist has access to two sources of information: how often the robot selects lever pressing as its action, and how often the lever is actually pressed. As this implies, there is not necessarily a direct link between these two acts.

When an agent (particularly a robot) selects a behaviour for execution, execution is not guaranteed. Failure in sensing or actuation in the robot may result in a "completed" gesture not being successful. For example, a robot attempting to grasp a can from a tabletop might fail to realize that it has in fact knocked the can off the table, and has grasped another object. Another source of failure is **interference**, a chosen action may begin, but then be interrupted by the selection of another action before the first action can complete. Additionally, arbitrary consummatory acts such as pressing a lever or knocking a ball into a goal may be achieved accidentally or incidentally, while the robot is moving to fulfill some other goal.

All of these potential differences between "intent" (selected action) and actual behaviour are also applicable to animals. However, except in special (usually invasive) experiments (e.g. Tanji 1996, Wilson & McNaughton 1994), the selection of the act cannot be registered independent of its successful completion. This was the motivation for the methodological decision by the behaviourists to use the act as their level of description and experimentation — it was considered the only level of description open to the scientific method (Skinner 1935 cited in Adams 1984 p. 4). This point will be taken up again in the discussion of the relevance of AI to psychology, below.

This difference in perspective and emphasis, between the expressed behaviour versus the selected behaviour, has methodological implications which are discussed in the following two sections. However, it also has an important consequence in terminology. The term **behaviour** is often used in agent-oriented AI to identify a program, or a program unit, which is expected when executed to express a particular external behaviour. **Behaviour-based** AI is an approach to programming an artifact where the units of the program are individual behaviours, as just described.

In contrast, the traditional approach to AI programming had been to decompose intelligence along traditional computer science lines. A single reasoning system would process a single monolithic database of facts stored under one uniform representation, and a general-purpose planning system would select which action to take given the current situation. The innovation of behaviour-based AI has been to move away from such global systems of intelligence to modular, specialised, and distributed ones. Such approaches have also been under investigation recently in psychology and philosophy of mind (Fodor 1983, Dennett & Kinsbourne 1992, Karmiloff-Smith 1992, Cosmides & Tooby 1994, Mithen 1996, Elman et al. 1996).

2.2.3 Behaviour Expression and Emergence

The fact that agent-based AI uses the word "behaviour" both to describe a program element and an explicit act would lead to ambiguity even if each software behaviour, when executed successfully, lead to a particular expressed behaviour for which it had been designed. In practice, there are further complexities in the relationship between programmatic behaviours and expressed behaviours, beyond the possible accidental failures of correspondence mentioned above. First, the expression of a programmed behaviour is highly dependent on the context in which it is expressed. The original specifications for behaviours in behaviour-based AI (Brooks 1986, 1991*b*) state that a behaviour should be a tight coupling between specialised sensing and specialised actions. Consequently, a single software "behaviour" may have several different apparent external expressions.

For an example, consider an "avoid obstacle" behaviour in a mobile robot with touch sensors embedded in bumpers around its periphery. The "avoid obstacle" behaviour may generally be fairly inactive, spending most of its time monitoring the state of the touch sensors. However, if one of the touch sensors is activated, the behaviour may take control of locomotion by initially starting away from the direction of contact. For some period after such a contact, the behaviour may have a third form of expression — it might influence other behaviours which are selecting direction, in order to avoid returning to the area where contact was made.

In addition to the fact that one software behaviour may map to many different expressed behaviours, some expressed behaviours may not be the result of any one software behaviour. Often multiple software behaviours will be active simultaneously; or in other words, *in parallel*. To continue the example in the previous paragraph, if the "avoid obstacle" behaviour is trying to avoid a point on the left and fore of the robot where there was a recent collision, and another behaviour, say "approach light" is encouraging the robot to move directly forwards, the result of both behaviours' simultaneous action might be a right-forwards arc around the obstacle. (See Figure 2.1.) Because there is no explicit software behaviour that tells the robot to move in an arc, the expressed behaviour of moving in an arc is said to be **emergent**. Emergence in the context of agent-based AI is any expressed behaviour for which there is no single corresponding programmatic behaviour active within the agent.

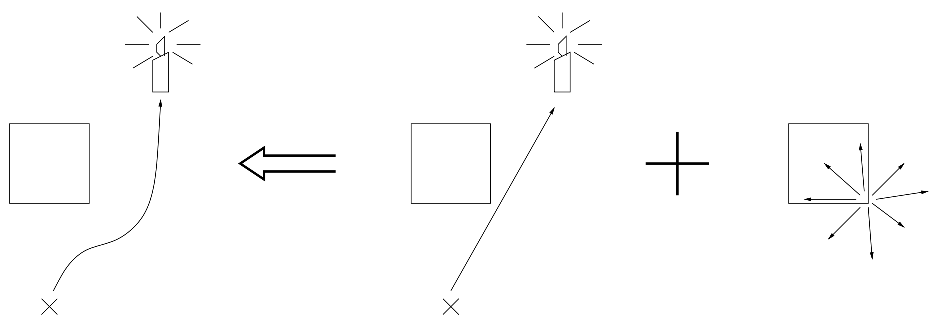


Figure 2.1: An elegant path may be the result of the combined impact of two simple behaviours.

2.2.4 Development

As described in the previous section, the perspective of the artificial intelligence researcher is significantly different from that of the psychologist; the AI researcher's view is more internal and algorithmic. This is partly because the AI researcher exploits the most accurate and accessible data available, the code executing on their agent and causing its behaviour, rather than the external behaviour itself. Another reason for a different perspective is a different motivation. Artificial intelligence is not only interested in description, but also in **development**.

Development in both psychology and artificial intelligence implies a process of improvement and expansion of abilities due to qualitative change in the agent. In nature, this development characteristically occurs as a process of maturation experienced by the agent between its conception and its coming to an adult form. In artificial intelligence, development is almost never something the agent performs itself, but is rather the process by which the agent is created by the researcher.

The AI development process typically takes the vast majority of a research project's lifetime. The typical AI project is to create a particular intelligent behaviour, and once this has been completed, the project is essentially over. To this extent, many artificial intelligence researchers are more engineers than scientists. Consequently, many AI papers address what can be done and how, rather than comparing two similar approaches or artificial agents. This issue is also further addressed below in the section on methodology, Section 4.4.

In contrast to development, **learning** is typically seen as changes in established representational systems brought about as a consequence of experience. The capacity to learn is consequently something that must be provided in advance, in both animal and artificial agent. Learning in this context is discussed in this dissertation in Section 5.3.4.

2.2.5 Experimental Method

As explained in the introduction to agent-based artificial intelligence above (Section 2.2.1), the agent-based approach was formulated as an alternative research methodology. Researchers such as Brooks (1991*a*), Steels (1994), and Horswill (1997) suggest that there is no other way likely to create "true" intelligent behaviour than through their complete agent approach. However, this reformulation has been aimed at the level of development. Little consideration was given to the issue of *scientific* methodology by the early researchers in this field; work in this area is still in its infancy (Hallam & Hayes 1992, Laird 1997, Nehmzow et al. 1997, Wyatt et al. 1998).

Another premise of this dissertation is that a reasonable way to fill this methodological void is to adopt the scientific practices of experimental psychology. This is discussed in Chapter 4, and has been adopted by some members of agent based community, (e.g. Webb 1996, Sharkey 1998). A basic consequence of this approach would be that progress or relative merit should be evaluated in terms of statistical measures of observed behaviour, by use of hypothesis testing. The traditional standard methodology in artificial intelligence has been the demonstration by prototype that a concept or development approach succeeds.

The emphasis on demonstration is not surprising if one considers the assertion made in Section 2.2.2, that AI researchers attend more to the selected software behaviour than actual expressed behaviour. Computers are built to be completely deterministic systems. Determinism is a characteristic of traditional logical systems, and logic has often been perceived as the ultimate rationality, and consequently the and goal of intelligence. Thus in traditional software design, the emphasis is to develop algorithms that can be proven to operate correctly in all possible circumstances.

One reason why this approach fails in artificial intelligence is that in AI domains it is more difficult to categorise and account for all possible circumstances than in conventional domains of mathematics and computer science. The situation is analogous to difference in difficulty in establishing controls for psychology vs. chemistry or physics. The problem domain is far richer and the systems studied the far more complex. The successful categorisation of environmental circumstances is in fact one of the tasks of an intelligent system, and our inability to explicitly account for these is one of the motivations for creating artificial intelligence. This problem has been demonstrated formally by Chapman (1987), who has proved that optimal decisions about action selection cannot be made by an agent that has finite resources of either time or memory.

The fact that the difficulty and reliability of both perception and action are major aspects of the task of intelligence has only recently come to be understood within AI (Ballard et al. 1997, Horswill 1997). Historically, most AI work has been performed in an environment where not only is perception provided, but actions are guaranteed to be executed as intended. For example, a chess playing program simply states its next move, which it then presumes to be canonical. Any information entered into the typical chess program that would indicate an incorrect move had been made accidently would be rejected as an illegal move on the part of the opponent.

In addition, any information in a central knowledge base used for making planning decisions is often guaranteed to be true and internally consistent. In fact, traditional artificial intelligence reasoning systems are incapable of storing or using contradictory information, and centralised systems for dealing with such information, such as non-monotonic reasoning systems and truth-maintenance systems, are still major areas of AI research. See for example any major AI conference proceedings, (e.g. IJCAI or AAAI of 1999).

As described in the previous section on behaviour (page 15), agent-based AI, particularly robotics, has a considerably different set of assumptions. Neither sensing nor action can be expected to be accurate. Different behaviours may hold contradictory "knowledge" about the world. To continue the example from that section, the obstacle-avoiding behaviour may be said to "know" going forward is bad, while the light-seeking behaviour may "know" it is good. Further, the obstacle-avoiding behaviour as described has no way of knowing whether the obstacle encountered was stationary or mobile, so its knowledge may easily be ill-founded. If the obstacle is mobile and has moved away, going forward may be neutral as far as the obstacle-avoiding behaviour should be concerned, not bad.

In summary, agent-based systems utilise algorithms which are not provably correct or even consistent, because the problems of operating in the real world are too complex to solve deterministically. Consequently, evaluation of such a system invites the methods for the testing of theories of behaviour used in psychology. These techniques were designed for studying animals, which are similarly non-deterministic, possibly for similar reasons. The need for creating a science of describing animal behaviour led to the development of psychological method. In general, however, such methodologies are not yet widely used within agent-based AI. This may be a consequence of not having sufficiently rethought the traditional approach to AI from which agent-based research emerged.

Robotics-based research also has another pressure against proper experimentation — it is difficult to achieve significant numbers of tests in robotics because the robots themselves are expensive, slow, and unreliable. Robots often physically degrade over the period of an experiment, because most of their components are not engineered to withstand protracted shaking or frequent collisions. In this respect as well they are like animals, requiring careful management of research schedules and specialised technicians. They also share the risk of unexpected delays and outright loss of individuals prior to completion of testing. Thus, the history and expectations by both researchers and funders of AI as an engineering subject, and the lack of a history of the discipline for research found in the animal sciences have slowed the progress of scientific methodology in the field of autonomous robotics. These issues and their implications for psychologists using agent-based AI are further discussed in Section 4.4.

2.2.6 Scaling and Complexity

As explained in the previous section, much artificial intelligence research can be characterised as existence proof, or more cruelly "Look ma, no hands" research (Simon 1995). Yet the assumption that, if an algorithm can be shown to work, then it can be expected to work reliably, has empirically often proven to be false. An algorithm that works in a particular domain may not work in another with different features, or more complex arrangements of the same features. A significant hurdle for any successful artificial intelligence system is whether it can operate equally successfully when there are simply *more* of the same features or problems. This is called the **scaling problem**.

The scaling problem is now recognised as a primary issue for artificial intelligence algorithms. It is in some ways the inverse of one of the classic problems of comparative psychology: the question of whether intelligence or at least learning is homologous across all animals, varying *only* in scale, or whether it employs qualitatively different mechanisms in different species. (See for example (Livesey 1986) for a review of psychological research in this domain.) AI being a science of production, the question is whether any particular known algorithm *can* be so scaled. In other words, is it feasible to significantly increase the scope of a capability by simply increasing the capacity of the system that provides it.

For a simple example of a system that doesn't scale indefinitely, consider the size of animals. One limit on how large an animal can be is bone strength. The weight of an animal increases as the cube of its height or length, but bone strength only increases as the square of the width of the bone. Consequently, an animal's bone mass has to increase faster than its length as an animal increases in size. Clearly, this sets limits for overall size. This constraint is a greater problem for flying animals since bones contribute significant weight, and less of a problem for swimming animals since their mass is supported more evenly by their environment.

These problems are sufficiently central to not only artificial intelligence, but computation generally, but they have become a major area of research in both mathematics and computer science. Significant progress has been made in the last few decades at describing both the difficulty of problems, and the power of algorithms. The metric of this difficulty is called **computational complexity.** These methods are now applied as matter of course to the analysis of AI systems. This section serves as an introduction to complexity theory, for a complete introduction to the theory of computation see Sipser (1997).

To give examples of different levels of complexity, we will consider the problem of recognising human faces. Assume that a face can be characterised as a large number of features that can be reliably read correctly[2], and a large number of faces have been stored in some form of knowledge base. One algorithm for recognising a new picture of a face would be to search each face in the database and determine which one is the closest match. This is a **linear** time algorithm, dependent on the number of known faces. That is, the amount of time it would take is directly proportional to the number of faces in the database.

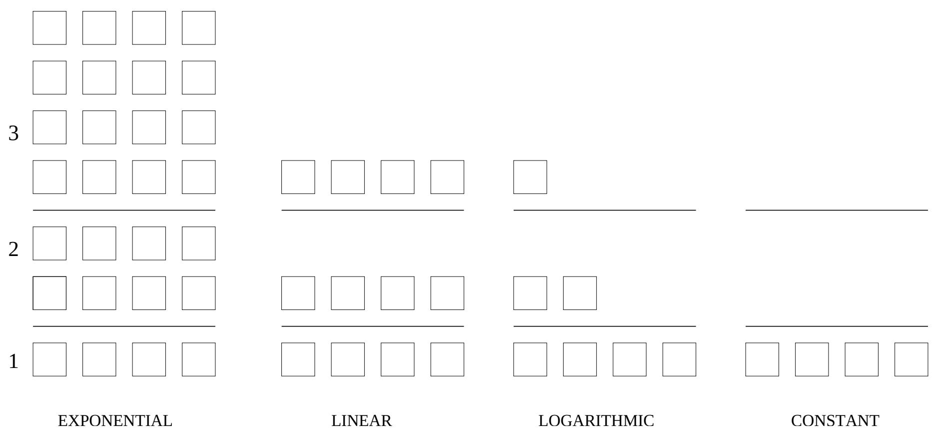


Figure 2.2: Different types of complexity. N is the dependent variable—for example, the number of faces needed to be recognised. Blocks represent resources (e.g. time, space, number of nerve cells) and levels represent *additional* resources required for an increase of N. Resources required for N = 1 are given to be 4 for all four conditions.

A faster algorithm would be one that splits the database depending on the value of a particular feature, for as many features as is necessary to find a unique answer. For example, if the first feature were gender, and the target picture is of a woman, approximately half of the database would not have to be searched. If the second feature were hair colouring, of which there were five possible values for the feature, than a further four fifths of that half could be ignored, and so on. This algorithm is better than linear (technically, it is **logarithmic**) but it is still dependent on the number of stored faces, since presumably the more faces there are, the more features will be needed to completely discriminate between them. However, this algorithm would be said to scale better than the previous, linear algorithm, at least with respect to the

[2] This is a very unrealistic assumption. Face recognition is also a significant area of current research in artificial intelligence. Most face recognition programs require uniform location and lighting constraints for the picture to be identified, then perform principle component analysis across a collection of such images to determine automatically the salient features (e.g. Brunelli & Poggio 1993, Dailey & Cottrell 1997).

number of faces known.

A rearrangement of the previous algorithm can be made to work so that amount of time to identify a face is **constant** with respect to the number of faces memorised. Suppose that we have established a list of features actually needed to discriminate all of the people in our database by the method above. Suppose also that for any face, the value of its features could be expressed numerically. Finally, imagine that each face can be uniquely identified by performing a mathematical function on the value of each of these features for that face. That is, the value of the function is different for any two faces. Then, a new face could be recognised simply by applying this function to the face's features.

The time of the above strategy for recognition is **constant** for (or *independent of*) the number of faces, but it is linearally dependent on the number of features required for the computation. This dependency is less important than the former, for two reasons. First, the first two algorithms are also dependent on the number of features needed, so this dependence is constant across all three algorithms, and therefore not a factor in discriminating between them. Secondly, as explained in the previous algorithm, the number of features needed increases only as a logarithm of the number of faces stored. Or, to restate the same concept from a different perspective, for every new feature, there is an **exponential** increase in the number of faces that can be represented. This is because any one new feature can be potentially combined with each former feature, thus it represents not only one new face, but as many new faces as there were formerly features.

The descriptors listed above; constant, logarithmic, linear, and exponential; are examples of complexity values used to label both algorithms and problems. A problem is said to be of the least complexity for which an algorithm is known to solve it. For example, a problem would be considered of linear complexity if a linear algorithm is known that can solve it, but no such logarithmic or constant algorithm is known.

Any agent in the real world needs to worry not only about complexity in time, but also in storage space and/or computing resources. An example of an algorithm that is time efficient but storage expensive is the "grandmother cell" theory of face recognition. In this theory, different brain cells represent different people or objects, and are able to uniquely identify whether a face is or is not recognised by them. The algorithm is constant time, since however many faces are stored, the cells all attempt recognition concurrently. However, the number of brain cells required to do this computation is linear on the number of people known.

2.3 Can AI Clarify Psychology?

Given the substantial differences in perspective, language and task described in the previous section, can artificial intelligence have any bearing on psychology? The hypothesis on which this dissertation is based is that the problem of intelligence has sufficiently universal characteristics that the progress made in understanding the construction of an artificially intelligent system can shed light on the study of naturally occurring intelligent behaviour (see for discussion McGonigle 1991, Hendriks-Jansen 1996). One can go further and suggest that many of the discrepancies between AI and psychology are actually the result of AI discovering and reducing ambiguities that had been latent in the original psychological theories. Artificial intelligence has been forced by the discipline of engineering to have clear, precise models of potentially ambiguous concepts because a complete, constructed system must have functional elements. These clarifications may be incorrect; AI may have focussed on the "wrong" aspects of behaviour, development or complexity from the perspective of psychology. But they are at least concrete hypotheses; as such, they should be tested.

One of the premises of this dissertation is that AI can provide not only a platform for testing hypotheses, but also for developing them. The rest of this section illustrates this approach on a high level. The following chapter will provide a detailed example with respect to the dissertation's central thesis, concerning the sequential and hierarchical nature of action selection.

2.3.1 Goals and Intentions

One of the first concepts introduced in the previous section was a simplified definition of intentionality. Often a behaviour or set of behaviours will have a terminating condition. For example, a foraging behaviour terminates when food has been found, an eating behaviour when the agent is satiated. These terminal conditions are often referred to as **goals** for the behaviour. They relate to the notion of consummatory behaviour in ethology, but notice that the specification above is of an extent, rather than a behaviour: e.g. consummation is achieving at least a threshold blood-sugar level rather than simply eating. This is another example of the

AI emphasis on internal rather than external perspective.

This set of language and perspective provided by AI allows us to ask psychologically relevant questions. For example, is it accurate to portray behaviour as having goals or terminal conditions? A simple alternative model would be to have behaviours persist until they are interrupted. In the above example, another goal could come to be of higher priority than eating as blood-sugar level increased. Examples of such an architecture in psychology is Hull (1943), or in AI Maes (1989), described below in Section 3.5.1.

As stated earlier, currently active goals (or just active software behaviours) are sometimes called intentions (Bratman et al. 1988). Translating such terminology into the common psychological meanings of intention can be problematic. For example, the obstacle avoiding behaviour described in the previous section is quiescent whenever the robot is neither hitting something nor near to where it has recently hit something. Would we say the robot only intends to avoid obstacles directly after it has hit one? Or, since the behaviour is constantly active at least to the level of checking its sensors, do we say that the robot always intends to avoid obstacles? It may seem that such a definition removes the utility of the term "intention." However, the word denotes a useful distinction, between actions that consummate a behaviour module, and those which do not. Thus, if the robot did in fact strike an obstacle while pursuing a light, the striking could be said to be a consequence of the agent's behaviour (particularly the move-towards-light behaviour) but not a part of its intent, whereas moving towards the light, or backing away from the obstacle after striking it are both intentional acts.

The common psychological definition of intention has something to do with conscious decisions, but consciousness is itself poorly defined, and its relation to intentional behaviour still not fully understood (Dennett & Kinsbourne 1992). While not denying the utility of the concept of declarative intent, that is, intention as stated by the subject, it has been obvious since the beginning of psychology, since James, Freud and even Hume, that there is also a need for a weaker understanding of goal and intention to describe meaningful behaviour that arises without declarative selection. Norman & Shallice (1986), for example, give a psychological account linking deliberate and automatic behaviour (including misordered behaviours) as slightly different control operating over a single distributed behaviour system. Their model is similar to some of the AI action control mechanisms described in this dissertation; it is discussed further in Section 5.3.3.

2.3.2 Rationality and Behaviours

Similarly, the behaviour-based approach to intelligence is at odds with logic-based models of rational behaviour, where a single consistent set of rules is applied across data to arrive at the conclusions. Although this is still the folk-psychological understanding of at least declarative knowledge and reasoning, it has been well established that human decision making is not necessarily logically consistent (e.g. Tversky & Kahneman 1981, Bacharach & Hurley 1991), nor do we apply the same computational skills and strategies in different situations (Cosmides & Tooby 1992). Even such apparently logic-based abilities as transitive inference have been demonstrated to be not "high level abstract thought" but "basically a robust biological mechanism dedicated to rational choice control" (McGonigle 1991). The ability to make transitive choice (if $A > B$ and $B > C$, then $A > C$) has been demonstrated both in monkeys (McGonigle & Chalmers 1977) and in children who were still incapable of the task of sorting small numbers of blocks by size (Chalmers & McGonigle 1984). From a logical perspective, being able to order a full list should precede or coincide with being able to order elements within it, particularly across transitive pairs.

Such results are typical of the findings that have led to what has been called "the contextual revolution of psychology, sociology and the philosophy of mind" (Hendriks-Jansen 1996). The behaviour-based approach of localising reasoning and computation patterns to modular behaviours is one abstraction for modelling such results. It is not a perfect model; for example it is difficult to imagine how a fully modularised system might model the ability of analogy — of applying some part of the form of a solution from one domain into another. Such difficulties with behaviour-based models are addressed further in Section 5.3.4.

A more accurate modelling of cognitive skills (as opposed to reflexes) would probably be distributed, as suggested by most neuroscience since Lashley (1950). A skill probably results from the bound activation of a sufficient number of nerve cells or cell assemblies, their activation triggered by context set both externally through perception and internally by cognition, which settle to established patterns. If the same nerve cells are assumed to serve in many different configurations, this mechanism could more easily explain problem solving via analogy than a fully modular system. Further, the fact that context, whether triggered by perception of internal or external events, is set by mechanisms affected and recategorised by learning, allows this sort of distributed model to account for insight and other such forms of cognitive dynamics. See Calvin (1996) for an example of a model along these lines.

Unfortunately, controlling such representations precisely enough to result in complex high-level behaviour is beyond the current capabilities of the parallel distributed processing (or artificial neural network) approaches to AI. Consequently, the choice within active experimental systems for representing behaviours is between the overly-modular simulations in behaviour-based AI, and the under-modular traditional single-database and reasoning systems. As argued above, the latter are simultaneously overly powerful and insufficiently biased to represent natural intelligence. Behaviour-based systems may be too modular to accurately represent development, cognizant learning or representational redescription (as in Karmiloff-Smith 1992). However, they can be used to model snapshots of skilled behaviour, and also specialised learning, which appears to account for the vast majority of natural adaptivity (Gallistel et al. 1991). In the case of specialised adaptation, the representational substrate for this learning can be contained within a single behaviour, thus the modularisation would not be a problem (Bryson & McGonigle 1998). This strategy was used in the robot experiments described in Section 4.2.

2.3.3 Context and Complexity in Behaviour

As stated above, the extent to which behaviour is determined by context or environment is coming to the fore of many fields of research, for a recent review see Hendriks-Jansen (1996) or Clark (1996). In artificial intelligence, this idea was first proposed by Simon (1969). The complexity of behaviour may be to some extent merely the reflection of the complexity of the environment in which it is situated, the computation involved in the behaviour itself might be simple. Simon's example was of the path described by an ant walking on a beach. Describing the path mathematically, full of intricate curves in three dimensions, would require an incredibly complex equation. The ant, however, is no doubt obeying just a few simple rules determining its heading and its finding footing across the convoluted terrain of pebbles and sand. Braitenberg (1984) illustrates that another source of complexity in expressed behaviour may be the simultaneous interaction of even a very few, very simple, internal behaviours.

The hope that animal-like complexity of behaviour can be created relatively simply via these two mechanisms is central to much of the research in agent-based artificial intelligence (Brooks 1991*a*, Malcolm et al. 1989, Brooks & Stein 1993). Multiple parallel simple behaviours, each reflecting through reaction the complexity of their environment, do in fact control some of the most convincingly animal-like of robots (Brooks 1990) and virtual reality agents (Sengers 1998). They have also been shown to perform some human cognitive tasks such as playing video games (Agre & Chapman 1987) or accompanying folk music (Bryson 1992). These latter examples readily illustrates Simon's ant metaphor. Pengi, a video-game playing program, reacts to the complexity of the many rewards and hazards of its video environment. The Reactive Accompanist reacts to the patterns of time and pitch in a melody in order to determine which accompanying chords should be played when. The extent to which human and animal behaviour can be successfully modelled with such systems gives us some evidence, at least as proof of possibility, that natural intelligence might also be composed of such elements.

Reducing the complexity of the sorts of behaviour expected to be produced by an intelligent agent is only one mechanism for reducing the overall complexity of action selection. As outlined above (Section 2.2.6), complexity is also highly dependent on the sort of algorithm applied to a task. For action selection, the complexity problem is fairly straightforward. The number of possible orderings of behaviours increases exponentially on the number of possible actions. Any action ordering system that depends on allowing for all possible orderings is consequently unlikely to scale successfully. Examples of such systems include Maes's and Tyrrell's, described in the next two chapters. A hierarchical organisation reduces action selection to logarithmic complexity since actions are clustered; they can only occur in a small number of well-delineated contexts. The combinatorial utility of hierarchy is consequently universally recognised (Dawkins 1976, Hendriks-Jansen 1996, McGonigle & Chalmers 1996). The main question addressed by this dissertation is whether the cost in terms of behaviour flexibility is too high for structured control to be psychologically plausible.

2.4 Common Problems Between the Fields

The remaining chapters of this thesis are dedicated to exploring one hypothesis; that the sequential and hierarchical structure of the control of action selection does not necessarily prevent an agent from reacting appropriately and opportunistically in a dynamic environment. This hypothesis is one example of a problem shared by both AI and psychology. This section details some other problems shared between the fields. This serves both as further evidence that the fields are mutually relevant, and as hints of other potentially rich sources of research to be developed between the two fields.

2.4.1 Top Down vs. Bottom Up Control — Action and Perception

To what extent is behaviour the result of cognitive decisions, and to what extent is it the result of reflexive reactions to sensory events? The early behaviour-based roboticists, like some psychological behaviourists before them, attempted to drive as much behaviour as possible simply from the bottom up, as reflexes. They called this **reactive control**, a strategy discussed further in Section 3.3. In the terms of the behaviourists, this was "connectionism[3]," a suggestion Lashley (1951) strongly repudiates:

> [I]nput is never into a quiescent or static system, but always into a system which is already actively excited and organized. In the intact organism, behavior is the result of interaction of this background of excitation with input from any designated stimulus. Only when we can state the general characteristics of this background of excitation can we understand the effects of given input. (p. 506 of Beach et al. 1960)

Of course, in actuality any reflexive system *is* necessarily already organised. The results of sensory input into any behaviour-based system is not arbitrary, but carefully designed and highly constrained (Brooks 1991*b*). This is a form of top down control, though provided by evolution in animals or a designer in artificial agents. What reactive AI researchers have often overlooked is the importance of expectation and selective attention, the "active excitation" mentioned above. Perception itself is an impossibly under-constrained problem without top-down information (see for discussion Barlow 1994). This claim is evidenced in biology, where even the LGN has more reciprocal connections coming *from* the visual cortex than "forward" connections going *to* it, or than it gets from the retina (Sherman & Koch 1990, Sillito et al. 1994). Recent work in machine vision (e.g. Rao & Ballard 1996) supports the hypothesis that "higher" layers in the cortex can learn to provide expectations that make low-level receptors more useful. This evidence is discussed at length in Chapter 5.

[3] Modern AI uses the term "connectionism" more loosely to describe any theory of intelligence rooted in distributed / neural network control rather than symbolic control. Ironically, Lashley himself strongly supported distributed representation of motor skills as favoured by modern connectionists (Lashley 1950).

One of the questions to explore and model in autonomous agent-based research, then, is the tradeoff between the complexity of the high level, top-down system and the complexity and accuracy of the sensing system required to drive the system from the bottom up. Another is the tradeoff between alertness — the ability to react to the environment, and persistence — the ability to complete even complicated courses of action.

2.4.2 Reasoning vs. Pattern Matching

This issue is closely related to the description of preformed behaviours as top-down control just proposed. The question here is whether rational behaviour is best characterised as constructive, that is, reasoned through some logic-like system to create new plans on demand, or simply selective, where plans are picked from a library of pre-programmed behaviours. If the first is the case, then one of the most important facets of intelligence is how the "logic-like" system works. If the latter, then the primary problem becomes how we recognise which plan is appropriate for the situation. Case-based reasoning is a branch of AI entirely devoted to studying this latter issue (e.g. Hammond 1990). As explained in the previous section on rationality, the lack of common reasoning strategies would make the pattern-matching theory of intelligence appear likely. Even novel situations may be solved by the application of rules at a more general level of granularity, or with relatively simple adaptations. However, such a system could be used to describe logic itself, bringing the problem in full circle.

The question this suggests for agent-based research is how much of "reasoning" is best modelled as looking up patterns, and how much is best modelled via construction. The applicability of such research to animals might seem questionable considering the vast differences (serial vs. parallel) in computation, but this actually affects both strategies equally, since constructive planning is also a form of search (Winston 1992).

2.4.3 The Nature of Categorisation

To the extent that intelligence is pattern matching, that is, recognising a situation in which an action should take place, then to that extent categorisation is a critical problem for understanding behaviour. This is again central to the problem of selecting an appropriate plan. Psychology is concerned with researching how one situation or object is judged as equivalent to another (Bruner 1990, e.g.), and AI requires the ability to judge such an equivalence.

2.4.4 Nature vs. Nurture

In an artificial agent, this question reduces to how much should be programmed by the developer, and how much should be learned by the agent. Considered as such, this is every bit as central a debate to agent research as to psychology.

2.4.5 Learning vs. Development

Again, in AI this is rephrased to a constructive question: how much can be learned in provided representations and processes, and to what extent do learning and representation need to be learned or provided in stages. These issues are discussed briefly both above, in Section 2.2.4, and below in Chapter 5.

2.4.6 The Nature of Representation

This problem can be seen as a summary of the previous five. Understanding exactly what information an agent stores and in what form is roughly equivalent to knowing the nature of the processes that exploit it. Is knowledge a bare shell of procedures made rich by sensing, or is the majority of perceptual experience already latent in the adult brain? If the former is true, then processing must be mostly bottom-up, the latter would indicate a more top-down bias. Similarly, if knowledge is mostly simple general rules, then we probably construct plans by reasoning, whereas if it is a large number of detailed scripts, frames or cases, then much of intelligent behaviour is done by recall. How we classify and acquire this information is obviously tightly related to how we represent it.

One great axis of research into representation is whether it is consistent across intelligence, or whether representation may be specialised. Modular or behaviour-based views of intelligence are conducive to specialised approaches, while single-process theories would indicate single representations. Notice also that both theories may be true at different levels of abstraction. There may be a nearly homogeneous process and storage theory at the neuron level of intelligence, though there are many highly differentiated types of neurons in the human brain (see for discussion Elman et al. 1996).

2.5 Conclusion

An essential premise for the relevance of this dissertation to psychology is that some aspects of animal behaviour can be explored via their modelling on artificial platforms. Further, since this approach to studying intelligence has been actively pursued for some time, at least some of the research already done should be applicable to psychology. This chapter has introduced the terminologies and problems of research in artificial intelligence considered most relevant to the aspects of psychology explored in this dissertation. Those aspects are the thesis problem of mechanisms for selecting or organising behaviour. The central term to be explained for such an application is *behaviour*. This chapter has described the difference between the constructive and the descriptive stance in terms of the understanding of behaviour, and the problems that result from this — the problems of design and development, of research methodology, and of demonstrating the scaling abilities of an algorithm.

This difference in perspective between AI and psychology does not necessarily imply the results of AI are relevant to psychology, but the fact that they may result in a source of psychological hypotheses. The second part of this chapter has demonstrated the commonality between the concerns of agent-based artificial intelligence and psychology by showing that the progress made in agent-based AI in defining the roles of intentionality, rationality, context and complexity do have relevance to current psychological research. The chapter concludes by delineating a number of current active research problems that both fields share. The hope is that this list will serve as inspiration for future work along the lines of this dissertation.

The next chapter moves from this background into a more specific exploration of artificial intelligence mechanisms of action selection, and presents the main model used for the experiments presented in the chapter following.

Chapter 3

Thesis: The Nature of Action Selection

3.1 Introduction

The central aim of this dissertation is the establishment of a single hypothesis: that natural, appropriate and alert behaviour can be exhibited by agents executing actions prestored as hierarchies and sequences. As detailed in Chapter 1, this thesis has been a basic assumption of a great deal of research in both natural and artificial intelligence (see page 2). However, recent research has called into question this assumption. The main objection to this hypothesis has been that allowing for such structures seems to necessitate some form of centralised control. Hypothesising such control is seen as recursive or homuncular: what executes the executive? What organises the behaviour of the organiser? Further problems include the frequency of mistakes made in even familiar, well-rehearsed sequences of actions, and the fact that some behaviours do not occur before or after each other, but in fact are expressed simultaneously as one fused behaviour (See Figure 3.1).

The alternative hypothesis offered is that the apparent structure of behaviour is in fact emergent. That is, the structure is not provided by any explicit means, but arises incidently as a consequence of other, more essential factors. Common metaphors for this concept of emergence are the harmonic and rhythmic complexities that emerge from a symphony, even though every individual member of the orchestra may play a relatively simple part, or the beauty that emerges from the brush strokes and paint of the Mona Lisa.

This argument for emergence is given by two related but to some extent antithetical perspectives. Some support comes from researchers presuming the modularity of intelligence.

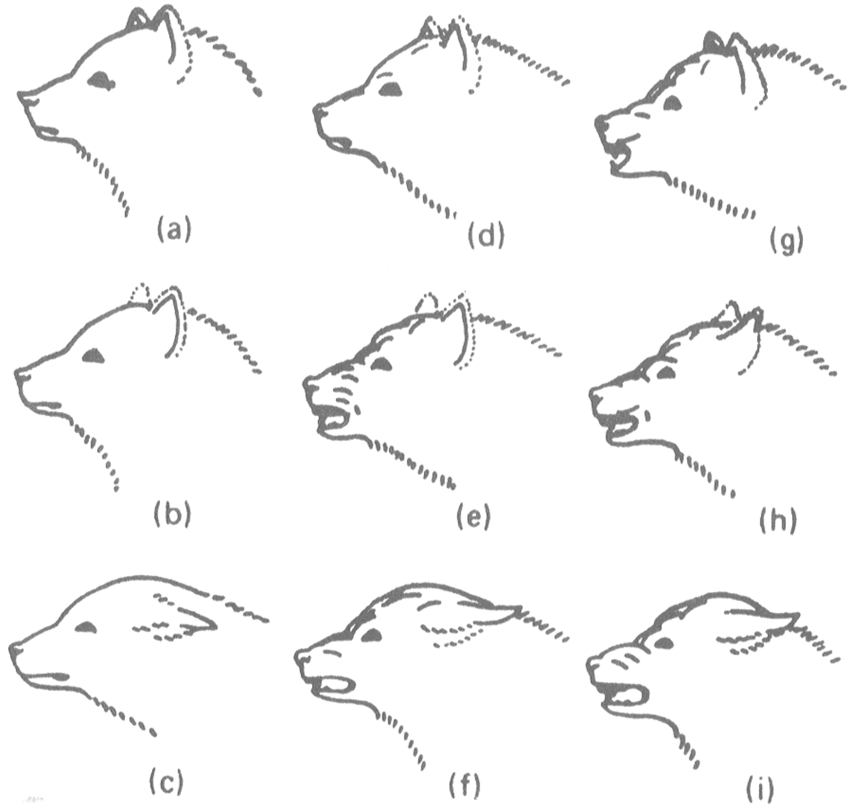


Figure 3.1: The canonical example that behaviour expression is a continuum, and can result from multiple sources (from Lorenz 1973 referenced in Tyrrell 1993, p.187).

Currently this is much of the behaviour-based AI community; earlier this century it was the behaviourists. Modularists dismissing structured control prefer to believe that each modular unit determines its own execution time, using its own perceptive abilities. Several agent architectures demonstrating this viewpoint are discussed below, but see for reviews Brooks (1991*a*), Maes (1990*b*) and (Blumberg 1996). The other source of support for emergence over explicit structure comes from researchers who favour dynamic systems theory. This field, arising from physics and mathematics, is highly related to chaos theory. In its most extreme form, dynamic systems theory resists thinking of any of the universe, let alone intelligence, as segmented or modular. The resistance to hierarchy and sequence is that such organisation implies segmentation of process. In their view process is continuous (Goldfield 1995, Bohm 1980). Hendriks-Jansen (1996) draws on both modular and dynamic backgrounds in making the analysis quoted in the introductory chapter as the challenge to this thesis (page 5). Similarly, the introduction of this dissertation referred to both paradigms as dynamic theories of intelligence, since they share the attributes of decomposing traditional theories of intelligence, and explaining apparent coherence as emergent phenomena.

Of course, both of these dynamic communities have been forced to address the issue of complex sequences of behaviour. The behaviourists initially believed that all behaviour sequences are learned as chains of conditioned responses. This hypothesis has been disproved. The behaviourists themselves showed both that reward fails to propagate over long sequences and that complex structures such as mazes are learned through latent mechanisms which require no external reward (see Adams 1984, for a review and history). Further, neuroscience has shown that the fine coordination between different elements of behaviours such as speech and piano playing occur too rapidly to allow for triggering of one act by its predecessor (Lashley 1951). Dynamic theory has been more successful at describing rapid, integrated sets of behaviour, provided these can be described as a part of a rhythmic whole. For example, swimming and walking can be explained and controlled through understood mathematical mechanisms providing there exists a pattern generator to keep the rhythm of the cycle (Goldfield 1995, Reeves & Hallam 1995). Such oscillatory pattern generators are basic neural structures found in all vertebrate life (Carlson 2000). However, the structuring of more complicated, heterogeneous behaviours (such as nest building or starting a car) has not yet been successfully addressed, and the significantly greater combinatorial complexity such heterogeneity opens up is often either ignored or dismissed (e.g. Bohm 1980, p.182 or Goldfield 1995, pp. 285–288).

It can be argued that emergence is merely a phenomena of the variety of levels of abstraction. Higher levels always “emerge” from lower ones, but this does not remove the possibility that the higher levels have their own organisational and computational properties (see for discussion Newell 1982). The firing of nerve cells can, for example, be described either at the molecular or the neural level of abstraction, but the behavioral consequence of that firing is more reasonably addressed at the neural level than the chemical. The brain is certainly a highly distributed, highly interactive computational device, and dynamic systems theories better describe both the fluidity and influence on each other of expressed actions as they operate in sequence and in parallel. Common examples of these influences include the effect of emotion on gait, situation on memory, and the influence of both preceding and following phonemes on an interval of speech. However, modular and hierarchical models of behaviour are not only useful as descriptions, but have significant biological evidence (see Chapter 5) . The hierarchical, modular structure of the visual and motor cortices for example has been generally accepted.

The purpose of this dissertation is to demonstrate that hierarchy and sequence are not antithetical to intelligence. This chapter advances the argument by reviewing the artificial intelligence agent architecture literature, both contrasting the relative merits of individual architectures, and looking at the development trends of particularly successful architectures. Both of the strains of evidence indicate that structured control is both useful and necessary. Further, this chapter introduces the specific agent architecture, Edmund, that will be used in the experiments of the following chapter. Edmund is introduced in terms of the specific hypotheses on the nature of intelligent action selection it embodies compared to other architectures.

3.2 Architectures as Models of Intelligent Control

As mentioned briefly in Chapter 2, an architecture is a design scheme by which an agent's intelligence is created (see further Wooldridge & Jennings 1995). Different architectures therefore embody different approaches to intelligence. These differences are primarily differences in decomposition, which reflect various axes of discrimination, such as:

- **Specialisation vs. Generality**: for example, can we consider all perception as a unit, or should the different sense modalities be modelled separately?
- **Process vs. Data** for example, is the behaviour for moving the legs during walking planned as one walks, or is that behaviour encoded in memory?

A traditional architecture for both psychology and artificial intelligence is shown in Figure 3.2. This architecture indicates that the problems of intelligence are to transform perception into a useful mental representation R, apply a process to R to create R', and transform R' into the necessary actions. This model has led many intelligence researchers to feel free to concentrate on only a single aspect of this theory of intelligence, the process between the two transformations, as this has been considered the key element of intelligence.

This model in Figure 3.2 may seem sufficiently general as to be both necessarily correct and uninformative, but in fact it makes a number of assumptions known to be incorrect. First, it assumes that both perception and action can be separated successfully from cognitive process. However, perception is guided by expectations and context — many perceptual experiences

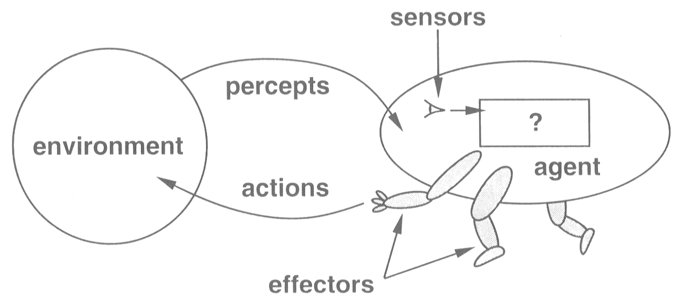


Figure 3.2: A traditional AI architecture (from Russell & Norvig 1995).

cannot be otherwise explained). Further, brain lesion studies on limb control have shown that many actions require constant perceptual feedback for control, but do not seem to require cognitive contribution even for their initiation. This research is detailed below in the chapter on biological evidence (see Section 5.2.1, p. 106 and p. 107, respectively).

A second problem with this architecture as a hypothesis of natural intelligence is that the separation of representation from cognitive process is not necessarily coherent. Many neural theories postulate that an assembly of neurons processes information from perception, from themselves and from each other (e.g. McClelland & Rumelhart 1988). This processing continues until the activations of the neurons settles into a recognised configuration. If that configuration involves reaching the critical activation to fire motor neurons, then there might be only one process running between the perception and the activity. If the levels of activation of the various neurons are taken as a representation, then the process is itself a continuous chain or re-representation[1].

3.3 Behaviour-Based Architectures

As first introduced in Section 2.2.2, behaviour-based architectures assume that a more useful and accurate way to model intelligence is to model behavioural skills independently of each other. The extreme view in this field is that intelligent behaviour cannot emerge from thinking and planning, but rather that planning and thinking will emerge from behaving intelligently in a complex world.

My feeling is that thought and consciousness are epiphenomena of the process of

[1] Notice that the idea of a stopping point is artificial — the provision of perceptual information and the processing activity itself is actually continuous; the activations of the motor system are incidental, not consummatory.

being in the world. As the complexity of the world increases, and the complexity of processing to deal with that world rises, we will see the same evidence of thought and consciousness in our [behaviour-based] systems as we see in people other than ourselves now. Thought and consciousness will not need to be programmed in. They will emerge. (Brooks 1991*a*)

Though traceable in philosophy at least as far back as Hume (1748), and in psychology as far back as Freud (1900), the notion of decomposing intelligence into semi-autonomous independent agencies was first popularised in AI by Minsky (1985). Minsky's model incorporates the novel idea of multiple **agencies** specialised for particular tasks and containing specialised knowledge. Minsky proposes that the control of such units would be easier both for a species to evolve and for a system to learn than the more complex control required for a single monolithic system. He also argues that such a model better describes the diversity and inconsistency of human behaviour.

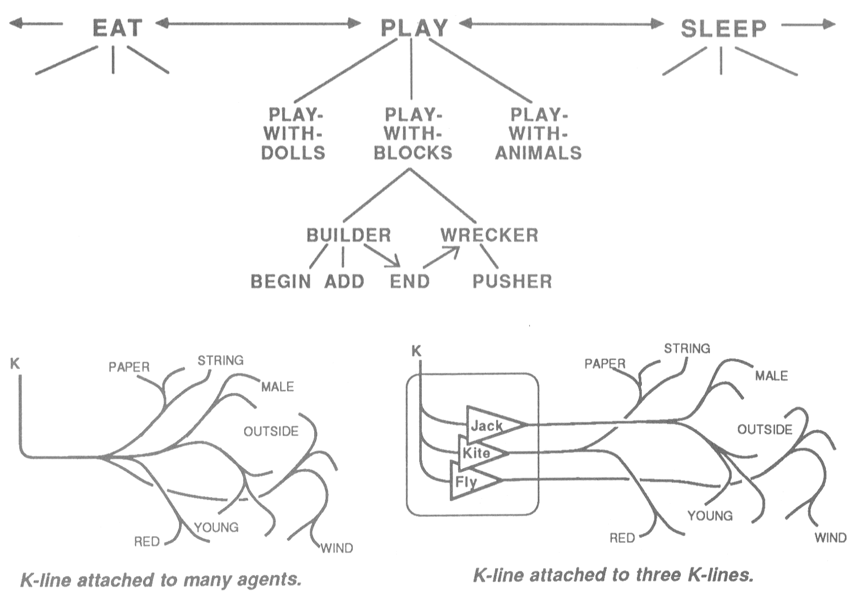


Figure 3.3: A "society of mind" architecture for a child playing, and example of k-lines (from Minsky 1985, pp.32 and 89).

Minsky's "agents of mind" are hierarchical and only semi-autonomous. For example, he postulates, a child might have separate agencies for directing behaviour involving sleeping, eating and playing (see Figure 3.3.) These compete for control, and when won, their subsidiary agencies in turn compete. Once playing is chosen, blocks compete with dolls and books; building and knocking down compete within playing-with-blocks. Meanwhile, the agency in charge of eating may overwhelm the agency in charge of playing, and coherent behaviour may be interrupted in mid-stride as different agencies swap to take control.

The cost of theories that successfully explain the incoherence of human thought and activity is that they often fail to explain its coherence. Minsky addresses this by postulating a modular rather than a completely distributed system of thought. He explains coherent behaviour as being the output of a single agency or suite of agents, and incoherence as a consequence of competing agencies. He also recognises that there can be coherent transitions between apparently modular behaviours. To address this, he postulates another structure **k-lines** which connect modules associated in time, space or entity. He also posits fairly traditional elements of knowledge representation, frames and knowledge hierarchies, for maintaining databases of knowledge used by the various agents.

Brooks, quoted above, took modularity to a greater extreme when he defined the behaviour-based movement in AI (Brooks 1986, 1991*b*,*a*). His theories reflect a major difference in approach from Minsky, in that Brooks was attempting to build running systems with demonstrable efficacy, rather than solely theories. Part of the reason for the enormous popularity of his approach was his success in this regard. Brooks' laboratory and students produced the first robots capable of moving through unaltered office environments at animal-like speeds[2], (Brooks 1990). Further, these robots could manipulate their environments exhibiting apparent goals, such as collecting empty soda cans (Connell 1990), giving tours (Horswill 1993*a*), or coordinating with other robots (Parker 1998, Matarić 1997).

In Brooks' model, **subsumption architecture**, each module must be computationally simple and independent. These modules, now referred to as "behaviours," were originally to consist only of finite state machines. That is, there are an explicit number of states the behaviour can be in, each with a characteristic, predefined output. A finite state machine also completely specifies which new states can be reached from any given state, with transitions dependent on the input to the machine.

For an example, consider a simple two-wheel robot similar to those described by Braitenberg (1984). Each robot also has two light sensors, left and right, corresponding to the left and right

[2] Behaviour-based robots navigating by vision (with cameras) can move at speeds of over a meter per second (Horswill 1993*a*).

wheel of the robot. To make an intelligence that would move the robot towards light would require two identical behaviours, one connected to the left light sensor and the right wheel, the other to the left sensor and right wheel. Each behaviour requires two states, corresponding to whether its motor is on or off, and two transitions, depending on whether the sensor senses light. If a sensor sees light, it should rotate its wheel forward, if it does not it should freeze its wheel. This would result in the robot moving forward if both sensors detect light, or turning towards light if only one does.

Brooks' intent in constraining all intelligence to finite state machines was not only to simplify the engineering of the behaviours, but also to force the intelligence to be **reactive**. A fully reactive agent has several advantages lacking in previous robotic or other AI systems. Because their behaviour is linked directly to their sensing, they are able to respond quickly to new circumstances or changes in the environment. This in turn allows them to be **opportunistic**, another advantage touted by Brooks of his systems over conventional planning systems. A conventional planner might continue to execute a plan oblivious to the fact that the plan's goal (presumably the agent's intention) had either been fulfilled or rendered impossible by other events. An opportunistic agent notices when it has an opportunity to fulfill any of its goals, and exploits that opportunity.

Two traits make the robots built under subsumption architecture highly reactive:

1. Each individual behaviour can exploit opportunities or avoid dangers as they arise. This is a consequence of every behaviour having its own sensing, and running continuously (in parallel) with every other behaviour.

2. No behaviour executes as a result of out of date information. This is because no information is stored, all information is a result of the current environment.

Although useful for the reasons expressed, these traits also create problems for designing agents capable of complex behaviour. First, if there are two behaviours pursuing different goals, then it might be impossible for both to be opportunistic simultaneously. Consequently, any agent sophisticated enough to have potentially conflicting goals (such as "eat" and "escape danger") must also have some form of **behaviour arbitration**. Subsumption architecture provides this through several mechanisms:

- Behaviours are organised into **layers**. Each layer pursues a single goal, e.g. walking. Behaviours within the same goal are assumed not to contradict each other. Higher layers are added to lower layers with the capability to suppress their behaviours if necessary to the higher goal.

- As indicated above, a behaviour may interfere with the behaviour of other elements — it may suppress their outputs (actions) or either block or change their inputs. These actions occur on communications channels between the behaviours (**wires**, originally in the literal sense), not in the behaviours themselves, and consequently must be part of the output of the dominating behaviour. In other words, all such interference is designed as part of the higher behaviour and does not truly affect the workings of the lower behaviour, just how they are expressed. No behaviour has access to the internal states of any other behaviour.

- After some experimentation, the description of behaviours was changed from "a finite state machine" to "a finite state machine augmented by a timer." This timer *could* be set by external behaviours, and resulted in its own behaviour being deactivated until the timer ran out.

The addition of the timer was brought in due to necessity during the development of Herbert, the can-retrieving robot (Connell 1990). When Herbert had found a can and began to pick it up, its arm blocked its camera, making it impossible for the robot to see the can. This would allow the *search* behaviour to dominate the *pick up can* behaviour, and the can could never be successfully retrieved.

Allowing the can-grasping behaviour to suppress all other behaviours via a timer was a violation of reactiveness. The issue being addressed here is memory — Herbert should presumably have been able to briefly remember the sight of the can, or the decision to retrieve the can, but such a memory would violate the second trait of subsumption architecture listed above. This is the second problem of fully reactive systems: they have no memory. Without memory, an agent cannot learn, or even perform advanced control or perception. Control often requires keeping track of intentions in order to exploit the result of decisions not evident in the environment. Perception often requires processing ambiguous information which can only be understood by using other recent sensor information or other stored knowledge of context to set expectations. This shortcoming of fully reactive systems was discussed earlier on page 29.

Brooks initially resisted having any sort of memory or learning in his architecture. The timer augmentation was the minimum possible addition to the reactive system. It registered globally (by setting the timers on each possibly interrupting behaviour) that a decision has been made without making any globally accessible record of that decision. The only evidence of the chosen activity is the single unsupressed behaviour. Brooks' early resistance to the concept of learning was similar to Lorenz's — both researchers initially thought behaviour was too complicated to allow for plasticity and still maintain order. Other users of subsumption architecture immediately saw the need for plasticity, and generally ignored the strict limitations on variable state. Polly had global variables for maintaining its current navigational goal as well as whether or not it was conducting a tour, and where that tour had begun (Horswill 1993*a*). The Reactive Accompanist had neural networks in some of its behaviours, which though theoretically finite state, were effectively variable storage of recent events (Bryson 1992). Another robot built under Brooks' direct supervision, Toto, actually learned a map. Toto had a module that constructed a landmark-based map as individual nodes with memory. When making navigational choices, the map itself determined the best route, it was not read by an external process (Matarić 1990). Brooks (1991*a*) acknowledges this need, and his current projects also contain neural-network type representations (Brooks et al. 1998, Marjanovic et al. 1996).

3.4 The Edmund Architecture and Its Hypotheses

3.4.1 Introduction

The architecture used for the research described in the next chapter is called Edmund, and was created by the author in separate research (Bryson 1996*b*, n.d.). Edmund was designed as an extension of subsumption architecture which includes more explicit representation of sequential and hierarchical control. It is a behaviour-based architecture, like the two presented in the previous section. Like the subsumption architecture, Edmund is a fully specified, explicitly coded architecture. Edmund has been used for experiments using both robots and **simulations**, software environments designed to simulate some portion of the real world. Previous to research in Chapter 4, Edmund has only been tested in his simulated blocks world — an environment too complex for a purely reactive system, but not sufficiently dynamic to fully test reactivity. (Section 4.4 discusses various methods of evaluating performance of architectures).

This section begins by describing the elements of Edmund's design that differ from subsumption architecture (the more fully described of the two architectures already presented) and explaining technical motivations. These differences serve as hypotheses on the nature of intelligence, which will be examined in the rest of the dissertation. The remainder of this chapter consists of a review of several other agent architectures along with their hypotheses, and consider some as further evidence of the thesis. The following chapter, Chapter 4, describes the results of experimental research done to validate Edmund's hypotheses from a behavioural standpoint, while Chapter 5 considers these same hypotheses from the perspective of natural intelligence.

3.4.2 Hypotheses

Edmund is intended to extend behaviour-based approaches to artificial intelligence to allow for behaviour of greater complexity. It differs significantly from purely reactive systems in three ways.

1. Perception requires memory.

Under Edmund, the individual behaviours *are* allowed variable state. The hypothesis underlying this difference is that intelligent perception requires memory. Perception combines information from sensing and previously known information, such as context and expectation. Thus the behaviour needs both storage capability to build context from recent experience, and processing space for combining multiple information streams into a single action.

In Edmund's implementation, the behaviours are programmed as objects in an object-oriented language.[3] Object-oriented languages are an approach to organising program code around the concepts or actors of the program. The core of an object is its state — the description of its features. Around this core are methods: procedures for accessing information about the object, using that information for computation, or changing information. In Edmund, each behaviour is coded as an object, and the behaviour's perceptual state is object state.

[3] The exact language is unimportant: so far Edmund has been implemented in Common Lisp Object System, PERL 5.003, and C++

2. Memory is modular but not strictly encapsulated.

Under Edmund, behaviours are able to observe the state of other behaviours, and to trigger those other behaviours directly. The hypothesis underlying this difference is that modularity is merely an abstraction, though one that occurs also at an operational level. Nevertheless, different behaviours are highly influenced by overall cognitive context, and "cognition" in this context is simply the operation of other behaviours. Although too much interconnectivity would lead to confusion and the loss of the advantages of modularity, limited awareness of relevant states in other behaviours is useful for coherent behaviour. This dissertation therefore separates two concepts linked in the term "modular" by Fodor (1983). **Modularity** is used only to express differential organisation, composition and / or functionality between different sections of an agent's intelligence, but it does not imply a lack of communication of information between these sections. That lack of communication will be referred to as **encapsulation**, a term borrowed from object oriented design methodology in computer science[4].

In Edmund's implementation, the accessibility is provided directly by the behaviours. The theory is that a behaviour may evolve perception either for internal or external sensing: the distinction is fairly arbitrary from the perspective of a single behaviour. In practice, the objects that make up the behaviours have methods that will return copies of their internal state, and also have methods which will prompt them to update their state. These prompts may be called by another behaviour on which the first behaviour's internal state depends, when the second behaviour has updated. There is an important distinction here between the model and the implementation: in the model each behaviour is constantly monitoring the things it perceives. The fact that method calls are the mechanism for executing this model is a concession to speed and the fact that the architecture is only simulating parallel processing.

3. Action selection is dependent on structured control.

Edmund allows for three means of ordering behaviour.

- As sequences. A sequence of actions, called an *action pattern*, once triggered simply activates each element of the sequence in turn, until either the end is reached or there is a radical failure of an element, in which case the sequence terminates.

[4] Unfortunately, Fodor also uses this term, but in a slightly different manner.

- As a prioritised set. The prioritised set is called a *competence*. When a competence is active, it in turn activates the element within it with the highest priority that is also released. In other words, of the elements ready to execute (where readiness may either be perpetual or triggered by the correct perceptual input), the one with the highest activation inhibits all the other elements resulting in only its own behaviour being expressed.

- And as parallel activities. A cluster of parallel activates is, somewhat confusingly, called a *drive*. A drive is roughly equivalent to the layer system in subsumption architecture. Each element is analogous to an individual layer, or to a single drive in conventional drive theory (e.g. Hull 1943). A drive's elements each operate in parallel, with a prioritization computed between them. A higher priority element may monopolise the agent's behaviour, but will only do so in particular perceptual contexts. High priority drive elements may also be made to habituate after firing if they control behaviours that are important to execute periodically.

Notice that, as in Section 2.2, the terminology above although familiar from psychology (in Lorenz 1950, Chomsky 1980 and Hull 1943, respectively), have a substantially modified, technical meaning within the context of the architecture. The top level of an agent's control system will consist of a single drive in the Edmund sense; the *elements* of that drive, which operate in parallel, are more equivalent to the drives in Hull. Both drives and competences may have as their elements either competences or sequences. This is a source of hierarchy in the Edmund architecture. Sequences consist of primitive actions; they are the interface to the actual behaviour of the agent. A sequence may have any number of elements, including just one.

A visual expression of the three structured control components is shown in Figure 3.4. The script in this figure is designed to control a robot — the figure is explained further in that context on page 82. In this figure, the control hierarchy is shown with its root to the left. Action patterns appear as sequences of commands placed in elongated boxes. The root of a competence is labelled **C**, and its elements appear to its right, arrayed along a bar with the highest priority element appearing close to the top. In this script, the element that represents the goal of the competence, which is necessarily its highest priority element, shares the same name as the competence. If the goal executes succesfully, the competence terminates succesfully. If none of the elements can execute, the competence fails. The hierarchy is rooted with a drive called *life*. As can be seen, drives are very much like competences; they are in fact a special

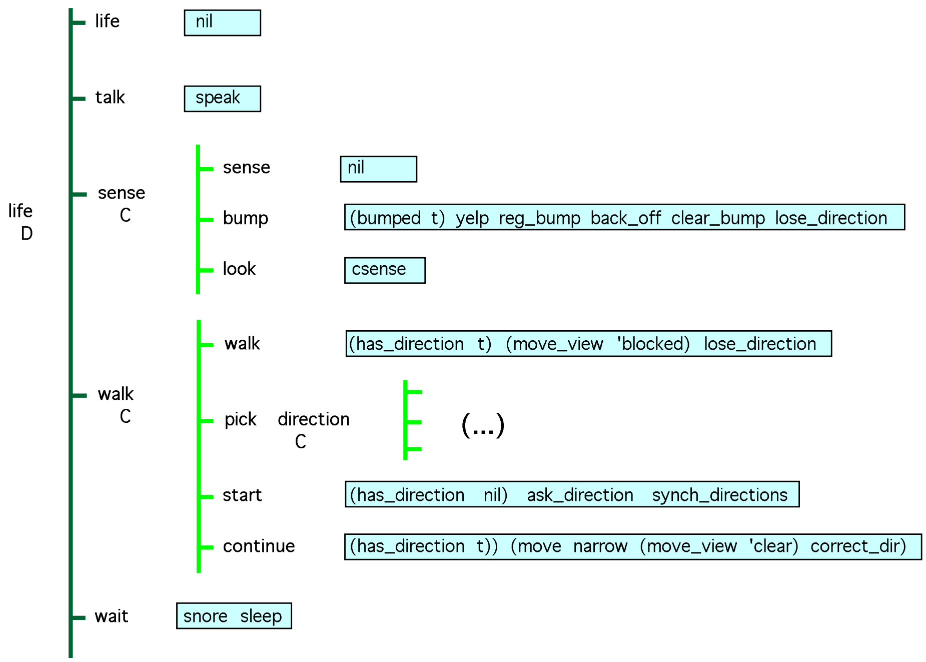


Figure 3.4: An example Edmund control script (see text).

type of competence. *Life* has no goal, so it can never succeed. Also, its lowest priority element has no precondition, so the element can always run, thus *life* will never fail.

3.4.3 Relevance to the Thesis

There are several hypotheses underlying these differences from a fully reactive system. One is that the combinatorial complexity of action selection is too high to be managed without additional structure. Of course, a hypothesis of this nature is impossible to demonstrate through experiments. There is an argument on mathematical grounds (see Section 2.2.6 for an explanation and Dawkins (1976) or McGonigle & Chalmers (1998) for the relevant analysis) and in lack of contradiction. To date, AI systems that do not use such mechanisms cannot master as complex of tasks as those that do (see Section 4.3 and Tyrrell (1993)), which can be seen as positive evidence. The rest of this chapter will provide further evidence along these lines.

Other hypotheses can be experimentally validated. One is that it is possible to both exploit structured control and to be sufficiently reactive to respond to opportunities and threats in the environment. The negation of this hypothesis is explicitly stated by both Hendriks-Jansen (1996) and Maes (1991). To disprove them (and validate Edmund's hypothesis) we need only demonstrate reactive systems we know to have structured control. This is the purpose of the following chapter, Chapter 4. Another is that animals utilise sequential or hierarchical control structures. Evidence for this is discussed in Chapter 5.

As mentioned in the introduction to this chapter (page 33) some researchers reject any sort of ordered or centralised control on the basis that this simply recreates the problem. The element that evaluates the sensory information and chooses actions is solving the entire problem of intelligence itself: it is an intelligent being within the intelligence. One of the main hypotheses of the sort of approach taken by Edmund is that problems such as following sequences of behaviour, the selective activation of a cluster of behaviours, and the selective inhibition of behaviours providing orderly prioritization are a significant reduction of the task of intelligence, and can in turn be performed by relatively simple modular behaviours. Since the definition of behaviour in Edmund has already allowed for the sorts of specialised state that allows keeping track of location within a sequence or activation level within a competence, and for the observation of internal behaviour state sufficient to tell if they are perceptually activated, the controlling behaviour does not need to be any more specialised than any other behaviour, at least in theory.

In practice, the behaviour that allows for the selection of action in Edmund is far more sophisticated than most other behaviours, because it has to be particularly efficient. It is this behaviour which simulates the parallelism of the drives, as well as comparing the relative activation levels and ensuring all active behaviours continue to perceive. The exact mechanisms of this behaviour are described elsewhere (see Bryson & McGonigle 1998) and the code is available on the Internet for inspection (http://www.ed.ac.uk/∼joannab/edmund.html).

3.4.4 Issues Not Covered in the Edmund Hypotheses

Edmund does not model the learning of its own behaviour, for example, the learning of new control sequences or new behaviours. It also does not allow for misorderings of sequences as occurs occasionally in normal animals and humans, and frequently under certain pathological conditions. Although these are both interesting areas of research, the current architecture and therefore the experimental research in the next chapter does not address them. However, both of these issues are discussed in Section 5.3, along with relevant artificial models.

3.5 Alternative Hypotheses

There have been a very large number of AI architectures proposed over the years, particularly as a result of the recent interest in complete systems (agents or robots, see Section 2.2.1). Since Edmund itself is not a contribution of this dissertation, this section will not attempt a full review of the related architecture literature. Rather, the emphasis will remain on examining the psychological relevance of the field. This section begins with a review of related alternatives from behaviour-based AI, some of which will be relevant in the following chapter. It then goes on to cover the most dominant, widely used agent architectures. The subsections are in approximate increasing order of the size of their currently active research communities.

Part of the reason for the emphasis on well established architectures is that these are in turn subject to a significant amount of selective pressure. They tend to evolve over time towards an approach that works better. As such, their success and their evolutionary trends should both be taken as evidence for their hypotheses, although as always in science such information is confounded by the effects of dominating paradigms and personalities.

3.5.1 Behaviour-Based Architectures

The use of the reactive and/or behaviour-based approach is widespread, particularly in academic robotics. However, no one architecture is used by even ten percent of these researchers. Subsumption architecture, described above on page 39, is by far the best known of the architectures. However, relatively few agents have been built that adhere to it strictly. For example, Matarić 1990, Bryson 1992 and Pebody 1995 all include adaptive extensions; Appleby & Steward (1994) make the behaviours nearly completely independent — they would now be called "agents" in themselves. Most roboticists, even within Brooks' own laboratory, seem to have been more inspired to develop their own architecture, or to develop code without a completely specified architecture, than to attend to the details of subsumption (e.g. Horswill 1993*a*, Steels 1994, Marjanovic et al. 1996, Parker 1998).

Of the many behaviour-based architectures inspired by subsumption, the one that has in turn attracted the most attention is Maes' spreading activation network (Maes 1989, 1991). Maes' architecture consists of a number of nodes, including action nodes, perception nodes, and goal nodes. The nodes are each connected to another by a two-way system of links. One link specifies the extent to which the second node requires the first node to have executed, the other specifies the extent to which the first node enables the second node to fire. These conduits are used to allow activation to spread both bottom-up, starting from the perception nodes, and top-down, starting from the goal nodes. When a single node gets sufficient activation (over a threshold) that node is executed.

Maes' greatest explicit hypothetical difference from subsumption architecture is her belief that agents must have multiple, manipulable goals (Maes 1990*a*). She is mistaken in her claim that subsumption architecture only allows the encoding of a single goal per agent, however the strictly stacked goal structure of subsumption is sufficiently rigid that her arguments are still valid. A somewhat more implicit hypothesis is the need for a way to specify sequential behaviours, which her weighting of connections allows. In these two hypotheses, Maes architecture agrees with Edmund: in Edmund, each competence is expressly designed to meet a goal, and competences become active and inactive as the situation warrants. However, Maes is very explicitly opposed to the notion of hierarchical behaviour control (Maes 1991). Maes states that using hierarchical methods for behaviour arbitration creates a bottleneck that necessarily makes such a system incapable of being sufficiently reactive to control agents in a dynamic environment.

This hypothesis was disputed by Tyrrell (1993), who showed several flaws in Maes' approach, most notably that it is insufficiently directed, or in other words, does not focus attention sufficiently. There appears to be no means to set the weights between behaviours in such a way that nodes composing a particular "plan of action" or behaviour sequence are very likely to chain in order. Other related behaviours often fire next, creating a situation known as **dithering**. There is actually a bias against a consummatory or goal behaviour being performed rather than one of its preceding nodes, even if it has been enabled, because the goal, being in a terminating position, is typically connected to fewer sources of activation.

Tyrrell's competing hypothesis is that hierarchy can be exploited in action selection, providing that all behaviours are allowed to be fully active in parallel, and the final decision is made by combining their computation. Edmund's distinction from Tyrrell is that it softens or drops these provisions. Tyrrell (1993) gives evidence for his hypothesis by comparing Maes' architecture directly against several hierarchical ones in a task in a simulated environment. Tyrrell's work is further reviewed, and Edmund similarly tested, in Chapter 4.

Correia & Steiger-Garção (1995) describe another behaviour-based architecture which allows for structures very like action patterns and competences (though interestingly they refer to these elements as "behaviours" and "fixed action patterns", respectively). This architecture has not yet been as influential as Tyrrell's but bears mentioning due to its similarity to Edmund. Correia & Steiger-Garção (1995) have successfully implemented their architecture on robots that could in turn manipulate their environment. The difference between their architecture and Edmund is, similar to Tyrrell's, a lack of an arbitrating behaviour or partial activation. Instead, FAPs compete in pairs until only a single FAP has won. Although there have been no direct comparisons of this approach to Edmund, the strategy seems a baroque (and more difficult to program) means of achieving the same end, with a likelihood of producing less, rather than more, reactivity due to the potential length of the process.

Another recent architecture mentioned with a relatively small but very active research community is that of Blumberg (1996), which takes considerable inspiration from both Maes and Tyrrell. Again, the system is similar to Edmund, but with a more complicated voting mechanism for determining an active behaviour. This time the system is a a cross between Maes' and Tyrrell's: the highest activated node wins and locks any critical resources. Nodes that are also active but do not require locked resources are allowed to express themselves. Thus a dog can both walk and wag its tail at the same time for two different reasons. Finding a way to express multiple behaviours addresses one of Tyrrell's criticisms of Maes, though it still doesn't allow for the hybrid behaviours as expressed in Figure 3.1. The spreading of activation is also more like the free-flow hierarchy advocated by Tyrrell than the simpler Maes networks. Blumberg's architecture is currently being used by his group for animation under the sponsorship of Disney, so it can be expected to be subjected to significant selectivist pressures in the near future.

Finally, Rhodes has extended both Maes (Rhodes 1995) and Blumberg's (Rhodes 1996) systems to allow variable state to be communicated between behaviours. In a limited implementation of the first hypothesis stated for Edmund, Rhodes allows for a reference to a particular focus of attention (a *proneme*, after (Minsky 1985)) to be passed from one behaviour to another. This radically reduces the number of behaviours that need to be implemented. For example, there can be a behaviour "grasp the thing I'm looking at" instead of three different behaviours "grasp the hammer", "grasp the camera", "grasp the cup", etc. This hypothesis is sufficiently common in AI that it has a name, **deictic representation**, and is used frequently, particularly in vision systems (Ullman 1984, Agre & Chapman 1988, Whitehead 1992, Brand 1995, Levison 1996, Horswill 1995, Grand et al. 1997). It also has considerable psychological evidence, see (see Ballard et al. 1997, for a review). Rhodes (1996) creates an autonomous agent that plays the character of the big bad wolf from fairy tales in an animated environment.

3.5.2 Multi-Layered Architectures

The achievements of behaviour-based and reactive AI researchers have been very influential outside of their own communities. In fact, there is an almost universal acceptance that at least some amount of intelligence is best modelled in these terms, though relatively few AI professionals would agree with Brooks' quote above (page 37) that all cognition can be described this way. The Edinburgh robotics group was one of the first to establish a hybrid strategy, where a behaviour-based system is designed to work with a traditional AI planner, which deduces the next action by searching a knowledge base for an act that will bring it closer to a goal. Traditionally, planners have micro-managed, scripting every individual motion. By making their elements semi-autonomous behaviours which will react or adapt to limited uncertainty, the planners themselves can be simplified. The following is a recent account of a project from the late 1980s:

"The behaviour-based plan execution was implemented bottom up to have as much useful capability as possible, where a useful capability is one which looked like it would simplify the design of the planner. Similarly, the planner was designed top down towards this interface, clarifying the nature of useful capabilities at which the behaviour-based system should aim. This design method greatly reduced the complexity of the planner, increasing the complexity of the agent much less than this reduction, and thus reduced the overall system complexity. It also produced a robust system, capable of executing novel plans reliably despite... uncertainty." (Malcolm 1997, Section 3.1)

Malcolm's system can be seen as a two-layer system: a behaviour-based foundation controlled by a planning system. More popular of late have been three layer systems. These systems are similar, except that there is a middle layer that is essentially like the control execution of Edmund — sequences of behaviours or small plans with conditional branches. The planner manipulates these rather than the behaviours directly. Probably the dominant mobile robot architecture right now is 3T (Bonasso et al. 1997), which has been used on numerous robots, from academic mobile robots, to robotic arms used for manipulating hazardous substances, previously controlled by teleoperation, to maintenance robots for NASA's planned space station. Leon et al. (1997) uses 3T in simulation to run an entire space station, including farming and environmental maintenance. See Hexmoor et al. (1997) for a recent review of many two and three layer architectures.

From a hypothesis standpoint, 3T is similar to Edmund, excepting that Edmund makes no attempt at modelling deliberate planning. The other distinctions are relatively subtle, at least for the context of this dissertation. 3T does not model non-modular state sharing at the behaviour level; its middle layer, RAPS (Firby 1987, 1996), does not have the same flexibility for constructing infinite chains or loops in its command structure. Bryson & McGonigle See 1998, for more information on this feature of Edmund.

3T may seem a more appealing model of human intelligence than the behaviour-based models discussed before, in that it has something approximating logical competence. However, traditional planners are exactly the sort of system Hendriks-Jansen (1996) based his criticism of hierarchical control on in the quotation in Chapter 1. Planning has been mathematically proven an unrealistic model of intelligence (Chapman 1987, Horswill 1997) because it relies on search. Search is combinatorially explosive: more behaviours or a more complex task leads to an exponentially more difficult search. Though there is no doubt that animals do search in certain contexts (e.g. seeking food, or for a human, deliberate choice of a gift), the search space must be tightly confined for the strategy to be successful. A nearer model of this process is ATLANTIS (Gat 1991), which has the top, planning layer activated only by the middle layer on demand. If one accepts that the top layer in a three layer system is the deliberate layer, and the middle layer is for implicit or automatic behaviour, than Gat's hypothetical difference from the top down architectures such as 3T is simply that deliberation is triggered when automation fails. The alternative model, which is more typical (Bonasso et al. 1997, Albus 1997, Hexmoor et al. 1997, Malcolm 1997) has the top level being in a continuous state of high-level control, even if it does not attend to the details managed by the lower layers. Gat's hypothesis agrees better with the psychological work of Norman & Shallice (1986), and is also a more natural extension of the behaviour-based approach.

Malcolm (1997) brings out another problem of accepting hybrid models as psychologically plausible. For a hybrid system, emergent behaviour is useless. This is because an emergent behaviour definitionally has no name or "handle" within the system, consequently the planning layer cannot use it. In at least human systems, acquired skills can be recognised and deliberately redeployed (Karmiloff-Smith 1992). This strategy is not well modelled in any of the systems presented in this chapter; it begs a more dynamic or distributed substrate for the behaviour modules than simple program code (see Section 5.3.4). Hexmoor (1995) attempts to model both the development of a skill (an element of the middle layer) from actions performed deliberately (planned by the top layer) and the acquisition of deliberate control of skills. His hypothesis of requiring both these forms of learning are probably valid, but his actual representations and mechanisms are still relatively unproven.

In summary, the success of two and three layer architectures can be taken to some extent as evidence of the central hypothesis of this thesis, that hierarchical control is a reasonable intelligent strategy. Although I have criticised the large scale utility and plausibility of planners, many of the tasks built in 3T itself actually do not utilise the planning layer (Bonasso et al. 1997), rendering it then also effectively a two layer architecture very much like Edmund. Provided with a properly constrained space and an auxiliary, rather than dominating, role, planning may effectively be another useful behaviour, like the control behaviour in Edmund.

3.5.3 Beliefs, Desires and Intentions

Although robotics has been dominated by three-layer architectures of late, the field of autonomous agents is dominated, if by any architecture, by the Procedural Reasoning System, or PRS (Singh 1998). PRS also began as a robot architecture, but has proven so reliable it has also been used in manned space flight and air traffic control (Georgeff & Lansky 1987). It was developed at roughly the same time as subsumption architecture, as a follow-up program to the longest running robot experiment ever, Shakey (Nilsson 1984). PRS aims to fix problems with the traditional planning architectures exposed by the Shakey project. Such problems include:

- Forming a complete plan before beginning action. This is a necessary part of the search process underlying planning — a planner cannot determine whether a plan is viable before it is complete. Many plans are in fact formed backwards: first selecting the last action needed to reach the goal, than the second last and so on. However, besides the issues of opportunism already discussed, many details of a real problem cannot be known until the plan is executed. For example, when crossing a room full of people, the locations of the people are determined at the time of crossing, and cannot be predetermined.

- Taking too long to create a plan, ignoring the demands of the moment. The standard example is trying to cross a road — a robot will not have time to replan if it suddenly spots a car, it will need to know reactively to move out of the way.
- Creating plans that contain elements other than primitive acts — taking advantages of skills or learned procedures.
- Being able to manipulate plans and goals, including allowing abandonment of a plan or the pursuit of multiple goals.

Obviously, this list is very similar to the problems the behaviour-based programmers attempted to solve. The two main differences for PRS are first, that it maintains as a priority the ability to construct plans of action. In fact, it allows for many specialised planners or problem solvers. The second difference is that PRS is couched very much in psychological terms, the opposite of Brooks' disparagement for conscious impact on intelligent processes. PRS is referred to as a BDI architecture, because it is built around the concepts of beliefs, desires and intentions.

The PRS architecture consists of four main components connected by an interpreter (sometimes called the "reasoner") which drives the processes of sensing, acting, and rationality. The four components are:

1. The database of *beliefs*. This is knowledge of the outside world from sensors, of the agent's own internal states, and possibly knowledge introduced by outside operators. It also includes memories built from previous knowledge.
2. A set of *desires*, or goals. These take the form of behaviours the system might execute, rather than descriptions of external world state as are often found in traditional planners.
3. A set of *plans*, also known as knowledge areas. These are not necessarily completely specified, but are more likely to be lists of subgoals useful towards achieving a particular end, like Edmund's competences. As mentioned earlier, these may include means by which to manipulate the database (beliefs) to construct a complete next action or new knowledge.

4. A stack of *intentions*. Intentions are just plans currently operating. A stack indicates that only one plan is actually driving the command system at a time, but multiple plans may be on the stack. Typically, ordering is only changed if one plan is interrupted, or if new information triggers a reevaluation of the plans.

Like multi-layer architectures, PRS works from the hypothesis that a system needs both to be able to plan in some situations, such as navigation, but must also be able to execute skilled acts for situations where search is not reasonable, such as avoiding trucks. In some sense, each plan is like a behaviour in behaviour-based AI. Behaviour-based AI is essentially a retreat to allowing programmers to solve the hard and important problems an agent is going to face in advance. A procedure to solve an individual problem is usually relatively easy to design. The interpreter, goal list and intention stack then are an action selection device. In comparison to Edmund, this system is probably sufficiently powerful to encode the structures of Edmund's controller. To simulate the top level drive structure one may need to use the parallel version of PRS, which has several complete structures as described above working independently simultaneously, each focusing on different goals, and sending each other's databases signals to allow for coordination. It may also be possible to simulate Edmund's behaviours using the combination of the plans and beliefs, but this would be more difficult to coordinate properly. Essentially, PRS does not see specialised state and representations dedicated to particular processes as worth the tradeoff from having access to general information. It has moved the procedural element of traditional planners closer to a behaviour-based ideal, but only allows for specialised or modularised data by tagging. Whether the benefit in terms of ability to generalise and share data would outweigh the reduction in clarity of object-oriented, data-organised modules can only be determined by direct experimentation. This experimental work has not yet been attempted.

Many researchers appreciate the belief, desire and intention approach in concept, even without agreeing with anything like the PRS architecture. (Sloman & Logan 1998) considers the notions of belief, desire and intention and feelings are central to an architecture, but proposes a three layer architecture, where the top layer is reflective, the middle is deliberative, and the bottom layer reactive. In other words, this is similar to Malcolm (1997) or the first and third layer of Bonasso et al. (1997), but with an additional layer dedicated to manipulating the goals of the top layer, and considering its own current effectiveness. Particularly with the advent of web-based and virtual reality characters, there has been a good deal of research into architectures for using emotional state to help select both goals and affective and social behaviours. This research is ancillary to the current thesis: the psychological issues are beyond its scope and the technical issues are not conflicting. But for examples see Bates et al. (1992), Grand et al. (1997) and Reilly (1996). Bates et al. (1992) in particular shares the concept of plan libraries with PRS.

Another interesting related set of research is the evolution of the Shakey project prior to the development of PRS (Nilsson 1984). Although Shakey had a traditional planner (called STRIPS), over the term of the project the concept of *triangle tables* was developed. A triangle table is a version of a plan which decomposes it into its steps and assumptions, and allows the plan to be restarted from any point when perception indicates that certain elements of the plan are complete. This allows action selection to be reactive within the confines of the plan, rather than relying on memory of what steps should have been executed. Triangle tables are also very much like competences, and are the foundation of teleo-reactive plans (Nilsson 1994), another recently developed form of storage for skilled behaviours developed by plans. Benson (1996) describes using this as the basis of a system that learns to fly airplanes in flight simulators.

The Shakey project also moved from having multiple world models in its first implementation to having a single storage place for predicates of observed data. Any predicate used to form a new plan was rechecked by observation. This development under the selective pressure of experimentation lends credence to the idea that too much modelling of the world is likely to cause difficulties.

3.5.4 Soar and ACT-R

Soar and ACT-R are the AI architectures currently used by the largest number of researchers, not only in AI, but also in psychology and particularly cognitive science. These architectures are fundamentally different from the previously reviewed architectures. Both are also older, dating to the late 1970s and early 1980s for their original versions, but both are still in active development (Laird & Rosenbloom 1996, Anderson & Matessa 1998). The Soar community in particular has responded to the behaviour-based revolution, both by participating directly in competitions with the approach (Kitano et al. 1997) and even in portraying their architecture in three layers (Laird & Rosenbloom 1996).

Soar and ACT-R both characterise all knowledge as coming in two types: data or procedures. Both characterise data in traditional computer science ways as labelled fields, and procedures in the form of a production rules. A production rule consists of two parts: a condition and an action. If the condition holds, the action is executed. For example, IF (the dog is on the table) THEN (shout “bad dog”).

Soar is a system that learns to solve problems. The normal procedure is to match its production rules against the current state of the world, find one that is applicable, and apply it. This is automatic, roughly equivalent to the middle or bottom layer of a three layer architecture. If more than one production might work, or no production, or nothing changed when the previous production was tried, then Soar considers itself to be at an *impasse*. When Soar encounters an impasse, it enters a new problem space of trying to solve the impasse rather than the current goal. The new problem space may use any means available to it to solve the problem, including planning-like searches. Soar has several built-in general purpose problem solving approaches, and uses the most powerful approach possible given the current amount of information. This process is thus something like the top level of ATLANTIS, except that Soar allows the process to recurse, so the meta-reasoner can itself hit an impasse and another new reasoning process is begun.

Soar includes built-in learning, but only of one type of information. When an impasse is resolved, the original situation is taken as a precondition and the solution as a procedure, and a new rule is created that takes priority over any other possible solution if the situation is met again. This is something like creating automatic skills out of declarative procedures, except that it happens quickly, on only one exemplar. This learning system can be cumbersome, as it can add new rules at a very high rate, and the speed of the system is inversely related to the number of rules. To partially address this problem, Soar has the concept of a *problem space*, a discrete set of productions involved in solving a particular goal or working in a particular context. Problem spaces are roughly equivalent to competences in Edmund.

ACT-R is essentially simpler than Soar: it does not have the impasse mechanism nor does it learn new skills in the same way. Nevertheless, ACT-R is still used for cognitive modelling, and can be used to replicate many psychological studies in decision making and categorisation. ACT-R also faces the difficulty of combinatorics, but it takes a significantly different approach: it attempts to mimic human memory by modelling the probability that a particular rule or data is recalled. Besides the two sets of "symbolic" knowledge it shares with Soar, ACT-R keeps Bayesian statistical records of the contexts in which information is found, its frequency, recency and utility. It uses this information to weight which productions are likely to fire. It also has a noise factor included in this statistical, "subsymbolic" system, which can result in less-likely alternatives being chosen occasionally, giving a better replication of the unpredictability of human behaviour. Using alternatives is useful for exploring and learning new strategies, though it will often result in suboptimal performance as most experiments prove to be less useful than the dominant strategy.

Several hypotheses differentiate both these architectures from the behaviour-based approaches. First, there is the assumption that fundamentals of intelligence can be universally recorded in fairly homogeneous data types. As mentioned earlier, this is no doubt true at least at the neural level, but it is more questionable whether this is a useful level of abstraction for developing intelligent systems. Of the two architectures, Soar is the only one that has been adapted to the problems of interfacing with the real world and representing noisy and contradictory data, and reasoning about events over time. As such, and with limits set to its potentially enormous learning and impasse processes, it serves reasonably well as a top level planner for a three layer architecture. It may be that Edmund could actually be implemented in the most recent version of Soar, possibly to some advantage since it would thereby gain a system capable of planning. However, this statement demonstrates that Soar is not a theory on the level of Edmund or the other behaviour-based architectures — it is much more general, which necessarily means it provides less information intrinsically about the nature of intelligent processes.

3.6 Conclusion

This chapter has examined a number of hypotheses on the optimal strategy for the control of intelligent behaviour as they have been encoded in AI agent architectures. This examination has shown several of the current trends in histories of agent architectures:

- the need for pre-coded plans or plan elements in order to solve problems reliably and quickly,
- the need for the provision of reactive and opportunistic behaviour, and

- the need for limiting the attention of action selection mechanisms to a subset of possible behaviours appropriate to the context.

The first and third of these trends are relevant to the thesis of this dissertation, in that they describe sequential and hierarchical structure, respectively. The second is indicative of why older models that made no such allowances have been open to such harsh criticism from dynamic theories.

This chapter has also described in detail a particular architecture, Edmund, which meets all three of these criteria. The next chapter will examine the functioning of Edmund in dynamic environments.

Chapter 4

Experiments and Results

4.1 Introduction

This chapter presents experimental research demonstrating the viability of Edmund's approach to action selection. Chapter 1, page 5 describes the position this dissertation challenges, that hierarchical and sequentially ordered behaviours cannot produce adequately responsive systems to model animal behaviour. Chapter 2 introduced the relevancy of artificial intelligence research to creating and testing psychological hypotheses, and Chapter 3 has both described the specific set of hypotheses to be examined here, and presented preliminary evidence in terms of a review of similar systems. Chapter 5 will examine similar evidence, this time from the natural intelligence literature, and suggest some future directions for research.

This chapter is broken into three sections. The first describes the validation of the hypothesis architecture through control of a mobile robot. The second describes comparison of this architecture with several other leading architectures, furthering the hypothesis claim. This work is conducted in simulation. The final section is a brief analysis of the two different approaches, robotics and simulation, for conducting research with the intention of psychological modelling.

4.2 Robot Experiments

4.2.1 Motivation

As mentioned in the previous chapter, the hypothesis architecture, Edmund, had already been constructed and demonstrated before the beginning of this research. However, it had never been applied to the problems of a real robot. Edmund's original domain was a simulation of blocks world. There Edmund's perceptual information was assumed to result from a visual routines processor which could both execute perceptual commands (e.g. "find something green" or "scan up") and return data (e.g. "what colour is the region I'm attending to"). It also assumed a robot arm capable of grasping and moving different coloured blocks. The simulation demonstrated that Edmund's command system provided sufficient structure to solve problems previously unsolved by behaviour-based systems, such as Steve Whitehead's block stacking problem (Whitehead 1992) and the copy demo (Horn & Winston 1976, Chapman 1989). These problems can be proven impossible to solve efficiently in a fully reactive system, because the environment does not provide sufficient information to determine the current stage of processing. These experiments had demonstrated that, although Edmund's control system was not a full-fledged programming language, it was sufficiently powerful to express these kinds of tasks.

The challenge addressed in this research was to prove that in having such a powerful central controller, Edmund had not thrown out the baby with the bath water — that the system could still be flexible and reactive while persistently pursuing complex goals when appropriate. As explained in Chapters 2 and 3, the most accepted platform for demonstrating reactive architectures is autonomous mobile robots. This is because robots necessarily have several special properties common to animals, but uncommon to most computer applications:

- they exist in complex, partially unpredictable dynamic environments,
- they have inaccurate and incomplete information about their world from their sensors,
- they will generally have multiple goals to pursue simultaneously, such as pursuing a course, avoiding obstacles, monitoring their battery levels, and attending to new commands,
- they must operate in real time,
- their actions are fully committed — if they get trapped or fall down a hole, they cannot reverse their situation simply by reversing computation,
- they have specialised tasks and environments they operate in, which they can exploit to simplify their computation.

The goal of the robot research then was to demonstrate that Edmund was reactive to the standard of the robots built under other behaviour-based architectures: that it could respond to its dynamic and poorly perceived environment, and juggle multiple goals, behaving opportunistically. It was also expected that, should this experiment succeed, the combination of having both a powerful and a reactive architecture would led to the development of a significant extension of mobile robot capabilities. Although the first goal of this research was met, the expected extension goal proved excessively ill-posed, and lead to the research covered in the other two sections of this chapter.

4.2.2 Experimental Situation

Previous laboratory work

The robotics research was conducted in the University of Edinburgh Laboratory for Cognitive Neuroscience and Intelligent Systems, under the direction of Dr. Brendan McGonigle. McGonigle, an expert in ordering in primates (including humans) and other animals, has been involved in behaviour-based robotics since the late 1980s. McGonigle (1991) proposed the need for sequential behaviour structures and tasks grammars, and criticised the limits of simple stack hierarchies as goal structures. Considerable robotics research has been conducted in this laboratory, both on a proprietary robot (Donnett & McGonigle 1991) and on an IS Robotics R2 model (Nehmzow et al. 1993, Nehmzow & McGonigle 1993, 1994, McGonigle & Johnston 1995). Unfortunately, both of these robots allowed very little processing overhead, so the architecture was not fully specified as a stand-alone code entity, but was rather fully integrated with the rest of the robots' program code. Much of this work focussed on the integration of novel perceptual strategies into the problems of situated intelligence and demonstrating the utility of task grammars and phased activity (see McGonigle in press, for a review). Donnett & McGonigle (1991) demonstrates using an acoustic beacon for navigation, including learning approximate locations of navigation hazards and making appropriate speed optimisations to negotiate the expected environment. Nehmzow & McGonigle (1994) also demonstrates a learning system, this one using a simple neural network controller rather than a task grammar, where drives for rewards are associated with behaviours in order to control the robot. Nehmzow & McGonigle (1993) demonstrates the use of a light compass to navigate in a way similar to some insects.

The best demonstration of the effectiveness of the task-grammar architecture is McGonigle & Johnston (1995), wherein the R2 robot utilises a camera, its single gripper, infrared sensors and dead reckoning to locate a cluster of upright poles, move them to an operating area, and arrange them for sorting. Dead reckoning from the calibration of the robot's wheels is used to locate the nest of tubes, then infra-red range sensing is used to approach a single tube. The robot then executes a fixed action pattern to bring its grippers into the vicinity of the pole, and vision to gripper sensing to do the final manoeuvre to lift the pole. Using vision in this specialised way removes the problems of segmentation and object identification common to full-scale machine vision problems. Many animals, such as frogs, spiders and horseshoe crabs have been shown to have similar specialised visual systems (Arbib & House 1987, Hendriks-Jansen 1996). The return of the poles home uses a similar approach of dead-reckoning and fixed-action final alignments. This demonstration displays the kind of specialisation of behaviour to task typical of animals (for example, the phased activity of the digger wasp in creating nests (Lorenz 1973)).

Such specialisation is impossible in a fully reactive systems such as subsumption architecture, because the identical perceptual input (e.g. the presence of a pole) would elicit the same high-level behaviour in every uninterrupted circumstance. Subsumption architecture does allow the pole to be treated both as an obstacle and as an object of attention, as the control shifts between levels. It might even allow for the different treatment of the poles when they are in the sorting area vs. in the collecting zone, although this would require an unnecessarily complicating step of deciding which behaviour was higher priority and should suppress the other. However, there is no clean way to express the phase transitions between the collecting and sorting behaviours. If the given task is to collect and sort all available poles, the reactive robot would have no means of telling in its nest whether it had collected all the poles. It would have to be prompted to sort on every entry to the nest, and it would then always go to search for more poles. Even if all the poles were collected, and it had some mechanism of time-out for searching for additional poles, when it returned to its nest it would not know that its task was complete. Even if all the poles were sorted correctly, it would only return to search for poles again, unless it had some in built sense of sufficiency, and felt placated by the number of poles in its nest. The relevance of these issues with respect to natural intelligence is discussed in Section 5.2.1.

Figure 4.1: The Nomad 200 robot, of the Edinburgh Laboratory for Cognitive Neuroscience and Intelligent Systems. In this picture it is encountering a desk. The desk's surface is too low for the sonar sensors (the circles at the top of the robot's tower) to detect when proximate, and the legs too dark for the infra-red sensors (the lights at the base of the same tower) to detect. Consequently, the robot must learn about the desk with its bump sensors (the two rings around the robot's base).

The Robot

The current robotics research platform in the laboratory is also a commercially available robotic research platform. The **Nomad 200** (figure 4.1) produced by Nomad Technologies, Inc., is roughly cylindrical, a little over 1 meter tall and a little under half a meter in diameter at its base. It consists grossly of two sections, a tower and a base. The tower contains an essentially conventional 486 personal computer, running the Linux operating system, as well as the sensory equipment for the robot. It has 16 identical faces, each equipped with a sonar sensor at its top and an infrared sensor near the base. The base contains motors and 3 wheels. It has three degrees of freedom with respect to motion: it can translate the entire robot either forward or backward at a variety of speeds and accelerations, rotate the direction of the wheels around the central axis, and rotate the turret with respect to the base. It has no means of rotating the base. The base is surrounded by two rings of **bump sensors**, each of which is broken into 10 regions for localising impact. The breaks between regions are offset between the two rings.

Sonar sensors are range sensors, meaning they can be used to determine distance to other objects in the environment. They function by emitting then detecting sound and timing the return of its reflection. On the Nomad 200, the sonars return values between 255 and 7 representing the number of inches to the nearest object in that direction. However, this information may be seriously distorted by a number of factors. The nearest object may be considerably closer than the readings indicate. If the sonar signal hits a plane at an oblique angle relative to the robot, the signal will reflect away from rather than towards the robot on its first bounce, giving an overestimate of free space when the signal eventually returns to the detector. In this case, the sonar sensor value consists of the summed distance between several objects rather than just the nearest one and robot. The nearest object may also be much further away than reported if the sonar receiver happens to pick up the stray reflection of one of the other face's emitters before receiving its own signal. Finally, the area covered by each sonar emission is not an arc around the source point, but is rather a complicated series of lobes that leaves some areas only slightly off the centre of the receiver almost completely uncovered.

Infrared sensors are also range sensors, but have a much shorter range. On the Nomad they provide coverage for the area too close for the sonar sensors to accurately reflect. They return values from 15 to 0, roughly equivalent to inches. The failings of the infrared are relatively simple compared to those of the sonar: they return higher values for light, reflective surfaces than for dark or matte ones. The ones provided by Nomad also return very different values depending on the amount of ambient infrared: they have to be recalibrated depending on the quality of sunlight in the laboratory at a given time. This last problem could be resolved through circuit design, but such engineering was beyond the scope of this research.

There were several reasons for the decision to move to the Nomad platform. The old robots had become less reliable electronically. The Nomad is of more human size and therefore its niche environment is more similar to human scale, making the laboratory itself into an interesting maze-like domain (see Figure 4.2). The Nomad is equipped with voice synthesis and radio ethernet, allowing for more easy real-time debugging as the machine can report on its status. But most importantly, it contains a full fledged personal computer with 24 Megabytes of RAM (in contrast to the R2's 1M), 80 Megabytes of hard disk, and an intel 486 processor capable of running standard operating systems and programs. As such, it is a sufficiently powerful computer to enable a full architecture to be modelled on it in a conventional high-level language such as C++, rather than having to resort to hardware or machine coding.

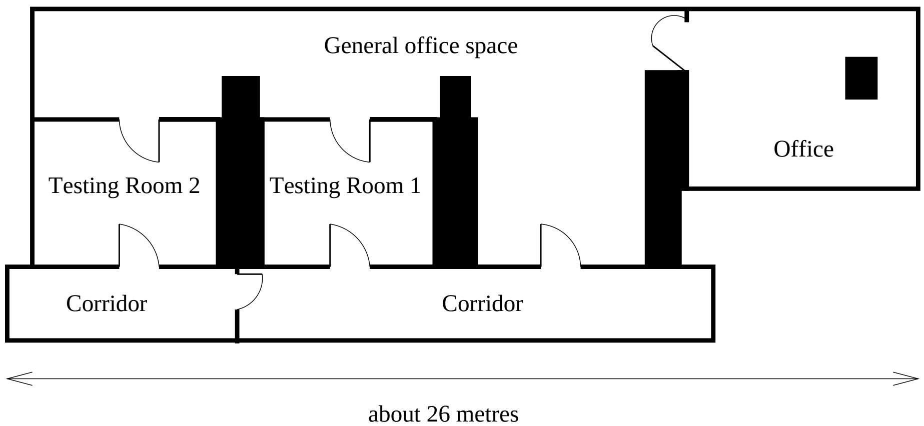


Figure 4.2: Robot experiments described in this chapter were performed in several rooms and a corridor of Laboratory for Cognitive Neuroscience and Intelligent Systems.

Concurrent Laboratory Work

The period during which the research described in this chapter was conducted saw significant research done on three different architectures. Besides Edmund, there were two other robot-related research programs. The first was a significant refinement and extension of the McGonigle task-grammar architecture to support cognizant error recovery, flexible behaviour prioritization as well as lifelong learning (see Figure 4.3). This work was proposed by McGonigle & Johnston (1995) and actually performed by Warnett (forthcoming) as his Ph.D. thesis research. Warnett's work represents the first fully specified and operational version of the task-grammar architecture.

The second project was work to extend and apply the architecture of Humphrys (1997) to robotics. This work, begun by Humphrys and McGonigle, has not yet been completed, but has had substantial impact on the laboratory. It involves allowing a robot to learn its own goal prioritization for manipulating a pre-coded set of behaviours and goals.

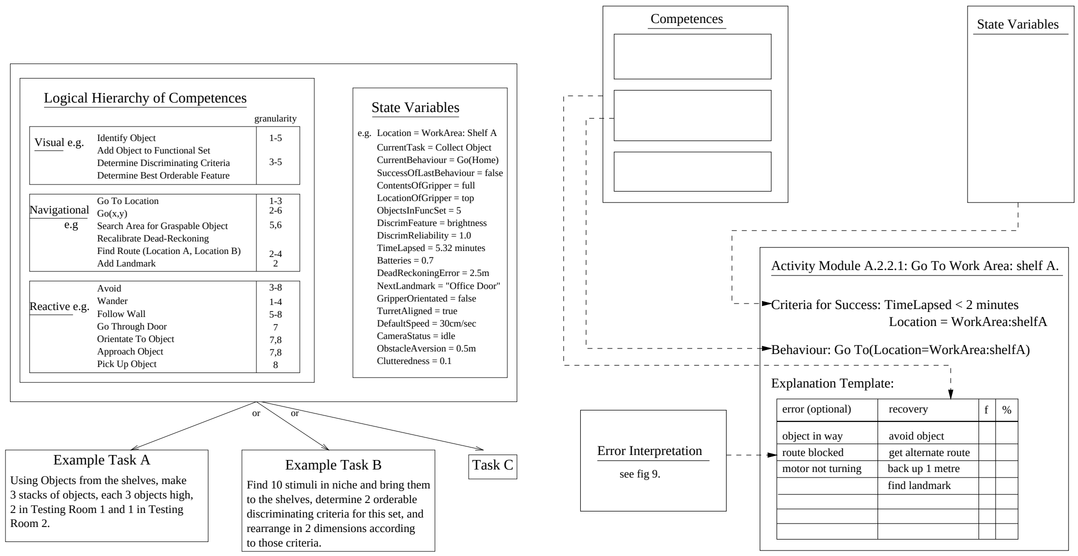


Figure 4.3: Elements of the task grammar architecture with cognizant error recovery developed by McGonigle, St. Johnston and Warnett.

4.2.3 Description of Behaviours

The Nomad robot comes complete with a set of primitive actions and senses, reflecting the output of the sensors and commands to the motors described in the previous section. Unfortunately, these are at a lower level of abstraction than is actually useful for the control architecture. Ideally, the control architecture should not have to make allowances for the kinds of problems and restrictions described above for the sensing, nor should it provide exact speeds in feet per second. A more natural level of command is to "slow down" or "speed up" as the situation warrants. Better yet is to simply choose a goal direction or location and move towards it at a safe speed. This level of abstraction is provided by the behaviours in a behaviour-based architecture.

Table 4.2.3 shows the behaviour modules implemented for the Nomad. As explained in Section 3.4.2, Edmund's behaviours are developed in a behaviour based language. Each different type of behaviour is implemented as a *class*, an object that defines the properties for a set of objects. The class name appears in the left column. In some cases there is a single behaviour corresponding directly to class, but for some classes there are multiple behaviours. For example, the class Direction has 16 behaviours which control motion in directions corresponding to each of the 16 faces of the robot's tower. Some early experiments were also run using 32 be-

Direction	For moving and detecting motion-relevant information. Sixteen behaviours corresponding to faces of the robot.
PMem	Perceptual Memory, a short list of recent combined sensor readings.
Bump	Provide information about local “invisible” obstacles. Multiple behaviours for avoiding specific collision points.
Wave	Detect people waving at sonar (a source of communication).
DpLand	Decision point landmarks, the basis for navigating (vs. wandering.) Multiple behaviours each recognise a choice point and remember previous decisions.
DpMap	Collections of landmarks organised topologically. Multiple behaviours each track an arbitrarily bounded region.

Figure 4.4: The behaviour classes written for the Nomad Robot.

haviours, the extra 16 corresponding to the directions directly between faces, but these proved unnecessary for smooth movement and were consequently discarded to simplify the agent.

This multiplicity of nearly identical behaviours corresponding to redundant hardware elements is similar to the strategy used by Brooks (1989) to control the six legs of an insect-like robot using the subsumption architecture. One difference is that Directions do not control the robot’s motors concurrently, although the behaviours are active perceptually in parallel. Another difference is that some of these behaviours’ attributes and decision processes are shared between them. To this extent, there is almost a 17th behaviour, and indeed this is how the shared capabilities are portrayed in Figure 4.5. Each individual Direction is able to move forward in the direction of its face, knowing how to detect obstacles that would interfere with that motion, but ignoring irrelevant obstacles to the side or rear. The Directions as a class sense the wheel orientation of the robot, though they do not share their perception of the sonar and IR sensors. The orientation information helps the Directions determine which individual Direction’s actions should be expressed, while the rest of the Directions are inhibited.

The 16 Directions embody the robot’s perception related to range data and obstacle avoidance. Consequently they each have the responsibility for evaluating their own current context with respect to obstacles, and determining the speed and exact direction for motion which falls into their 16th of the space surrounding the robot. The precise algorithms for steering and determining speed are described below in Section 4.2.4.

Figure 4.5 (below) shows the state and interactions of these behaviours for the benchmark mobile robot task of obstacle avoidance. This demonstrates the utility of Edmund’s first and second hypotheses (Sections 3.4.2 and 3.4.2.) The Edmund Nomad behaviours exhibit a continuum of learning and memory, from transient to long-term. Some forms of perceptual learning are very short term, such as the $< \frac{1}{6}^{th}$ of a second persistence of previous sonar readings used for "sanitising" the current readings (see Section 4.2.5 below). Obstacles detected by bump sensors are remembered long enough for the robot to maneuver around them, approximately one or two minutes. The decisions made at particular "decision points" become a part of long term memory, for map learning. This way the robot learns paths that are likely to miss obstacles invisible to the two forms of range sensing, but does not necessarily permanently avoid their location. This can be useful if an obstacle moves and a new path needs to be determined.

The Edmund control architecture, introduced in Section 3.4.2, interfaces with these behaviours via a limited number of activation points. These points are both the primitives of the control structures and member functions of the objects that compose the behaviours. The specification of the connection between these two very different representations is a code file which pairs the function calls to the behaviour library with the English names used in the control scripts. Figures 4.5, 4.6 and 4.7, illustrate these elements of the Edmund system for the Nomad. Control scripts for the Nomad are also described further below in Section 4.2.5.

4.2.4 A Sample Interaction Between Behaviour and Control

This section describes in detail a set of control primitives and how they relate to a single Direction behaviour, as an example of the functioning of the architecture. The primitives form an action pattern, *advance*, and are listed in order in Figure 4.6.

The most complicated primitive of the set is *move*, which works as follows. First, the Direction behaviour that corresponds to the direction the robot is currently moving is allowed to control the robot's motion. This behaviour computes a new velocity for translation (moving forward), based on the room it has to maneuver. It also computes a velocity for rotation, based on the difference between the robot's current steering angle θ_r, and the Direction's own preferred orientation, θ_d, which is directly perpendicular to the Direction's face. The formulae for both computations are based on power curves, making the robot's trajectories fluid and natural[1].

[1] The steering rate is $k_2 e^{-k_1 \frac{1}{\sqrt{\theta_c}}}$ where the k_1 (26) adjusts the slope of the exponential, k_2 (450) adjusts the scale of the result, and θ_c is the difference between θ_r and θ_d, adjusted for the nearest direction of rotation. The translation velocity is $k_2 e^{-k_1 \frac{1}{\sqrt{dist}}}$ where k_1 (5) is as above, k_2 (100) is the maximum velocity in tenths of an inch per second, and *dist* is the nearest obstacle in the direction of motion also in tenths of an inch.

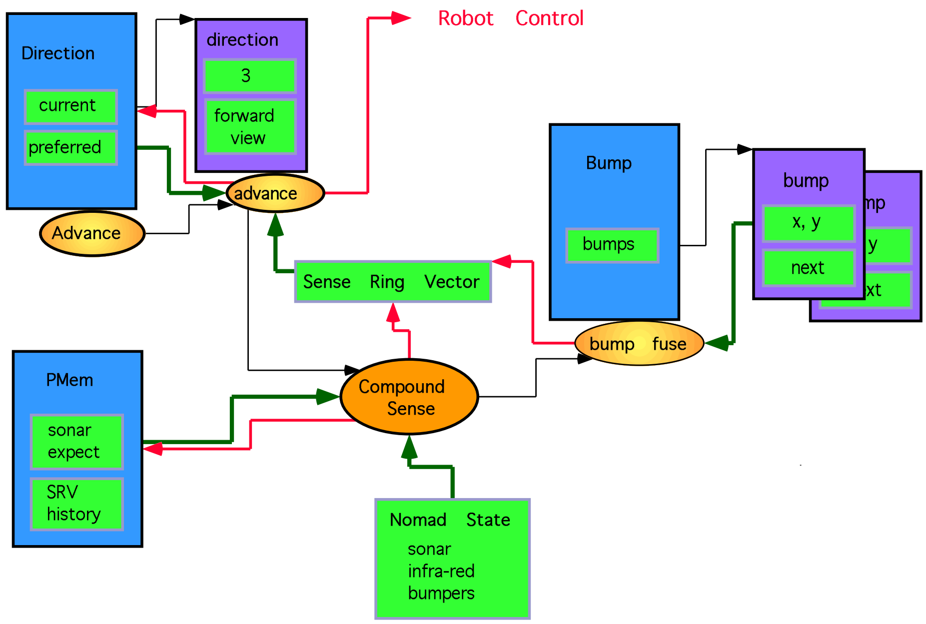


Figure 4.5: The behaviours involved in obstacle avoidance. The key function being highlighted here is a member of the behaviour class Direction called *advance*, but many other behaviours affect the state involved in determining the direction of motion. Thin black arrows indicate a reference link, one behaviour knows the location of another. Thick lines indicate the flow of information in related variable state, and medium-width lines indicate potential action triggers.

Advance directs the Nomad's motors to move at these velocities. In addition, if the active Direction determines there is an obstacle to one side, it will pass its activation to the behaviour corresponding to its neighbouring face opposite the obstruction.

Move operating repeatedly on its own will keep the robot wandering around the laboratory. However, in order to order the experiences of the robot, so they could be remembered as a sort of map, a check was introduced to break the robot's motion into coherent chunks. This is expressed in the behaviour *narrow*, which simply notices if the robot's current active direction is too far removed from the one it originally intended on this leg of the robot's path. If it has, the behaviour "forgets" its intention, which results in motion halting, and a new direction being chosen. This also triggers catastrophic failure which aborts the action pattern.

Normally, the next primitive is *move-view*, which simply directs perceptual attention to the current Direction. Depending on the result of *move* this may be a new version of "forward." Again, the simple directing of this attention may have the side effect of stopping the robot, if the Direction is in fact too close to an obstacle for safe motion.

move	`Direction::current->advance()`
narrow	`if (Direction::current - Direction::  pref > 4)` `{lose-dir && fail}`
move-view	`Direction::current->state()`
correct-dir	`Direction::correct-to-pref()`

Figure 4.6: The interface between the library and the control script. Left are the motion primitives in the action sequence **move** in Figure 4.7. Right is the corresponding code using member functions from Figure 4.5.

Finally, if there are no obstacles in the near vicinity, *correct-dir* directs attention back towards the previously mentioned intended Direction for this leg.

The *advance* action pattern is an excellent example of how an apparently continuous motion can be controlled by an architecture that operates in terms of discrete function calls. Operating on the Nomad robot, *advance* results in smooth, continuous motion. In fact, in real operation, high level attention from the action scheduler is being switched between motion, learning, sensing the external world, and checking and reporting internal functions like battery level. Motion itself is expressed as a competence (see Section 4.2.5 below), and that competence routinely checks the triggers for starting and terminating motion before choosing the case of continuing motion, executing **move**. The behaviour that underlies them however, has been modelled as one of a suite of continuous, parallel behaviours. So long as it is not told to stop expressing itself, it continues to move the robot forward for up to one second after its most recent command. The one second time-out is for safety — if the robot's computer crashes or some other bug occurs, the motors have been instructed to freeze to prevent collision. During normal functioning, the robot receives at least 20 commands a second. In Nomad's commercial simulator, provided to assist with debugging robot code, the calls to *advance* do appear as discrete events, because the commands cannot be performed by separate hardware systems as they are on the real Nomad. The continuity of the robot's motion is a consequence of being embodied as a physical object with wheels and momentum.

4.2.5 Experimental Record

Introduction

Developing evidence on a constructive platform is considerably different from developing it by observation of animals. Research in robotics typically attempts to demonstrate the power of a particular model or approach. As such, it attempts to make an **existence proof**. This term is borrowed from mathematics, where, in some cases, one needs only to present a single counter-example in order to disprove a hypothetical claim. The analogy here is that the robot may be used to demonstrate the reactive capability of an embodied agent controlled by a hierarchical structure, in order to disprove the claim by (Hendriks-Jansen 1996) cited in Section 1.2.2.

Psychology is not mathematics, however, and very few characterisations of intelligent behaviour are anywhere near absolutely defined. Experimental method in psychology, like all science, is rooted in increasing the probability of a hypothesis towards certainty, rather than providing absolute truth. Similarly, a single instance of behaviour taken in isolation cannot be considered strong evidence, as it may be an isolated event resulting from unknown co-occurrences or coincidence, rather than necessarily being generated by the agent's intelligence. However, some allowances are made if a behaviour is sufficiently complex, extended over time and appropriate to its context; such that it is highly improbable to have been generated by coincidence. Examples of this form of evidence are found in primate (particularly ape) literature, such as in chimpanzee language experiments (e.g. Premack & Premack 1983). This has been typical of some of the sorts of research results roboticists report, and is the form of evidence provided in this section. Discussions and presentations of alternative methodologies occurs in the following two sections of this chapter.

The following report of the development of the Nomad behaviour is provided not only as evidence for the thesis, but also as an illustration of robot development. As such, it also provides material for the analysis of robotics research as a psychological methodology that appears in the last section of this chapter. On a gross level, development can be viewed as having alternated between creating new control scripts or tasks for the robot, and refining the behaviours (including the behaviour which is the control architecture) that these tasks relied on.

Original Goals for Nomad

One of the challenges for a complex agent is that it should be able to manage several potentially conflicting goals. An agent should take advantage of any good opportunities to meet a goal, but should also persist sufficiently in its pursuits that goals are consummated.

When planning the Nomad research, the initial draft set of goals were for the robot to do the following:

1. Monitor its own battery level, it should complain and go “home” (where it started from) if the level becomes too low.
2. Monitor its environment when moving, keeping its speed proportional to the nearness of obstacles.
3. Explore and map any unknown open area.
4. Stay near home when not otherwise occupied.
5. Monitor people approaching (detected by persistent range reduction without robot motion.)
6. If a person has approached and waved, follow them for as long as contact is maintained and the person is moving.

The intention of these goals was to have the robot be more or less instructable, so that a person could lead the robot into an unexplored area of the laboratory, then allow the robot to build a map of that territory. The idea was to essentially combine the work of Matarić (1990) and Smart (1992), which demonstrate robots learning maps in small arenas by sonar features and proprioception respectively, and Horswill (1993*b*), which demonstrates a robot navigating by vision and using a provided map to conduct tours over a large laboratory floor.

Unfortunately, all but the first of these goals proved significantly more time consuming than the literature might lead one to expect, particularly given that the robot platform was commercial rather than a home-built, thus reducing the time spent in construction and repair. Even with a map provided, the location re-calibration issues for a robot without vision proved more difficult than anticipated. In the end, only the first three goals were implemented, and the third was not entirely satisfactory.

Apparently, the map-building techniques of the first two were not sufficiently robust for a large, complex domain and extended trials. This is evidenced not only by the experience of this research, but also by the much more sophisticated sonar map learning techniques that *have* been proven fully robust and are reported in (Kortenkamp et al. 1998). Even using a map requires periodic localisation difficult in an ambiguous environment. This is a problem that vision, such as used by Horswill (1993*b*), makes simpler because it provides a richer sensor signal. Ongoing research in the laboratory is attempting to use unique sonar stamps for landmarks similar to the extremely low resolution visual images used as landmarks by Horswill. However, this sonar-stamp technique has also proven more reliable in relatively small arenas (a single room) and short time frames. This is probably due to the fact that the robot rotates absolutely, and consequently produces results too noisy for the machine learning algorithm. We are attempting to compensate for this by use of a compass. However the cumulative effect of this complex sensing routine has led to a serious increase in the number of time-consuming tasks the robot is required to do regularly simply to keep track if its place within its environment, and thus further slowed the rate of experimentation.

Nevertheless, the Nomad experiments do illustrate the ability of the robot to switch between goals, as unforeseen additional goals were added to the scripts.

Getting Started

The first task in the development of this research was to port Edmund to the Nomad robot. Other research on the robot had already established an interface to the robot's primitive sensing and action routines through a library written in the PERL programming language. Consequently, the first months of the project involved rewriting Edmund in PERL 5.003 (the first version of PERL with a working object system). Edmund had previously been written in the Common Lisp Object System (CLOS), which while a standard language for artificial intelligence, is not normally used for real time applications.

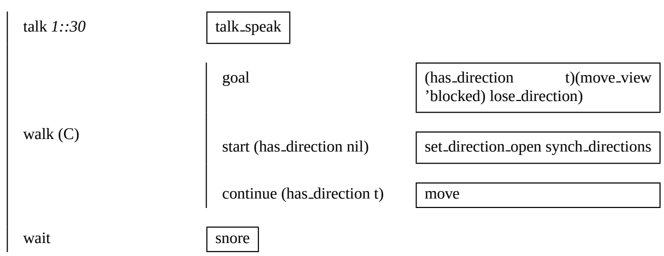
Rewriting the architecture also allowed the code to be cleaned and clarified, since the design and development process inevitably leads to progressively more arcane code as problems are found and solved in place (Brooks 1975). Examples of this were the simplification of the action pattern element format to being a simple sequence preceded by a trigger[2], and making recovery from habituation time dependent rather than decision dependent. This latter change reflects the move from a simulated environment with discrete "time" steps to a the realistic, constant model of time shared by robots and animals.

[2] The previous version had been more compartmentalised to allow for rational "crossover" if new behaviour-patterns were generated from old by genetic-algorithm like procedures.

The rewrite process took approximately two months, the time being partly spent coming to terms with a new programming language, the object system of which still had bugs. An additional two weeks were needed to fully connect the control architecture to the behaviour primitives provided by Nomad with the robot.

The First Script: Wander

Once Edmund's control architecture was operational on the Nomad, the task first chosen for development was the typical one for a behaviour-based robot — obstacle avoidance. In this task, the robot should move freely about its environment without colliding with any people or things. The initial control structure for this task was called **wander**, and looked like this:



In this diagram, the hierarchy of drive and competence reads from left to right, as do action patterns, which are in boxes. The left-most components above are actually elements of a single drive, called *life*. *Life* has no goal, and consequently never ends unless the program is killed. Its highest priority, in the absence of a goal, is the action-pattern *talk*. *Talk* actually stands in for the drive to eat, not communicate, as it reports on the robot's battery level as that level drops. The primitive action *talk_speak* is only executed once every 30 seconds. *Talk_speak* is a part of a battery monitoring behaviour, which contains state to examine whether the battery level has fallen significantly since last reported. Thus, *talk_speak* does not actually speak every 30 seconds, but is only given the opportunity to speak every 30 seconds. This checking of state could also have been checked in the control architecture, with an extra sense primitive, *battery_level_decline_since_last_reported*, but this seemed an unnecessary complication since the sense was only required by a single action. Such issues of design are treated further in Bryson & McGonigle (1998).

The next drive element, *walk* is a competence (indicated by "(C)") for moving. It has three members: a goal, which results in the motion terminating; a procedure for starting motion if stopped, and a procedure for continuing motion if already started. The "goal" in this situation is actually just a circumstance under which motion should stop — if the robot has moved too close to an obstacle in its way. The starting procedure is simply to chose a direction in which to move — in this case, the direction that seems to have the longest clear trajectory is chosen. The *continue* element runs the primitive *move*, described in the previous section, where it was part of a more complicated action pattern.

Because *walk* is not time sliced like *talk*, it uses all the available attention the robot can provide, unless it is not able to operate. Normally, the robot should always be able to find a direction to move in. In exceptional circumstances, when it is either boxed in or there is a bug in one of the behaviours, the competence will fail. *Wait* provides a bottom level drive which operates in these conditions. *Wait* simply produces a snoring sound, which communicates the robot's predicament to its handlers while keeping the drive *life* active, which allows for time to experiment. If all the elements of a drive fail, the drive (like other competences) will itself terminate with a failure.

Developing Motion

The major technical challenges of this stage of experimentation were debugging the new PERL version of Edmund, and developing the behaviour library to support **move**. This required early design decisions about the nature of the robot's motion. For example, many researchers choose to rotate the Nomad's turret to face consistently into the direction of motion, but for this research I chose to exploit the robot's ability to keep the turret's rotation absolute relative to the wheel base in order to make the sonar and IR readings' expected values relatively continuous over time. This assisted in disambiguating sensor noise from actual obstacles.

The most difficult design issue was the sensor fusion for the long-range sensors. This required not only design, but considerable experimentation. Previous work in the laboratory had abandoned the use of infrared sensors because their readings were too inconsistent to serve for obstacle avoidance. This in turn led to a large number of difficulties with manoeuvring in tight spaces, since the sonar sensors are inaccurate within 7 inches. Given the size of the robot, these "tight spaces" included doorways, where only 1 or 2 inches could be spared on either side of the robot. I determined through experimentation followed by contact to the vendor that the inconsistencies with the infrared detectors were due to poor circuit design which resulted in decreased sensitivity in daylight. This was corrected by creating several different infrared parameter files for the following conditions: sunny, overcast and dark. Unfortunately, in the daytime, it is impossible for the robot to move between exterior and interior spaces, that is, between daylight and interior lighting. This is because these files have to be set at boot time, so cannot be changed dynamically by the robot's architecture.

Even with this correction, designing an algorithm sufficiently robust to allow the robot to pass through doorways was difficult and time consuming. Previous efforts to achieve this by any means other than vision or elaborate exploration had often failed, including some previous efforts to fuse sonar and infrared (Zelinsky 1991). The final algorithm used in these experiments reduces to the following:

1. *Determine a sonar value.* Take the current sonar reading. If it is within $tolerence$ distance of the previous accepted sonar reading, accept it, where

$$tolerance = \frac{maximum\ speed}{minimum\ sonar\ sensing\ rate}.$$

   If the current sonar reading disagrees with the last accepted reading, but agrees with the two preceding readings, accept it anyway. If the current reading is rejected, compute the expected sonar reading based on the previous accepted reading and its average change over the last three accepted readings (simple difference). This strategy still required careful selection of a sonar fire order and firing rate. The need to sense frequently had to be compromised slightly in order to largely eliminate the interference in navigation of sonar "ghosts," or false reflections from neighbouring sonars.

2. *Determine the fused long-range sensor readings.* If the infrared reading is less than 12, take whichever of the two sensor readings is lowest (representing the nearest obstacle). The value 12 was determined by experiment as the highest infrared value likely to be scaled roughly in inches, or in other words on the same scale as the sonar. The sonar gives a more reliable distance reading and is accurate to at least 7 inches, but it may not see obstacles detected by the infrared due simply to their differing heights on the robot's turret. Consequently, this step errs on the side of safety, even though it reacts more strongly than necessary to the white glossy radiators scattered around the laboratory.

3. *Scale readings by relevance to direction.* Sensor readings more than 33 degrees to the left or right of the direction of motion are multiplied by four, thus making the obstacles

appear more distant. Those more than 56 degrees to either side are multiplied by six, those more than 78 degrees are discarded. This bias allows a single set of parameters to determine the danger of the robot's current trajectory, setting the speed and determining stopping distance, while allowing for the fact both that the robot is round, and that looming obstacles that are not being approached directly may be passed.

This algorithm combined with that for motion allowed crossing the threshold of doors to be a robust primitive — the robot need only be travelling in the general direction of an open doorway, and it would naturally, in a smooth continuous fashion, reduce its speed, move towards the centre of the door and pass through. Note that although the above algorithm is described linearly for clarity, the actual implementation of the algorithm is behaviour based, as described previously in Sections 4.2.3 and 4.2.4.

Overall, the wander task is very successful, and has become the fastest and easiest way to move the robot around the laboratory. Direction is provided by herding — standing behind or in the way of the robot, thus encouraging it to move in the opposite direction. Developing this robustness took approximately two and a half months, with additional sporadic debugging over the next year. The complete process of becoming familiar with the robot and its primitives, rewriting the architecture, writing the first script and creating and debugging these first motion primitives took approximately six months.

Scaling Up

One of the most important claims for an agent architecture is that it facilitates the creation of intelligent behaviour. This was certainly true for the Edmund architecture over the course of the Nomad experiments. Existing behaviours and primitives were quickly and easily recombined through varying the control structure. For example, simply adding the *correct-dir* primitive, described above, to the "wander" script resulted in an elegant looping behaviour from the robot once it found a wall, as it would search for a way through. This emergent behaviour was a consequence of the fact that when the robot would find a wall or door, it would follow this obstruction to either its right or left until reaching the next obstacle. As it began to move around that obstacle, *correct-dir* would notice a shorter route back to the original direction through looping. The script that produced this behaviour was called "fly", due to the resemblance of the robot to a fly on a window (though in slow-motion). This looping was curtailed in the script “directed-wander”, which would give up the search for an opening rather than turning too far from the preferred direction.

“Directed-wander” was actually a hybrid between “fly” and “ask-direction”, another modification of “wander”. “Ask-direction” added a single primitive *ask_direction* for choosing the initial direction the robot moved in. As the name implies, the robot would voice the question “which way?”, which would have to be answered at a computer console. The answer took the form of a number between 1 and 16, corresponding to a Direction to which attention was directed. In “wander”, the Direction with the longest clear view in front of it had been selected in response to the primitive *set_direction_open*.

Some substantial changes in expressed behaviour cannot be brought on simply by restructuring control. Gross new capabilities require the addition of behaviours, in the “behaviour based” sense of the word. Several new behaviours were added to the robot over the course of experimentation.

- **Bump** When these experiments were begun, the robot did not have bump sensors. It rapidly became obvious that bump sensors would be useful for two reasons: first to detect the class of objects invisible to the two types of range sensors (see Figure 4.1), and second to provide perceptual input that clearly required short term memory. The information provided by the bump sensors, a single point contact, is significantly different from the continuous data provided by the range sensors. Consequently, a new behaviour or intelligence module, was needed. A Bump behaviour remembers a collision, and affects perceptual memory (see below). A dedicated action pattern responded to the sensation of a bump as reported by the robot, and both created a Bump behaviour, and operated a sequence of motions (through Directions) to disengage the robot’s rubber bumpers from the obstacle.

- **Wave** Wave was an attempt to communicate directly to the Nomad’s Directions through the sonars, rather than through the keyboard. Wave was separate from Direction because of the very different nature of the signal to be recognised. It did not prove to be practical on a moving robot, due to the difficulty of persistently signalling only a single face.

- **Pmem** Pmem, for “perceptual memory”, is the behaviour that fused current sonar and

IR readings with very short term recall of past sonar readings to provide a coherent impression of obstacles in the vicinity, as described earlier. Bumps violate Pmem's encapsulation to further enrich the information with short term memory of obstacles. Pmem originally violated encapsulation of the Directions in order to get the raw sonar and IR readings. However, it was eventually realized that having Direction dependent on hardware feedback was slowing the system. At this point, direct sonar and IR reading became the domain of Pmem. This function was then scheduled through the control architecture as the primitive "compound-sense" ("csense" in Figure 4.7).

- **DpLand** Finally, a set of behaviours was created to attempt to recognise decision points and recall the correct behaviour when they were reached.

Although the Edmund strategy of modularising these behaviours and leaving their integration largely to the easily-created control scripts greatly facilitated development, many of the capabilities described above took a considerable amount of time to develop, even as simple stand-alone behaviours.

4.2.6 Discussion

There are two expressions for the standards of properly reactive robots. The first is essentially an aesthetic sense that the robot moves smoothly, in a believably animal like way. This is the standard set out in Brooks (1990). The second is a practical argument of the necessary capabilities for robot system, such as the following:

> The robot must not only be able to create and execute plans, but must be willing to interrupt or abandon a plan when circumstances demand it. Moreover, because the robot's world is continuously changing and other agents and processes can issue demands at arbitrary times, performance of these tasks requires an architecture that is both highly reactive and goal-directed. (p.667 Georgeff & Lansky 1987)

By both the standards, the experiments conducted using Edmund on the Nomad robot were a success. Figure 4.7 shows the top few levels of one of the final scripts created in the course of this research. Notice that despite the major behaviour additions mentioned above, the changes to the control structure are very succinct.

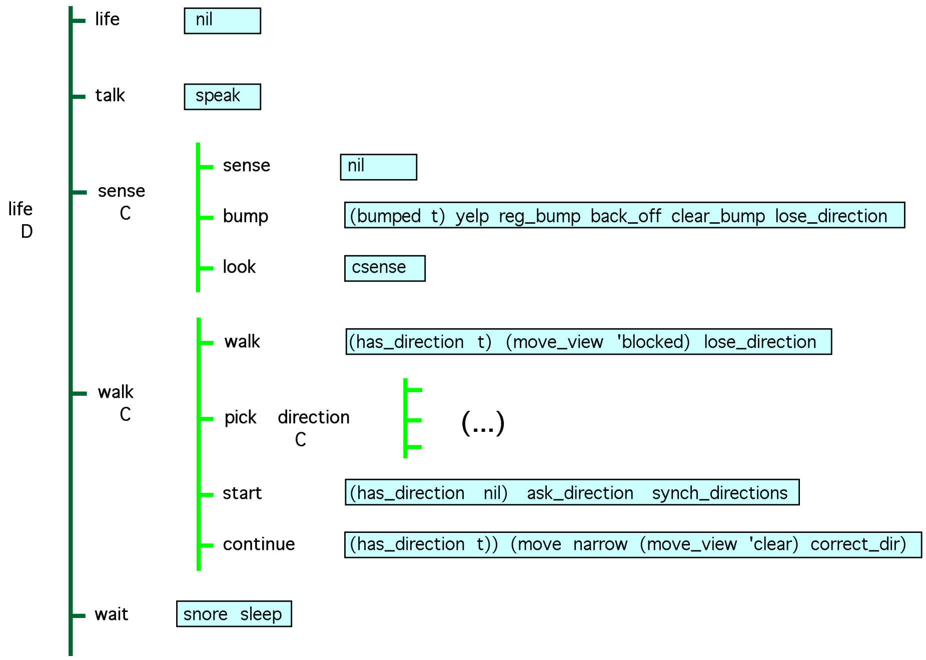


Figure 4.7: The control script for "proto-nav". Priorities are not shown, but the initial priority is ordered from top to bottom. Drives are labelled **D**, Competences **C**. Senses are paired with test values, unpaired primitives are Acts.

A new distinct goal structure, in the form of the drive element *sense* has been added. This facilitates both the Bump and the Pmem behaviours mentioned above. The Direction behaviours operating under the *walk* drive element now use reliable perception rather than direct sensing. The other major change is the addition of the *pick direction* competence as an element of the *walk* competence. This allows for the testing of several navigation strategies prior to the robot conceding to ask a human directions with the *start* competence. The navigation routines were never adequately stabilised, but the typical competence had the elements:

- *pick_neighbor*, which recalls a decision made near to where the robot is currently located,
- *pick_continue*, which continues in the same direction, and
- *pick_bigneighbor*, which recalls a decision made in a wider region around the current estimated location.

The above improvements took an additional seven months of research time beyond the six months that led to the initial successful "wander" script. Two of these months were spent in a second translation of the control architecture and established behaviour libraries into the language C++. This new version of the control architecture was three orders of magnitude faster than the PERL implementation for computational tasks, and one order faster for tasks involving direct sensing. However, the process of prototyping new behaviours was significantly slower in C++, due to the greater rigours of the language.

The results of the robot experiments presented here are further discussed as the motivation for the comparative research described in the following section, and also in the concluding section of this chapter, which discusses methodology.

4.3 Direct Architecture Comparisons in a Simulated World

4.3.1 Motivation

As mentioned at the end of Section 4.2.1, research on the robot, while validating to some extent the reactiveness of the architecture, was not completely satisfactory. Edmund was intended to support modelling of complex animal behaviour, in order to support psychological experiments and facilitate research dialogue between researchers in behaviour-based robotics and those working with natural intelligence. However, the most natural route towards simulating animals with a mobile robot, navigation (see Trullier et al. 1997, for a review), was severely limited by the robot's sensor capabilities. We found ourselves in the situation of painstakingly reimplementing already established methods for compensating for the sensor configuration. For example, the navigation achieved was significantly less robust than that already reported by Thrun et al. (1998), but establishing this kind of reliability with our robot requires adding a compass and significant computational power for doing statistical work on the sonar readings. This process would be very time consuming, and worse, lead to no immediate advantage — once the work was replicated, it would be difficult to compare the results against the original performance based on the sparse robotics results reported in the literature. These issues are examined more extensively in Section 4.4.

Two events led to a radically different approach for the next phase of evaluation and comparison of Edmund. First, I was asked to assist another researcher, Robert Ringrose, who needed an architecture to control a simulated humanoid robot. The simulation represented a sitting infant, the task was to study a neural network model of learning to coordinate visual input and grasping. The simulation had previously been controlled with simple code, but the task of the infant was complicated by the addition of a single extra degree of freedom: if the infant was unable to view its target due to an occlusion, it was meant to attempt to bend sideways from the hip to observe the target. Because the robot was meant to be learning its capabilities, the cognitive model was meant to support retrying a strategy several times in case of failure, then in case of repeated failure switching strategies again. Encoding this simple problem revealed several coding bugs in the C++ version of Edmund's control architecture[3]. The bugs were in the code that allowed one competence to transition to another — a key theoretical aspect of Edmund which had been used frequently in its blocks-world simulation, but not in over a year of robot programming.

This event led to the realizations that:

1. the robot platform, while challenging perceptually, had allowed relatively little complexity in hierarchical or sequential control: in a year, there had been no programs of over 3 layers of depth hierarchically; and
2. simulated tasks can make interesting demands on the architecture.

. The second event alluded to above was the finding of a thesis dedicated to comparing alternative action-selection mechanisms (Tyrrell 1993). The comparison was done in simulation, and the simulation itself was preserved in several Internet archives. The simulation provides substantially more goals, many of which are conflicting, and radically more dynamicism of the agents' environment than has ever been documented in real robot research. The thesis evaluates several well-known architectures, including Maes (1989, 1990*a*). This section documents the comparison of Edmund to the other architectures previously evaluated by Tyrrell.

4.3.2 The Simulated Environment

Tyrrell, in his dissertation (Tyrrell 1993), creates a task in a simulated environment which requires an architecture to manage a large number of conflicting goals. He defines an envi-

[3] These were fixed, though in the meantime Ringrose moved to using a Markov model controller instead. Ordinary Markov models provide less realistically organised cognitive control than Edmund or the other previously mentioned BBAI architectures. However, the nature of the statistical learning involved did not require rational behaviour on the part of the model, only visiting each possible behaviour at approximately the correct frequency.

ronment in which a small omnivorous animal is required to survive and breed. He defines six “subproblems” for the animal to solve.

1. Finding food and water. There are three types of food and three forms of nutrition satisfied in varying degrees by the different food types.

2. Escaping predators. There are both feline and avian predators, which move at different speeds and have differing powers and problems of perception.

3. Avoiding hazards. Benign dangers in the environment include ungulates, cliffs, poisonous food and water, and temperatures varying above and below ideal during the period of the day. 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. Mating. The animal is assumed to be male, thus its problem is to find, court and inseminate mates. While it is hazardous to try to inseminate unreceptive mates, rearing of offspring is considered to be the problem of the mate.

6. Sleeping at home. The animal is essentially blind at night; sleeping in its den is important for avoiding predators and for avoiding being trodden on by ungulates. It is also necessary for maintaining temperature while conserving energy.

These problems vary along several metrics: homeostatic vs. non-homeostatic, dependency on external vs. internal stimuli, periodicity, continual vs. occasional expression, degree of urgency and finally, whether it is prescriptive or proscriptive with regard to particular actions. In addition, the environment is dynamic and sensing and action uncertain. Perception in particular is extremely limited and severely corrupted with noise. The animal usually misperceives anything not immediately next to it, unless it has chosen to expose itself by looking around in an uncovered area. Finally, action is also uncertain, particularly if the animal is unwell.

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.

Tyrrell checks the independence of his results from these parameters by running all his models in four worlds. Besides the standard model he first designed, there are three variant worlds. According to the thesis (p. 162) these changes are:

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 taken, and by changing the conspicuousness and energy consumption of various actions.

4.3.3 Tyrrell's Extended Rosenblatt & Payton Architecture

Tyrrell implemented and tested four theories of action selection, often extending and providing necessary and obvious modifications to the theories in the attempt to make them operate on a real system. The first was drive theory ("simple motivational competition") as described by Hull, the second combined Lorenz's two "Psycho-Hydraulic" models, the third implemented Maes' spreading activation networks, and the fourth Rosenblatt and Payton's "connectionist, hierarchical, feed-forward network." Tyrrell's champion is a modified version of the latter architecture, which he examines and defends as being as close to optimal action selection as possible. Lorenz's simple and explicitly ethological Drive theory was placed a fairly close second, whereas the only previously fully implemented, simulation-based AI architecture, Maes's spreading activation network, was by far the poorest performer. Shortcomings in Maes architecture were already discussed in Section 3.5.1.

The Rosenblatt & Payton model had, like Edmund, been developed and proven as a robot architecture (Rosenblatt & Payton 1989). The main feature of their 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. The ultimate leaf nodes of the tree are actions, which are activated on a winner-take-all basis. What

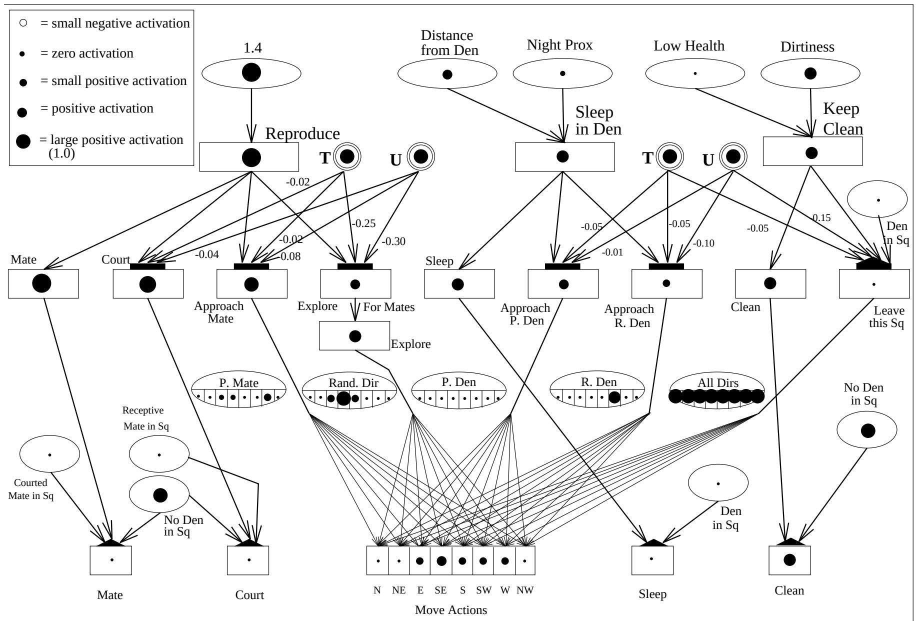


Figure 4.8: A fraction of the Extended Rosenblatt & Payton action selection network Tyrrell developed for controlling an animal in his simulated environment. This shows the interaction of the hierarchies associated with reproduction, staying clean, and sleeping in the den. **T** and **U** are temporal and uncertainty penalties for actions that take time or pursue a goal uncertain to succeed. "P." stands for perceived, and "r." for remembered. From Tyrrell (1993), page 247.

differentiates the model from the drive model or other normal hierarchies is that no decisions are made until the leaf or action nodes. This allows for *compromise candidate* behaviours to be selected, that is, behaviours that satisfy multiple drives. The model is distinguished from Maes's network by its lack of preconditions (or, in the ethology literature, "releasers"), and the fact the energy flows in only one direction.

An example of a section of Tyrrell's ERP hierarchy for his simulation is shown in Figure 4.3.3. The internally gauged distance from the den and the externally cued night proximity both feed the high-level node "Sleep in Den", which in turn sends positive energy to nodes like "Approach Remembered Den" and "Approach Perceived Den" and "Sleep." For the remembered den temporal and uncertainty penalties send negative activation to the node as well. Many nodes send weights to the various possible directions the animal can move in; care must be taken that motion doesn't always triumph over consummatory acts with few sources of input.

Tyrrell's main extensions to Rosenblatt & Payton's original architecture are to add penalties for temporal delays and uncertainty of rewards for items that are distant or based on poor memory or perception. The architecture then tends to exploit certain local reward rather than choosing to pursue uncertain but potentially larger goals. All perception in the simulation is real valued, so food sources of the same type in different locations will have different effects on the levels of activation energy to the network. He also added explicit penalties to appetitive vs. consummatory actions. Finally, Tyrrell created 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 last idea has been recently carried further by Humphrys (1996), which discusses various action-selection mechanisms based on considering positive and negative opinions of multiple goal-embodying behaviours.

4.3.4 Comparing Edmund to the Extended Rosenblatt & Payton Architecture

Understanding the Problem

The first step was to recompile Tyrrell's simulation and his Extended Rosenblatt & Payton (ERP) to it, and to test that my results for running this architecture matched the thesis. I decided to test only to a 95% certainty of being within 10% of the correct result, that is to run only 245 simulations. I actually did this twice, with the results:

Run	Mean	Std. Dev.	*SE*
1	8.94	8.33	0.54
2	8.14	7.78	0.25
Thesis	8.31	6.63	0.16

The fitness measure here represents the average number of times an animal was able to copulate before dying. Although these values do overlap each other's ranges within the required ten percent interval (and Tyrrell's reported result is also situated within that overlap) this shows the high degree of variability still possible, despite the small standard errors.

Since the ERP had been developed by the same person who wrote the simulator, I determined to test both competing architectures in an additional new scenario, a different set of simulation parameters, as well as the reported sets. I intended to try to emphasise problems requiring systematic ordering of behaviour or persistence to a goal, since these are particularly difficult problems for reactive architectures. However, analysis showed that very little in Tyrrell's simulated environment requires persistence. Mating is the only action which requires multiple steps: getting in the same square with a receptive female, courting and then consummating.

Because the potentially elaborate courtship ritual had been reduced to a single act, and the receptive state of the female may be sensed, each of these elements may actually be treated as a separate reactive task.

The simplest way to test persistence appeared to be to substantially reduce the food supply. In the original test environment, food and water are both plentiful, and likely to be available nearby to any given square. Decreasing the probability of local availability of food would presumably increase the probability of dithering between two distant goals, which should lead to decreased efficiency and therefore genetic fitness.

In fact, however, even severe reduction of food had relatively little impact on the performance of Tyrrell's animals. Eating turns out not to be a particularly important part of life, because average life expectancy is fairly short, food is plentiful, and energy consumption is low. Consequently, changing the range of possible initial amounts of the various food types as follows:

Food	Original	Sparse
fruit	[50, 81]	[5, 35]
cereal	[45, 71]	[2, 27]
prey	[25, 36]	[2, 15]

where the range is square-bracketed, resulted in only a slight loss of fitness:

Run	Mean	Std. Dev.	*SE*
Sparse	6.09	6.95	0.09

Linear regression across the availability of food, water, cover and the initial number of mates showed only a few barely perceptible dependencies. The amount of water varied from 4–36 instances, cover from 0–50, and initial mates from 1–23.

Variable	gen. fit.	water	cereal	cover	fruit	mates	prey	lifespan
gen. fit.	1.000							
water	0.026	1.000						
cereal	0.083^{*}	-0.018	1.000					
cover	0.046	0.013	0.025	1.000				
fruit	0.206^{**}	-0.039	-0.008	0.023	1.000			
mates	0.038	-0.009	-0.012	-0.032	0.057	1.000		
prey	0.034	0.003	0.034	0.019	0.061	-0.016	1.0000	
lifespan	0.912^{**}	0.022	0.081	0.070	0.231	-0.010	0.018	1.000

The values on this chart show the correlations between the amount of a resources in the home range of an animal, the number of days it lived, and its “general fitness” (the number of times it copulated.) One asterisk denotes a correlation of significance at the 0.05 level ($p < .05$), and two asterisks indicates $p < .01$. As expected, there are no correlations between the availability of the various resources, since these are determined randomly by the program and do not affect each other. The number of initial mates is probably less significant because mates wander in and out of the animal’s territory continuously, so their initial numbers are not particularly relevant to their overall prevalence.

Implementing a solution in Edmund

The development of an Edmund architecture for the Tyrrell Simulated Environment happened in two phases. The first was a period of rapid development without testing of initial versions of the primitive senses and actions for the architecture. The initial actions were: sleep, hold_still, pick_safe_dir (for fleeing), go_fast, go, look_around, pick_home_dir, copulate, strut, pick_mate_dir (for pursuing a mate), exploit_resource (take advantage of anything in this cell that the animal needs) and pick_dir (the general case for direction selection.) Senses were: at_home, covered, see_predator, late (in the day), day_time, fit_to_mate, courted_mate_here, mate_here, perceive_mate, and needed_resource_available. As may be clear from this selection, the main problems addressed individually were fleeing predators, pursuing mates, and getting home before dusk. It was assumed other needs could be met opportunistically when nothing else was troubling the animal.

The next phase of development involved testing which behaviours were actually useful and in what order of priority, and tuning the behaviour of the primitives. The first step was the most basic program that could generate a genetic fitness measure – one that only pursued mating. Thus the drive “life” had a single competence “mate”, which consisted of:

mate (C)	inseminate	(courted_mate_here t) copulate
	court	(mate_here t) strut
	pursue	pick_dir_mate go

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, immediately outperforming Tyrrell’s best implementation of Maes (1991). Other obvious behaviours were added gradually, testing the new additions and ordering on approximately 800 runs. Simplicity was favoured unless the program improved very significantly. After this phase of development the components of life in order of priority were:

flee (C) (sniff_predator t)	run_away	(see_predator t) pick_safe_dir go_fast
	freeze	(see_predator t) (covered t) hold_still
	look	observe_predator
mate (C) (sniff_mate t)	*as above*	
home *1::5*	(late t) (at_home nil) pick_dir_home go	
check *1::5*	look_around	
exploit (C) (day_time t)	use_res	(needed_res_avail t) exploit_resource
	leave	pick_dir go
sleep_at_home	(at_home t) (day_time nil) sleep	

Homing in the evening and looking for predators take priority intermittently, but habituate to allow other, lower priority behaviours to express themselves immediately after they had 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 graphic user interface. This was done over approximately 80 animal life-times. This resulted in the following observations and changes:

- The animal seemed too concerned with staying in shelter and failed to explore much unless chasing mates. This was altered by reducing the desire for shelter in pick_dir.
- The animal was often killed by feline predators while freezing. Consequently we added a prohibition to the freezing action pattern that prevented freezing in the presence of felines.
- Once the animal was less concerned with staying in shelter, it sometimes explored to such a degree that it lost track of its bearing entirely and could not recover by nightfall even though it began looking for its den with plenty of time. This was solved by adding the action pattern:

triangulate	(lost t) pick_dir_home go

  which had the same priority as homing above, but was active all day and did not habituate. Thus, once the animal becomes slightly disoriented, it goes towards its den until it either recognises its own location or finds the den (unless it happens across a mate or a predator.)

- The animal often had unforced accidents with irrelevant animals and cliffs, particularly while it was pursuing mates. This was solved by altering the behaviour pick_mate_dir to increase the avoidance of these problems.

Some observations were ambiguous. For example, because the animal had very erratic perception, it frequently hallucinated a predator nearby. Attending to any trace of a predator disrupted behaviour coherence sufficiently to interfere with mating, while paying too little attention to predators reduced mating opportunities by reducing the expected lifespan. This problem was addressed by running a large number of simulations (4400) with randomly selected values for six critical thresholds: four for detecting predators (observing or fleeing each type), one for seeking shelter and one for avoiding dangerous places.

Linear regression performed on this data was uninformative, because the variables' relationships to the animals' success were neither linear nor independent. Taking inspiration from the genetic algorithm approach to problem solving, I 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 animals' mean performance by 25 percent.

Edmund and the ERP were then tested, both under the conditions of Tyrrell (1993), with his four sets of environmental parameters for the simulation, and under the second set of parameters with considerably sparser food supplies mentioned earlier. Under the above control system, Edmund out-performed the ERP in this sparse world, except for variant world 3. Examination of the simulation code disclosed that the largest variation in this world was a very significant reduction in energy consumption for certain activities (such as mating and running) and significant changes in the conspicuousness of some behaviours, both positive and negative. Conspicuousness leads to increased danger from predators. The ERP's superior performance under these conditions, implied Edmund's strategy was wasting too much time or conspicuousness on grazing. Consequently the **exploit** competence was divided into constituent parts, incidently eliminating the one case of "compromise candidate" determination in Edmund's solution. Pick_safe_dir (the predator fleeing behaviour) was also altered similarly to pick_mate_dir above to attend to environmental dangers. These changes resulted in a system that surpassed Tyrrell's ERP in both the sparse food and the original versions of the SE, excepting variant world 3 to which the ERP solution seems particularly well adapted.

4.3.5 Results

Edmund performed better (that is, mated more), but not significantly so (Critical Ratio = 1.36) in the set of world parameters chosen by Tyrrell. However, it was very significantly better than Tyrrell in every world except for variant world 3, and also better than any other architecture reported in Tyrrell (1993). The final results (over 6600 trials) were:

World	Edmund	ERP	CR
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.

In the set of worlds where food was significantly scarcer (see 89) the Edmund agent did perform significantly better overall than the ERP one, though again being beaten in the third variant world.

World	Edmund	ERP	CR
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^{**}

Notice that Edmund succeeds even when perception was reduced, in variant world 1. This is surprising, since perceptual ambiguity should be more damaging to Edmund's more ordered action selection. The ERP is essentially always seeking the optimal action, thus perceptual ambiguity affects all behaviour equally, whereas Edmund has different active goals and concerns. One of Edmund's advantage as an architecture lies in reducing the complexity of design for the programmer and focusing attention for the animal: actions that require an ordering can be set to one explicitly. The deficit in the third world shows that Edmund's animal probably still wastes too much time and/or exposure to predators in stalking food, which indicates the current control system is not sufficiently flexible to adapt to those changed circumstances. Again, however, the ERP animal, despite considerably more information, has more trouble adapting to the inverse situation of less plentiful food. In fact, under linear regression, Edmund's animal failed to exhibit even the slight dependency on food availability that the ERP's animal had shown, with absolutely no significant correlations other than between genetic fitness and lifespan.

Variable	gen. fit.	water	cereal	cover	fruit	mates	prey	lifespan
gen. fit.	1.000							
water	0.033	1.000						
cereal	0.027	0.017	1.000					
cover	0.009	-0.027	-0.040	1.000				
fruit	0.032	0.016	-0.001	-0.053	1.000			
mates	0.049	0.032	-0.037	0.005	-0.052	1.000		
prey	0.014	0.057	-0.009	-0.004	-0.017	0.028	1.000	
lifespan	0.924^{**}	0.019	0.052	0.013	0.027	0.015	0.030	1.000

Another interesting result is the complexity of the engineered control systems. Edmund's solution required 20 primitives that were action-oriented, and 22 dedicated sensing primitives. It also had 7 competences and 23 action sequences defined in its final program script. These can be seen as intermediary behaviours when considering organisational complexity. The ERP solution had 61 sensing nodes and 278 other nodes, of which 216 were intermediate. As another metric of complexity, 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.

4.3.6 Discussion

The research in this section demonstrates that a system using selective, hierarchical control which patently ignores most of its sensory input at any particular time can out-perform a carefully designed fully-parallel architecture. There might be reason to be cautious in over-interpreting this result. For example, the system built under Edmund was very sensitive to some parameter tweaking, particularly the threshold at which predators were attended to. Given the sorts of thresholds in its implementation, this seems to be also true of the system built under ERP, but it is never a desirable trait in an AI system. Nevertheless, both the relative lack of complexity of the Edmund animal's control and its relative success serves as strong evidence that managing complexity through selective attention is more important than having a fully reactive architecture.

Reducing information is the primary task of any perceptual system; it is inseparable from the problem of gathering information in the first place. Further, there can be no doubt as to how crucial perception is to the intelligence of a behaviour-based artifact, and little more of the role it plays in natural intelligence. Nevertheless, it seems unlikely that simply the loss of information is itself sufficient or necessary to the improved behaviour. Rather, it is more likely that ignoring some information reduces the complexity of the design problem, and thus improves the probability of developing a better system. The design problem does not exist simply because hand-coded architectures were used for attacking the SE. Any form of learning, including evolution, has a similar search space to traverse as an intelligent designer — for all systems, the simpler design is easier to find.

4.4 Analysis of Approach

The preceding sections have presented experiments which demonstrate the claims of the Edmund hypothesis. They also provide evidence that the detractors of such structured control are incorrect: hierarchy and sequence can be used to control agents at least as capable of coping with dynamic situations as more purely reactive architectures. On the other hand, they have not produced conclusive proof of the hypothesis stated in Section 3.4.2. Science, unlike engineering or mathematics, rarely yields conclusive proof; however, further evidence from psychological research and suggestions for further experiments are both offered in the next chapter.

This section evaluates the evaluation of Edmund. Perhaps more importantly, it addresses the contentious issue of using robots as research platforms, taking the results of the two previous sections as evidence. It begins by examining the arguments for the use of robots, concluding that robots are neither necessary nor sufficient for valid research in behaviour-based AI. It then discusses the aspects of action selection that have and have not been adequately tested in this chapter, and describe further potential platforms. It concludes with a brief summary of the circumstances under which robots may still be a desirable part of a research inquiry, using the above sections as illustration.

4.4.1 Historic Motivation for Robots as Research Platforms

Alan Turing, one of the founders of both artificial intelligence and computer science, invented the best-known test for the attainment of "true" AI. Now known as the Turing Test, the criterion consists essentially of having several judges unable to determine whether they are communicating with a computer or a human (Turing 1950). The reason for this criterion is straightforward; it is functional, and therefore open to simple evaluation. Turing did not expect that AI would be able to exactly imitate the computations of the human brain on a radically different architecture, but believed that functional equivalence should be possible.

Turing also believed the most probable way of achieving such functional equivalence was by making the artifact as like a human as possible, so that it could share as much of human experience as possible (Turing 1970). He observes that even with a mechanical body and anthropomorphic sensing, the computer would be unable to share the experiences of eating fine food, becoming drunk or falling in love, but should none-the-less be significantly more likely to acquire human-like intelligence than otherwise. This argument is repeated in inverse by Dreyfus (1992), who is critical of the notion of true AI. He observes that without being immersed in human biology and culture, there is no reason to believe an artifact, even if it were intelligent, would be recognisably so by us. As an example, he points out that analogy is very important to human reasoning, but asserts that the axes of analogy are relatively arbitrary. For instance, humans tend to take mirror symmetric objects as being nearly identical. What would happen if the computer, instead, chose the colour red as an important axis of similarity[4]? Arguments of this nature have led to a recent trend to root AI not only in robotics, but in humanoid robotics (Brooks & Stein 1993, Kuniyoshi & Nagakubo 1997, Dennett 1997, Brooks et al. 1998).

The behaviour-based AI community, though often disinclined to attempt to create full humanoid intelligence, has historically been fully invested in demonstrating their theories on robots. Below are several reasons for this approach, derived mainly from Brooks (1991*b*,*a*), Steels (1994).

A AI has made substantial progress on specialised problems such as chess and theorem

[4] Shortly after I first read this argument, a friend told me (without prompting) that his four-year old daughter had just surprised him by announcing that she and a family friend, an enormous man of a different race, were "just the same." When my friend asked her what she meant, she said "We're both wearing red shirts!"

proving which have not proven generally useful to creating intelligence. One hypothesis about this failure is that the solution space for any one task is under-constrained: there may be many ways to do the task, but not all of them involve computational strategies that could be a part of a human-like intelligence. One way to force appropriate constraints is to build a *complete* agent, one that solves all the problems of animal-like intelligence. Robots necessarily have to address most of the issues of daily animal existence: perception, motion, action selection and navigation. They may also be asked to find their own food (at a recharging station or power point) and in theory, even to replicate themselves.

B By a similar line of argument, animal-like intelligence would best be achieved in solving the subtasks of intelligence in the order they are found in nature. For example, it may be necessary to have a primate-like vision system before addressing language. Again, robots are more appropriate than simple computer platforms for developing these critical systems, because they more nearly share our environment, and therefore our selective pressures.

Even if researchers are persuaded by these arguments that we should build complete agents which achieve animal-like skills in natural environments, it would be far easier to address these problems in the better controlled environment of an artificial simulation, run entirely on a computer. In this case, one program would create the world and its problems, while other, experimental programs would solve these problems. The following criticisms have been levelled against this approach.

C No simulation replicates the full complexity of the real world. In choosing how to build a simulation, the researcher first determines the 'real' nature of the problem to be solved. Of course, the precise nature of a problem largely determines its solution. Consequently, simulations are only a special case of the first point above, (*A*). They enable a researcher to reduce the full problem of intelligence to particular subproblems.

D Alternatively, if a simulation truly were to be as complicated as the real world, then building it would cost more time and effort than that saved by not building the robot. This argument assumes one of the axioms of the behaviour-based approach to AI, that intelligence is by its nature simple and its apparent complexity only reflects the complexity of the world it reacts to. Consequently, spending resources constructing the more complicated side of the system is irrational as well as unlikely to be successful.

E With a simulation researchers may deceive either themselves or others as to the complexity of the AI solution they have provided. Since both the problem and the solution are under control of the researcher, it is difficult to be certain that neither unconscious nor deliberate bias has entered into the experiments. A robot is a clear demonstration of an autonomous artifact, its achievements cannot be doubted.

4.4.2 Criticism of Robots as Research Platforms

The arguments presented in the previous section are the main formal reasoning behind the adoption of the autonomous robot as a research platform. This adoption has occurred in a large and growing community of researchers, despite several well-recognised difficulties with the platform. This section begins with a review of these difficulties, and then moves to address whether the benefits espoused above have proven to compensate for these difficulties.

The main problem with robots as a research platform is that they are extremely costly. Although their popularity has funded enough research and mass production to reduce the initial cost of purchase or construction, they are still relatively expensive in terms of researcher or technician time for programming, maintenance, and experimental procedures. Consequently, they are of significantly more net value in a context where research or education is intended to include work in hardware and software design than in laboratories dedicated to work in psychology and cognitive modelling.

Further, the difficulties of maintaining an operational plant reduce the utility of the robot for psychological experiments by making it difficult to run enough experiments to have statistically significant results when comparing two hypotheses. This situation has been improved by the advent of smaller, more robust, and cheaper mass-produced robot platforms. However, these platforms more often fall prey to a second criticism: mobile robots do not necessarily address the criticisms levelled against simulations above (*C*–*E*) better than simulations do. This is caused by two factors: the simplicity necessary in the robots to make them usable, and the growing sophistication of simulations.

The constraints of finance, technological expertise and researchers' time combine to make it highly unlikely that a robot will operate either with perception anything near as rich as that of a real animal, or with potential actuation with anything like the flexibility or precision of even the simplest autonomously animate animals. Meanwhile, the problem of designing simulations with predictive value for robot performance has been recognised and addressed as a research issue (e.g. Jakobi 1997). All major research robot manufacturers now distribute simulators with their hardware. In the case of Khepera, the robot most used by researchers running experiments requiring large numbers of trials, the pressure to provide an acceptable simulator seems to have not only resulted in an improved simulator, but also a simplified robot, thus making results on the two platforms nearly identical. While some researchers claim this similarity of results validates use of the Khepera simulator, in absence of other evidence it can be equally argued to invalidate the robot.

When a simulator is produced independent of any particular theory of AI as a general test platform, it defeats much of the objection raised in charges *C* and *E* above, that a simulator is biased towards a particular problem, or providing a particular set of results. In fact, complaint *E* is completely invalid as a reason to prefer robotics, as experimental results provided on simulations can be replicated exactly, thus proving more easily tested and confirmed than those collected on robots. To the extent that a simulation is created for and possibly by a community — as a single effort resulting in a platform for unlimited numbers of experiments by laboratories world-wide, that simulation also has some hope of overcoming argument *D*. This has been particularly true on two platforms. First, the simulator developed for the simulation league in the Robocup football competition has proven enormously successful, and to date provides much more "realistic" football games in terms of allowing the demonstration of teamwork between the players and flexible offensive and defensive strategies (Kitano et al. 1997, Kitano 1998).

The second platform is independently motivated to provide the full complexity of the real world. The relatively new commercial arena of virtual reality (VR) has provided a simulated environment with very practical and demanding constraints which cannot easily be overlooked. Users of virtual reality bring expectations from ordinary life to the system, and any artificial intelligence in the system is harshly criticised when it fails to provide adequately realistic behaviour. Thórisson (1996) demonstrates that users evaluate a humanoid avatar with which they have held a conversation as much more intelligent if it provides back channel feedback, such as eyebrow flashes and hand gestures, than when it simply generates and interprets language. Similarly Sengers (1998) reviews evidence that users cannot become engaged by VR creatures operating with typical behaviour-based architectures, because the agents do not spend sufficient time telegraphing their intents or deliberations. These aspects of intelligence are often overlooked not only by conventional AI researchers, but by roboticists as well.

4.4.3 The Evaluation of Edmund

The two sets of experiments described in this chapter both provide evidence for the dissertation's main thesis, that sequential and hierarchical control structures do not necessarily lead to rigid, brittle behaviour, unable to react to dangers and opportunities in the environment. However, they exploit very different aspects of the Edmund architecture to meet their respective challenges. The Nomad robot experiments demonstrate the power of behaviours given the capacity of variable memory, and of compromises to the modularity of those behaviours by allowing them to be influenced by each other as well as the external world. The arbitration between goals, however, was severely constrained by the simplicity of the tasks for which adequate sensing and actuation on the robot were available, and by the amount of programming time that could be provided. Essentially, they perform no more complex arbitration than the standard subsumption-architecture robots (Brooks 1990). The Nomad experiments demonstrate only that a robot controlled using the Edmund architecture is not so severely hampered by hierarchical control that it behaves significantly below the behaviour-based standard.

The experiments in Tyrrell's simulated environment, on the other hand, demonstrate an ability to manipulate far more goals and avoid far more dangers simultaneously than any real robot has exhibited, despite operating under very poor perceptual conditions. It also allowed for the direct comparison of Edmund's results with that of several other architectures, providing for the use of statistical verification of Edmund's utility. Further, the simulated research, including familiarisation with the platform, construction of the specific control hierarchy, and tens of thousands of experimental trials, took less than one sixth the time that the robot research took, which had far less satisfactory experimental analysis. However, because a model of perception and navigation were provided as part of the simulated environment, Tyrrell's task required no memory that persisted longer than a single decision cycle of the simulation. Consequently, only one behaviour (for selecting direction) was developed beyond the behaviours provided as part of the simulation. Consequently, the Tyrrell simulation did not provide a test of the first two of Edmund's hypotheses (Section 3.4.2).

Even taken together, neither of these domains fully tested all of Edmund's capabilities. For example, Edmund's hierarchical mechanism is capable of supporting recursion: a competence may have an element which refers to that same competence. This ability was not really needed in either of these domains, but almost certainly would be if Edmund were being used to produce natural language for a conversational agent. Also, neither situation required complex action patterns for either perception or motion. Performing vision tasks using a visual routine processor (Ullman 1984, Horswill 1995) or controlling a jointed manipulator such as a hand or a leg (either on a robot or a simulation) probably would. None of these tasks address the issue of whether Edmund-like behaviours and action-selection are sufficiently powerful to simulate conventionally human / cognitive tasks such as following a conversation or displaying personality. These problems would probably best be approached in creating characters for virtual reality.

4.4.4 Summary and Recommendations

In summary, autonomous mobile robots have not proven the ultimate research platform many researchers would like them to be (Brooks et al. 1998, Kortenkamp et al. 1998). However, neither has any other single research platform or strategy (Hanks et al. 1993). While not denying that intelligence is often highly situated and specialised (Gallistel et al. 1991, Horswill 1993*b*), to make a general claim such as the thesis of this dissertation requires a wide diversity of tasks. Preference in platforms should be given to those on which multiple competing hypotheses can be tested and evaluated, whether by qualitative judgements such as the preference of a large number of users, or by discrete quantifiable goals to be met, such as Tyrrell's fitness function, or the score of a Robocup football game.

Robots are still a highly desirable research platform, for many of the reasons stated above. They do provide complete systems, requiring the integration of many forms of intelligence. The problems they need to solve are indeed more like animal's problems, such as perception and navigation. These attributes are witnessed by the need for building perceptual and memory systems for the Nomad research above, but not for the simulation. In virtual reality, perfect perception is normally provided, but motion often has added complication over that in the real world. Depending on the quality of the individual virtual reality platform, an agent may have to deliberately not pass through other objects or to intentionally behave as if it were affected by gravity or air resistance.

Robots being embodied in the real world are still probably the best way to enforce certain forms of honesty on a researcher — a mistake cannot be recovered from if it damages the robot, an action once executed cannot be revoked. Though this is also true of Tyrrell's simulation, particularly in the case of younger students, these constraints are better brought home on a robot, as it becomes more apparent why one can't 'cheat.' Finally, building intelligent robots is a valid end in itself. Commercial intelligent robots are beginning to prove very useful in care-taking and entertainment, and may soon prove useful in areas such as construction and agriculture. In the meantime are highly useful in the laboratory for stirring interest and enthusiasm in students, the press and funding agencies.

4.5 Conclusions

The first step towards assessing the success of an architecture intended to support the development of complex behaviour should be a definition of complex behaviour. Unfortunately, no universally accepted definition of the term exists, although this chapter has discussed several criteria along these lines. Similarly, no single benchmark task could be sufficient for a thorough evaluation of a system, nor is any one task likely to embody all that is "hard" about intelligence. Even the Turing test omits the tasks humans find difficult, such as chess or mathematics. AI has come under considerable criticism for making general claims about systems designed explicitly for a particular task or simulation (Brooks 1991*a*, Hanks et al. 1993).

This chapter has demonstrated Edmund both on a mobile robot and in a simulated environment. No serious revision was made to Edmund for the latter task. The only fixes made were to a slow memory leak that had not manifested itself over the shorter "lifetime" of the robot experiments, and errors in the script parser. These two sets of experiments are the evidence promised in this dissertation that sequential and hierarchical control structures do not necessarily lead to rigid, brittle or unreactive agents. Further, the relative success of the Edmund animal vs. animals controlled by other dominant architectures in Tyrrell's simulated environment we take as evidence that such organisations are critical to organising behaviour efficiently.

Nevertheless, substantially more work remains to be done for creating and demonstrating animal level behavioral complexity. This chapter has reviewed several alternative domains for testing. For AI to make a substantial contribution to psychology, however, it must not only produce interesting behaviour, but use psychological research as the ultimate test of its varying hypotheses. The next chapter turns to psychological evidence supporting the Edmund hypotheses. It then concludes with proposals for refined hypotheses, for further work on improved or different architectures for examining the next set of questions.

Chapter 5

Analysis: Action Selection in Nature

5.1 Introduction

The previous chapter, Chapter 4, has described experimental evidence that natural, appropriate, alert behaviour can be exhibited by agents executing actions pre-stored as hierarchies and sequences. This point is necessary to the thesis of this dissertation: that hierarchical and sequential behaviour organisation is an integral part of intelligence. This thesis was presented and examined at length in the first three chapters of this dissertation. Chapter 3 also presented Edmund, the architecture used for the experimental work, and a body of research from artificial intelligence giving evidence for the thesis in the form of research trends and successes. This chapter begins by presenting psychological evidence for both the hypotheses embodied in Edmund, and the overall thesis. This examination leads to a discussion of the limits of both Edmund as a model, and of the evidence in this dissertation. These limits in turn lead to suggestions for future work, both in terms of psychological experiments and of AI research platforms. The dissertation results and suggestions are summarised in the conclusion at the end of this chapter.

5.2 Psychological Evidence for Edmund's Hypotheses

Section 3.4 introduced the Edmund architecture, and contrasted it with the number of other agent architectures. These contrasts were described and discussed as hypotheses on the nature of intelligent action selection. Chapter 4 demonstrated the utility of these features for constructed intelligent systems in two separate test beds. This section reviews the hypotheses associated with Edmund, this time examining them from a psychological perspective.

5.2.1 Perception requires memory.

There can be no question that a variety of types of memory are essential to the perceptual processes of at least reasonably advanced organisms. The simplest demonstrations of this position are the many examples of critical learning periods required for an animal's perception to function correctly or at all. Examples include:

- the experiments of Hubel and Wiesel (Hubel 1988) demonstrating the deficits in kittens' vision after being either blindfolded or exposed to a deficient visual environment (e.g. only vertical lines) through a critical period of their infancy,
- the experiments of (Gallistel et al. 1991, pp. 13–16) demonstrating that barn owl chicks forced to view the world through refracting prisms while in the nest systematically misjudge the location of sound sources for the rest of their lives, and
- the fact human infants lose the capacity to discriminate phonemes that do not occur in their native languages (Jusczyk 1985).

These are examples of the impact of long-term memory on perception. There are also ample examples of short-term memory affecting perception, such as semantic priming (Neely 1991) and the McGurk effect (McGurk & MacDonald 1976). Much sensory information is ambiguous, or least too difficult to process in a reasonable amount of time, without expectations set by previous context. This finding has been used to explain the massive efferent, "top-down" connections, significantly outnumbering "bottom-up" afferents, even between the cortex and the sensory organs (Sillito et al. 1994, Nicolelis et al. 1995, Rao & Ballard 1996).

This is not to say that all intelligent action triggered by the environment necessarily requires advanced memory or processing. In terms of AI, purely reactive, learning-free sorts of behaviour are appropriate for certain sorts of tasks, generally characterised by stereotypical responses to a continuous signal. For example, a lawn-mowing robot may brainlessly search for long grass and mow it; so long as it moves quickly enough to randomly cover its territory faster than the grass can grow. Long grass can reliably trigger such a robot to engage its blades.

In general, it is indeed better to rely on information in the environment rather than information in memory, providing that the environmental information can be gathered sufficiently reliably and efficiently. Animals, including humans, will preferentially use the environment to indicate the next phase of activity, even when they have the resources and opportunity to use memory instead. Humans given the task of copying a colourful abstract design using small plastic pieces will rely on the memory of no more than one of the colour or the location of the piece at a time if they have the model continuously available to them. This has been shown by using eye tracking devices, and by changing the pattern to be matched, which subjects fail to notice (Ballard et al. 1997). This is despite the fact that humans have the cognitive capacity to copy the entire pattern from memory if the original is not made available to them during the construction.

Even insect-level intelligence, originally considered the necessary first target for proving theories of dynamic intelligence (Brooks 1991*b*), has examples of both memoryless and memory-based behaviour control strategies. Mound-building termites are often used as an example of dynamic or emergent intelligence. If termites are placed in an open, dirt filled tray, they will only gather dirt into piles rather than constructing a mound. However, once there are enough termites and hence a sufficient density of piles, the termites switch behaviour into constructing arches, the fundamental component of a mound. On the other hand, digger wasps have been shown to be able to remember the state of progress of multiple nest sites in different stages of completeness, without apparent visual cues, and honey bees perform dances to communicate their recollection of recently visited food to the other members of their hive.

Lesion experiments indicate that many behaviours that may be categorised as simple acts actually exhibit combinations of memoryless and memory-based strategies for control. Frogs are able to accurately scratch the location of skin irritations with their back leg even after surgical removal of their brain (Bizzi et al. 1995, Mussa-Ivaldi et al. 1994). However, the scratching of these spinal frogs and intact frogs is significantly different. Intact frogs are able to also scratch one leg with the other, no matter where the itching leg is situated, whereas spinal frogs can only scratch a leg that is in a stereotypical pose. This behaviour is similar to that of the intact locust, which, if stimulated on the wing case will scratch the location on its back where the stimulus would be if the wing case were closed, regardless of whether the case is there or has been extended (as in flight) (Matheson 1997). Reflexive behaviours are useful because they are reliable, but extra processing ability allows for the addition of more complex computations and therefore of more flexible and elegant behaviour.

In conclusion, although there is evidence that nature prefers reflexive behaviour for some situations due to its reliability, it abounds with examples of memory and learning. Even minimal processing requires some transient state, and probably some amount of training or calibration. For example, could a spinal juvenile frog scratch itself accurately after it had matured? Two of the main tasks for an adaptive intelligent agent are learning to categorise and learning to discriminate. Whether these skills are attributed to perceptual learning or to cognitive learning is irrelevant from the perspective of action selection. Within Edmund, the distinction is unimportant: all learning takes place within the behaviours, though those behaviours may be perceiving either external or internal state changes.

5.2.2 Memory is modular but not strictly encapsulated.

Similarly, there is no shortage of examples of the interconnectivity between essentially modular sections of the brain — where modularity is defined by differential organisation or composition relative to neighbouring parts of the brain, or by an obvious dominant functionality, as in for example the visual or auditory cortices. Many of the best understood "modules" of the brain are parts of the system for processing visual information. For example, the first processing layer of the retina is understood to be recognising colour and intensity of light, the next to be recognising edges between hues and intensities (Carlson 2000). The earliest layers of the visual cortex are then considered responsible for detecting edges and their orientations in the image, and so on. Regions of the brain have even been identified for identifying both individual's faces, and the orientation of generic faces (Perrett et al. 1987, 1992).

A supporter of the reactive approach to intelligence might argue that the visual system *could* be described as reactive if considered at the proper level of abstraction: the individual cell level. Each cell taken individually only responds to its input, they might argue — the labelling of visual system components as modules is incorrect. This argument is wrong for several reasons. First, the "modules" are not so called simply because the individual cells perform similar functions. The cells do not behave coherently as individuals — each cell's output is only meaningful when compared with that of its neighbours. The cells inhibit and excite neighbouring cells within their own layer, as well as transmitting information to further layers.

The second reason the visual system should be thought of as state-sharing modules is that each individual module (with the possible exception of the retinae, though see (Smirnakis et al. 1997)) *learns*. This learning is not only the transient passing and processing of state during visual sensing, but also the long-term organisation and classification of such stimuli (e.g. Hubel 1988, Gallistel et al. 1991, as above). This learning is again only meaningful on a modular level, not an individual cell level. For example, very early ontogenetic learning is concerned with simply organising cortical cells to be topological maps of the retinal input (Sirosh & Miikkulainen 1996).

The third reason that a simple reactive model cannot be applied to the visual system is the multidirectional interaction between the internal, learned state of the visual systems' various components. This is evidenced not only by their interconnectivity (MacKay 1956), already mentioned above, but also by experimental evidence from machine learning and cognitive science (e.g. Barlow 1994, Rao & Ballard 1996, Hinton & Ghahramani 1997) as well as psychology. Psychological experiments have shown also thorough functional interconnectedness between different sense modalities for even perceptual experiences that have dedicated single sensory systems. For example, vision has been shown to contribute to both the human vestibular perception (Lee & Aronson 1974, Lee & Lishman 1975), and the perception of phonemes (Jusczyk 1985, MacDonald & McGurk 1978, McGurk & MacDonald 1976).

Although the above arguments concentrate on the area of perception, this should not be taken to support the model of Fodor (1983). Fodor's influential but now largely discredited theory on modularity not only argues for full encapsulation between modules, but also that they serve merely to translate perceptual information into a uniform representation to be processed by a "central system" (see Hendriks-Jansen 1996, Elman et al. 1996, for a discussion). In Edmund, there is no such centralisation. Cognition is assumed to be distributed, perception and action are considered to be *parts* of cognition, perhaps the only parts. This is in keeping with the evidence of Ballard et al. (1997) and the philosophy of Clark (1996) and Dennett (1995, 1991), as well as the spirit of behaviour based AI. It also relates fairly well to the popular recent theories of emergent modularity in psychology (Karmiloff-Smith 1992, Elman et al. 1996, Bates in press). Unfortunately, Edmund can only serve as a snapshot of a system such as would be developed under emergent modularity, because its modules are designed by its programmer. This issue is addressed below in Section 5.3.4.

5.2.3 Action selection is dependent on structured control.

As was made clear in the first chapter, the final hypothesis of Edmund is controversial in psychology as well as artificial intelligence. That natural intelligence might use structured control is a theory — the theory this dissertation is intended to help defend. The previous chapters have demonstrated the viability of such architectures in controlling reactive, opportunistic behaviour in dynamic environments, in contradiction to the claims of some critics of hierarchical control. This section continues such support by reviewing evidence that natural action selection does in fact use these structures, at least on some occasions.

Neurological Evidence for Explicit Sequencing

There is fairly clear neurological evidence of explicit rather than dynamic sequencing of behaviour. For example, a region of the mammalian brain, the preaquaductal grey matter, has been identified through lesion and activation studies as storing species typical behaviours such as mating rituals (see for review Carlson 2000). Such behaviours have traditionally been described in terms of fixed action patterns, but even if their representation is in fact more dynamic or flexible than that label implies, the very fact they are separately stored and activated through a specific region is an example of hierarchical control. These species-typical behaviours are subroutines which can be repeatedly accessed in appropriate contexts.

Evidence of brain cells directly involved in processing sequential recall and behaviour has been found by Tanji (Tanji 1996, Tanji & Shima 1994). In this set of experiments, Tanji & Shima trained monkeys (*macaca fuscata*) to perform several sequences of tasks, both on touch pads and on cranks which might be rotated, pushed and pulled. He trained his subjects both to do the appropriate task as cued by coloured lights that corresponded to specific task elements, and by memorising a sequence of elements. Tanji & Shima found cells firing in the medial frontal cortices, (the supplementary motor area) in three different functional contexts. A few cells (12%) functioned like those in the primary motor cortex, corresponding directly to the particular motion or task, such as turning a crank, as it is occurring. These were used for either the recalled or the cued sequences. However, 36% of the cells fired only for a particular transition between tasks, such as moving from turning a crank to pushing it (see Figure 5.1). These cells were used in multiple sequences if the identical ordered pairing occurred, but not if the two components were separated or in the reverse order. That is, if the monkey knew more than one sequence with the same transition (from turn to push), the same cell seemed to represent the same transition. Finally, 26% of the cells were active only preceding a particular complete sequence, during the reaction time between the signal to start the sequence and the monkey's initiation (see Figure 5.1). Both of these two classes of cells were active only when the monkey was recalling the sequence from memory.

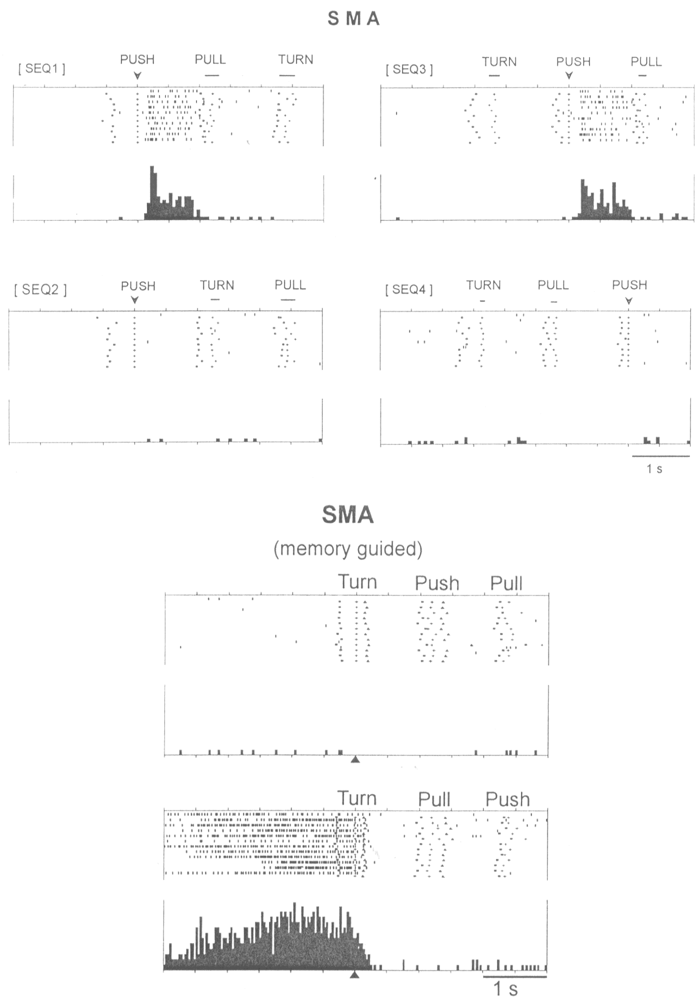


Figure 5.1: Two different neurons from the medial temporal cortex of *macaca fuscata* implicated in controlling learned sequential behaviour. The first fires in preparation for particular transition, regardless of the identity of the overall sequence; the second only before a particular sequence.From (Tanji & Shima 1994, p. 414)

Behavioural Evidence of Hierarchical Control

Most versions of hierarchical control, including Edmund, reduce the combinatorics of the problem of action selection by selectively focusing attention on only a few appropriate behaviours. While this may seem rational from a computational viewpoint, it could also lead to a failure of opportunism which could be inefficient and even dangerous. This is the criticism of Maes (1991) and Hendriks-Jansen (1996), and explains why the linear hierarchy of subsumption architecture (Brooks 1986, section 3.3 of this dissertation) and the free-flow hierarchies of the Extended Rosenblatt & Payton architecture (Tyrrell 1993, section 4.3.3 of this dissertation) are fully parallelised. Edmund, seeking the advantages of both systems, is only partially parallel. If this is a valid psychological model, then there should be evidence both of neglect in some instances of complex control, while also evidence for parallel monitoring for important conditions.

There is, of course, ample evidence that animals may be interrupted from their activities by external events. Evidence of the monopolising of behaviour or intelligence by a particular cognitive context is more difficult to come by, as behaviour continuity could be described as a reasonable, rational strategy. In other words, it is difficult to determine whether an animal is persisting at a particular behaviour because it currently has no option, or because that persistence is an optimal choice. However, there are some examples where behaviour is exposed as irrationally persistent in unusual circumstances. For example, when a digger wasp is creating a den in which to lay an egg, one of the most time consuming elements of the process is to hunt, kill and retrieve a caterpillar to serve as food for the hatched larvae. However, the wasp will only recognise a dead caterpillar provided locally by an experimenter if she is in the phase of her activity when a caterpillar is required. While she is digging a nest or plugging a nest, the caterpillar in the region is treated as if it were dirt. A caterpillar may be used to plug the entrance of one nest even when the very next activity of the wasp is to hunt a caterpillar for a second nest being constructed in parallel (Lorenz 1973). Another ethological example of behaviour persistent beyond the rational is that terns have been shown to make the gesture of rolling a misplaced egg all the way back to their nests even when the egg is surreptitiously removed (via a trap door arrangement) midway on their path.

Rats have been shown not to engage in any appetitive behaviour (such as foraging or eating) without the appropriate triggering of their limbic system (Carlson 2000). This means that an equally hungry rat (on some absolute caloric level) may or may not eat food present in front of it if it is engaged in some other activity. Starving Siamese fighting fish will still patrol their territory at regular intervals even when finally confronted with food (McFarland 1987, Hallam et al. 1995). Their patrol timing is perturbed by the high priority of eating, but not eliminated. Further examples of hierarchical and modular behavior are reviewed by Gallistel (1980, 1990).

Pöppel (1994) argues that the phenomena of ambiguous visual illusion, (e.g. Figure 5.2) show evidence of a human incapacity to entertain two contexts simultaneously. This has been explained as a constraint of the visual system, that only one high level interpretation can be entertained by the system at a time. Such a constraint would be an example of modularity constrained not by locality within the brain, but by the the functioning of the cognitive system. The activation levels forming the neural representation of a particular pattern are incompatable, despite sharing the exact perceptual information and presumably the bottom-up activation of many lower level visual processes.

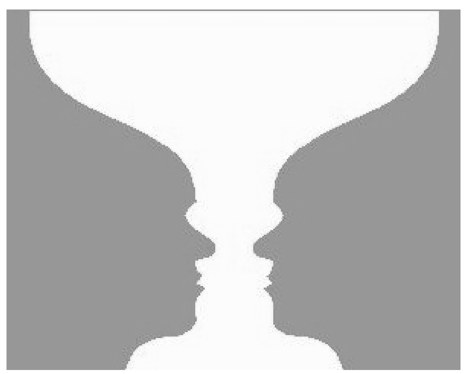


Figure 5.2: An ambiguous image. This figure can be seen either as a vase or as two faces, but not both same time. From (Gleitman 1995, p. 213)

Neurological Evidence of Hierarchical Control

The most promising area of research in examining mental context is the very recent advance in neuroscience of multi-cell recordings of various brain regions, most typically the hippocampus (Wilson & McNaughton 1994, Wiener 1996). The hippocampus is a particularly interesting brain region because it has extremely sparse representations — that is, relatively few cells are firing in any particular context. This is believed to be an attribute of its function, where it seems to be involved in the fast memorisation of episodic events (Teyler & Discenna 1986, McClelland et al. 1995, Levy 1996). In order to record a great deal of information (many events) without the various memories interfering with each other, the encoding of the memories must have a relatively small number of cells involved in their recording, which in turn means that each cell must be highly significant. In rats at least, this significance has been sufficiently simply encoded that the awareness of many items of importance to a rat can be identified from tracing the firing of a hundred or so cells. These salient features include the rat's location in a cage, its orientation, what its current task is, and what phase of activity within a task it is in (e.g. during a pause after a light has come on signalling that a test is about to begin.)

The evidence from this work that bears most strongly on hierarchical plan representation is that there is evidence that the ordering of the encoding for these various salient features is context specific: the coding is persistent for weeks given a particular context, but shifts instantly when the rat switches between contexts. For example, if a rat runs to a different room, it may have a completely new mapping of even the persistent features external to the two rooms, such as how its head is oriented. Observers can thus tell from the recordings of the cells what context the rat *thinks* its in, for example, whether it can be fooled into mistaking one room for another. Interestingly, some rats may construct nearly identical representations for two rooms that other rats distinguish clearly between (Kobayashi et al. 1997).

This research is only in its infancy: multi-cell recording is a difficult technique so far existing in only a few laboratories, and tends to be run on only a very small number of rats. The research is being applied primarily to trying to decipher the purpose and mechanisms of the hippocampus and similar brain organs. However, since the hippocampal representations have already been shown to reflect events of internal significance, such as the current behaviour, as much as external features, such as location in the room, there is no reason to suppose they do not represent a significant amount of whatever might be called a rat's "plan." If this is the case, it may be possible to see the representations shift as the rat learns a task, or achieves a generalisation or insight in its learning. Presumably, insight of this sort is identical to recognition of context, which has already been witnessed. If recognisable features of plans were ever spotted in analogous rather than identical situations, then hierarchy could be proven. And this has indeed happened already, if one accepts that the same task on different days *is* only analogous, it is not identical. In this case, we already have strong evidence of hierarchical representations of some significant part of a rat's mental life.

Behavioural Evidence of Learned Hierarchies in Primates

Finally, there is also evidence that at least some non-linguistic animals can learn and use hierarchical organisations for their explicitly controlled behaviour. In the University of Edinburgh Laboratory for Cognitive Neuroscience and Intelligent Systems there has been an ongoing research program into the ability of non-human primates to learn to manipulate objects in their environment in ways that require structured organisation (see for a history McGonigle & Chalmers 1998). McGonigle & Chalmers (1977) show that squirrel monkeys are capable of performing transitivity tasks. That is, on learning a set of ordered pairs such as $A > B, B > C, C > D, D > E, E > F$, the monkeys were able to generalise this knowledge to know that $B > E$. This works even when the metric for ordination was chosen arbitrarily and was controlled between animals, so for example the metric might be colour, and different animals might have to learn different orderings.

These findings led to the current research program in the laboratory which has given evidence that Capuchin monkeys are capable of learning to exploit hierarchical organisation to extend their ability to learn sequences of twelve items (McGonigle & Chalmers 1998, Dickinson 1998). The stimuli for these monkeys are variously shaped, sized and coloured items on touch screens, which must each be visited in order. The required sequences involve visiting all items of a particular shape or colour contiguously, so for example, first visiting all squares, then all circles. The subjects show clear boundary effects when switching between element types — that is, they take significantly longer choosing the first square or first circle than choosing the second or third of a category. This reaction time result indicates even more conclusively than the increased sequence length (which might be some form of practice effect) that the monkeys are viewing their problem space hierarchically. Moving between classes is fundamentally different than moving between members of the same class.

Of course, the fact monkeys can use structured control in deliberate activity does not necessarily indicate that all of their action selection is controlled in a similar way. However, it does demonstrate that such control strategies can exist in not only humans, but also other primates without interfering with their native intelligence. In fact, such capabilities are well associated with advanced intelligence, having also been demonstrated extensively in chimpanzees and humans (Greenfield 1991, see for discussion).

Conclusion

It is difficult to demonstrate conclusively the strong claim that action selection is *dependent* on sequential and hierarchical control structures. This requires not only showing such strategies may be used and are used for action selection, but that no other strategy is possible. That no other strategy is possible has in fact been argued persuasively on several occasions (c.f. Dawkins 1976, McFarland 1985, McGonigle & Chalmers 1996, 1998). The argument is straight-forward — that no other strategy can handle the combinatorial complexity of the problem of action selection, nor explain the regularity of animal behaviour (see further Section 2.2.6 above). However, this argument is still in dispute by some who claim our understanding of dynamic, chaotic processes and parallel algorithms is still too immature to rule out other forms of organisation and emergent regularities (e.g. Goldfield 1995, see review in Section 1.2.2). Although such arguments based on ignorance are necessarily inconclusive and therefore deeply unsatisfying, they are also very difficult to disprove. However, even if the rationality and regularity of action selection is emergent from complex dynamic processes, there would still be a question as to whether this really precludes a hierarchical model of these processes. From a functional perspective, the order that emerges from the (possibly dynamic) underlying processes seems to be well described hierarchically, and therefore hierarchical artificial intelligence may be a completely appropriate model of the natural process.

5.3 Further Psychological Evidence, and Limitations of Edmund

Edmund is not the central thesis of this dissertation; rather it is the model through which a part of the thesis has been demonstrated. This evidence that sequential and hierarchical control structures do not necessarily lead to overly brittle or insufficiently reactive behaviour can be used in support of the thesis that such structures underlie some naturally occurring intelligent behaviour. The previous section discussed evidence supporting this thesis from within the Edmund context; this section discusses other research related to the thesis. As such, some of it also touches on shortcomings of Edmund, which are summarised and addressed.

5.3.1 Neurological Mechanisms for Action Selection

One basic assumption of Edmund is that there is a neural mechanism by which ordered action selection can be achieved. This assumption is sufficiently common in the AI literature that it wasn't listed as an Edmund hypothesis, but it is none-the-less controversial and one of the central arguments of Brooks (1991*a*) and Hendriks-Jansen (1996). As mentioned in Chapter 3, the primary argument against such structures is homuncular — what can be doing the selecting? This argument has been seriously undermined recently by Gurney et al. (1998), who have demonstrated a neurologically plausible algorithm for action selection which could be learned in an unsuprevised manner. They have proposed their mechanism as a hypothesis for a function of the basal ganglia (Redgrave et al. 1999), and have outlined the implications for this research for artificial intelligence Prescott et al. (1999). Their argument, whether accurate in detail, is solid in general: the problem of action selection between competing systems can be learned by a relatively simple system that has no model of the actual complexity of the actions to be undertaken, but knows only the relative importance of various systems and the strength of the present signals. Such a mechanism could very well be used for selecting between the competing parallel elements of drives within Edmund. Activation propagation within a competence or action pattern is more likely explained by some form of spreading activation and thresholding, as modelled by Glasspool (1995) or Wennekers & Palm (to appear).

5.3.2 The Combined Expression of Multiple Behaviours

Another criticism levelled against hierarchical control such as Edmund's is that it precludes the combination of various behaviours' desires into one fused behaviour. Tyrrell (1993) discusses this at some length, but actually confounds two different forms of combination. The first, *compromise candidates*, as defined by Tyrrell are when the desires of several different behaviours with similar consequences are combined in the selection of the final action. Tyrrell's example is the selection of a direction to run that is both away from a predator *and* towards shelter, things that are recognised by two different elements in his system. A further contribution to this approach was made by Humphrys (1997), who found he could improve the performance of a society of problem-solving agents by allowing them to perform simple negotiation for optimal solutions. In fact, it is only through this mechanism that he was able to achieve better performance with a learned multi-agent system than a monolithic one. However, these sorts of approaches can only be relevant when there is a clear mapping between the possible behavioural outcomes of the various agencies. This will probably occur only in relatively few situations that truly bridge different behaviours. In Edmund, such occurrences may be encoded either by re-decomposing the problem so that the outcomes that share state are part of a single behaviour, or by allowing for an encapsulation violation between two behaviours.

The second form of behaviour combinations is *fused behaviour expression*. This is when two behaviours with distinct action attributes have impact simultaneously in the agent's expressed behaviour. This is a much better description than "compromise candidate" of what happens in the continuum of facial expressions which Tyrrell discusses (see Figure 3.1). There are two ways to address this problem that have been demonstrated in the literature, neither of which require either compromise or encapsulation violations. Thórisson (1999) demonstrates in Ymir a mechanism for emergent combination of behaviours. Ymir is an architecture for multi-media conversational agents. The agents created under Ymir have, among other things, facial expressions and hand gestures. Thórisson solves the fusion problem by having a separate, cerebellum-like element of the architecture select the actual expression of particular behaviour based on the current position of the agent's features. Thus, if behaviour 1 sets a particular expression on an agent's face, say a smile, then when behaviour 2 determines to acknowledge a comment by another speaker, it might choose to nod, or to widen the grin. A simpler mechanism is provided by Blumberg (1996), where behaviours are organised by the resources they are going to control on an animated agent, where resources are enumerated by degrees of freedom[1]. Behaviours that "win" in action selection but don't necessarily need to control all the degrees of freedom of the agent thus allow for the expression of other behaviours. Consequently, an animated dog might have one behaviour causing it to walk, and another affecting its facial expression or tail wagging.

In order to exploit Blumberg's strategy under Edmund, one would need to have the fused behaviours on different drives, running in parallel. This is actually fairly plausible, not only from an engineering but also a biological standpoint, since most of the examples of behaviours that need fusing in the literature discuss the fusion of functional behaviours with emotional or expressive behaviours. From an engineering perspective, there have been many architectures

[1] A *degree of freedom* in this context is any thing that could move, in a robot it would be represented by a single motor. The tail of a dog, for example, might have two degrees of freedom – it might wag from side to side, but also raise or lower to wag high or low.

proposing independent hormonal or emotional systems for affecting behaviour or learning (see for a recent review Gadanho 1999, pp. 50–54), and from a biological perspective the limbic system has long been considered a parallel brain mechanism for determining mood and emotion. Exploiting Thórisson's system is even simpler, because it involves only having Edmund's action selection passed into an Ymir-like action processor rather than being directly expressed in the agent. This is an extremely viable extension of the work in this thesis, and in fact is already underway (Bryson 1999).

5.3.3 Explanations for the Misordering of Actions

One simple objection to the theory that actions are ordered, in fact, one that has inspired many of the theory's detractors, is the simple observation that behaviour sequencing sometimes fails. Even well-known and frequently practised sequences may abort or become disordered. Fortunately, the psychological literature already has mature mechanistic models for how sequences are performed which account for these problems (see for reviews Houghton & Hartley 1995, Henson 1996).

There is no generally accepted theory of the exact mechanism for sequential control. Indeed, there may be multiple mechanisms for controlling sequencing at various rates and with various levels of deliberate control. Henson & Burgess (1997) is a typical example of the sorts of models that best fit the psychological data for sequencing errors. It involves a single-neuron representation of each stage of the sequence, with all stages interconnected by inhibitory connections. By some mechanism, the cells are ordered so that the first cell is fired, then self inhibits, resulting in the second cell becoming the self-expressing, other-inhibiting cell, and so on. In the Henson & Burgess model, firing priority is determined by comparing associated values indicating relative positioning within the sequence. A slightly older model, *competitive queueing* Glasspool (1995), has a simpler mechanism of higher initial activation being given to the earlier members of the sequence. However, this simpler model fails to adequately explain a particular class of errors where an element from one sequence replaces an element in the same ordinal location in another sequence — an error that indicates some form of homogeneity between sequences.

Norman & Shallice (1986) present a higher-level model of action selection which has several interesting parallels with Edmund. Norman & Shallice separate control into two different systems: one for routine behaviour control, called *contention scheduling*, and a second for the non-routine, called the *Supervisory Attentional System*. The latter system is an attempt to formalise the role of conscious attention, but its model is not well fleshed out. Contention scheduling, on the other hand, has been formally and computationally modelled. As described by Cooper et al. (1995),

> The central component of the CS theory is the schema network. In formal terms, this is a directed acyclic graph in which nodes correspond either to action schemas (partially order sequences of actions) or goals. Edges from the schema nodes point to goal nodes, and vice versa, so the network may be thought of as consisting of alternate "layers" of schema and goal nodes. The schema nodes pointed to by a goal node correspond to distinct methods of achieving that goal, and the goal nodes pointed to by a schema node represent the subgoals which must be achieved in order to execute the schema. Thus, the goal of preparing a mug of coffee might be achieved in a variety of ways (e.g., either by preparing instant coffee or by preparing percolated coffee), and the schema for preparing instant coffee will include a number of subgoals (e.g., boil water, add coffee grinds, add cream, etc.).

At any point in time, each node of the network has some activation level between 0 and 1. A node requires not only a high activation but also that triggering conditions are met before it can operate. Nodes are also affected by lateral inhibition between schemas that share the same goal, to prevent more than one schema from firing at a time, and between schemas that share the same element, to prevent an element's activation from exciting inappropriate schemas. In order to ensure the system is constantly active, excited nodes also self-activate to maintain their aroused state.

The Norman & Shallice model shares with Edmund some of the characteristics of a distributed hierarchical system that allows interrupts from parallel systems. However, it is radically different in that it is designed explicitly as a psychological model, rather than an AI architecture, and thus gives less consideration to the design of a complete system, but more to modelling psychological realities. Cooper et al. (1995) uses a model of contention scheduling to model the sequencing errors made by psychiatric patients. For example, the researchers found that by changing the relationship between the strength of self activation and lateral inhibition, they were able to simulate the behaviour of Parkison's disease patients. This sort of research is impossible on Edmund, because sequences are encoded in a much simpler representation which is guaranteed to fire in order.

Extending Edmund to have more psychologically plausible sequencing behaviour like competitive queueing or contention scheduling is in principle possible, but as a design decision would have to be carefully weighed. There might be an advantage in having potentially more flexible behaviour, particularly if Edmund were to become a learning architecture where serendipitous "mistakes" that lead to more effective behaviour might be recognised and remembered. On the other hand, such flexibility might equally well be provided by the mechanisms of Thórisson at a lower cost. Spreading activation networks are very expensive in terms of both design and computational overhead, and are also notoriously difficult to control in large-scale projects (see the previous discussion of Maes' network in section 3.5.1, page 48). Edmund gives a fairly optimal level of abstraction for the research detailed in the previous chapter, and Contention Scheduling for the research detailed by Cooper et al. (1995). What might be more useful further work is a careful analysis of the two architectures that leads to a mapping between their behaviours, so that they can be fully understood in terms of each other. This might lead to further insights useful for extending one or both architectures, or establish definite differences between them which could serve as hypotheses for further research.

It is worth mentioning that the simplest form of generating sequential behaviour, *chaining*, though still studied by some members of the machine learning community, is generally discredited among psychologists and neuroscientists. Chaining, favoured by the behaviourists, assumes that behaviours are stimulus-response pairs, and a sequence is a chain of such pairs, each element of which is triggered by its predecessor. Lashley (1951) points out that such an arrangement is physically impossible because of timing issues — many sequences are performed more rapidly than the response time for such a circuit would require. Further, the errors such a system would predict, such as confusion between different elements following a repeated element of a sequence, are not found in human experimental data (Houghton & Hartley 1995, Henson 1996).

5.3.4 Learning

The fact that perception requires memory indicates that perception requires learning — at the most basic level of abstraction, learning is the process by which memory is changed. Edmund, then, provides for learning so long as it occurs within a particular behaviour. Assuming an Edmund behaviour is reasonably specialised in advance, this means that the only learning afforded in Edmund is specialised learning, that is, learning that can be anticipated, and for which accommodation can be made. This is not as limited as it might sound: by some accounts, specialised learning accounts for the majority of learning in animals (Roper 1983, Gallistel et al. 1991, Bryson 1996*a*). As has been shown in Chapter 4, such specialised "learning" includes a model of intelligent perception. Ambiguities can be resolved by comparison to past readings, and simple filters can throw out impossible interpretations of the sensor information. In other words, Edmund allows for short-term perceptual memory. Edmund could also potentially be used for other sorts of memories, provided their representation can be determined in advance. Thus it might be used to model episodic memory, imprinting and other forms of perceptual or muscle skill tuning that happen over the lifetime of an agent.

Nevertheless, there are definite limits to Edmund's ability to model animal learning — it cannot develop new behaviours or representations, nor can it develop new control patterns. Both of these problems will now be considered in turn.

Learning New Behaviours

In order for any system, including Edmund, to allow for the learning of new behaviours, the behaviours themselves would have to be represented in a variable space. The extent to which this is true for animals is under debate, but there is reasonably strong evidence that this is a good description of cortical learning in animals. Even well established modularity in mammalian brains is able to be violated in extreme cases, such as losing part of the brain or part of the body. Lashley (1950), showed that a monkey could recover from the removal of an entire motor cortex. Lashley took this experiment to disprove any notion of modularity in the cortex. This conclusion neglects the fact that the animal was paralysed on the corresponding side of its body for over eight weeks — a more likely conclusion is that modularity is real, but plastic. Further evidence comes from the description by Ramachandran & Blakeslee (1998) of the phenomenon of phantom limbs. After amputation, many patients continue to perceive feelings and pain in their now absent limb. This is a consequence of neural invasion of the areas of the sensory cortex previously used by the missing limb, by other nearby active regions of the sensory cortex. Again, this is an indication of both plasticity and persistent modularity. The sensations felt in limbs that are no longer present are the result of the old machinery of the cortex interpreting current neural activity in the context of the former physical state. These sorts of recalcitrance are a demonstration that modularity is founded in something substantial. This "something" may either be a consequence of physical location — the presence or absence of connections to relevant information sources, or of well entrenched learned behaviour, which may make the reuse of theoretically plastic connections difficult because of the stability of the current activation patterns. Bates (in press) puts forth the latter hypothesis as an explanation for the critical period in language learning.

Despite this evidence that biological systems develop some of their modularity over a uniform substrate, such an approach for an AI system intended to express complex behaviour is probably impractical. In this respect, the problem of learned behaviours is similar to the problem of variability in sequencing above: it is a biological reality that an ultimate model of intelligence should reflect, but it is beyond the scope of a system such as Edmund. In this case, there are no existing systems that control behaviour to nearly the complexity of Edmund that express this sort of modularity. The nearest model to date is probably that of Plaut (1999). This models modality-specific aphasias by learning lexical maps on an essentially uniform substrate (square matrix) representation. The modularity is a consequence of the fact that different edges of the lexical matrix are adjacent to separate networks for controlling modality-specific actions and receiving various modality inputs. Like the work of Cooper et al. (1995), this work is interesting as a model of a particular psychological phenomenon. However, the modularity produced by this model is nowhere near the level of encapsulation even of Edmund, and no specialised representations are developed.

Some slightly more practical means of learning behaviours has been demonstrated in the behaviour-based AI literature, including Warnett (forthcoming) mentioned above, see Dorigo & Colombetti (1998) and Perkins (1998, pp. 109–118) for recent reviews. These approaches consist primarily of researchers using either neural networks or genetic algorithms to learn either new behaviours composed of established primitives or new action-selection devices for established behaviours. In all cases, the learning is effectively supervised to greater or lessor extent by the provision of fairly specific *reward functions* by which the learning algorithms can evaluate their own success. Thus far, no system has demonstrated learned behaviours surpassing the quality of hand-coded behaviours[2] (Perkins 1998, Matarić & Cliff 1996). The problem facing this work is again one of combinatorics: providing enough bias to make the learning problem tractable is so far more difficult than programming the behaviour by hand.

Learning by Imitation

Of course, nature has conquered this problem, and in time the techniques of AI may also build a sufficiently useful and powerful representation of the learning space that this sort of learning may be tractable. One particularly promising area of learned robot behaviour is learning by imitation (see for a recent overview Dautenhahn & Nehaniv 1999). The advantage of imitation is that an agent can benefit by the experience of conspecifics, thus attacking the combinatoric problem by using combinatorial numbers of learning agents. The power of this sort of strategy can be observed in the rapid advances of human culture.

An example of an AI model of imitation learning is proposed by Demiris & Hayes (1999). Demiris & Hayes demonstrates agents in which one module attempts to recognise the behaviour of another agent by comparing the observed behaviour to behaviours in the observing agent's own repertoire. Simultaneously, another behaviour stores episodic memories of the observed behaviour's components. If the recognition phase fails, the episodic memory is formed into a new behaviour which is added to the behaviour library, otherwise it is forgotten. The hypotheses behind Demiris & Hayes's work are also grounded in neurological research.

Learning New Strategies

Creating and modifying catalogues of behaviour as a strategy for creating AI is actually a major subdiscipline of the field, known as **case based reasoning** (Hammond 1990, Kolodner 1993). Case based reasoning assumes that intelligence consists of learning or being provided with a set of scripts or **cases** which can successfully control an agent's behaviour in a particular situation.

[2] Humphrys (1997) does claim to have learned a superior controller (given provided behaviours) to any one that can be designed. However, his simulation is not yet publicly available, so this claim has not yet been evaluated against the design methodology of Edmund.

To quote Hammond's web page, "At its core, all of my work is derived from a single idea: Reasoning is Remembering." Bringing together the accumulated experience and ideas of the fields of behaviour based AI and case based reasoning would almost certainly be a very fertile area of research. One example of such a combined program has been demonstrated by Ram & Santamaria (1997), who have developed a case representation for the sorts of continuous events present in robot navigation. This work is strikingly similar to that of the behaviour based imitation researchers Michaud & Matarić (1999).

Traditionally, however, case based reasoning has been applied at a higher level of abstraction than the behaviour-based behaviour. Typically, cases operate at the symbolic level. Consequently, case-based reasoning might be more relevant to providing adaptation to Edmund's control hierarchies than to providing a source of new behaviours. The two primary problems of case based reasoning are:

1. *case selection* determining when to apply a particular case, or which particular case to apply to a situation, and

2. *case adaptation* adapting a stored case to the current situation.

These two problems interact: the extent to which a case can be adapted, or to which a system knows how to adapt a case, determines the number and nature of situations to which it can be applied. In Edmund, two things determine when a competence is applied: the current control context, and perceptually based trigger conditions. A competence will only be applied if it is the highest priority relevant component with trigger conditions that are met, and is under one of the drives or competences that has the current attention of the action selection system. Edmund also has a means to modify the current context. These are entirely at the behaviour level, since that is the only variable state. For example, a behaviour might set its internal attention on a particular target, such as a landmark on a map or a block in a pile. The competence would then attempt to navigate to that landmark, or grasp that particular block.

Thus, to some extent, Edmund already implements a version of case-based reasoning. In order for further adaptivity to be present, there would need to be a module capable of manipulating the control scripts, but this would open up a large number of problems. For example, how would the scripts be described in such a way that a most appropriate one might be recognised? And how would they be adapted?

Another potential mechanism for adapting behaviour scripts is genetic algorithms: a script might be modified by cross-over with another script, or by mutation to some elements of the script, then tested for fitness. The best resulting scripts would then serve as material for the next batch of modifications. Such a strategy, applied in real time with some sort of internal criteria of soundness for the evaluation, is similar in some ways to the theory of Calvin (1996), pertaining to thought as evolution. It could perhaps more easily be envisioned running in the Tyrrell simulated environment described in section 4.3.2, which also illustrates the problems with this approach. Unless a modification is very significantly more or less fit than the current best match, it will will require approximately 6000 trial runs to determine its relative fitness. This again is the problem of combinatorial complexity: it is difficult to imagine an automatic system for generating new scripts that would be sufficiently selective as to find a productive increase in fitness in a sufficiently small number of guesses. Nature solves this problem with a very rich system of representation and enormous quantities of trials conducted in parallel. So far, artificial intelligence has been best served by relying on human intelligence for the generation of action selection schemes that are highly likely to be fit.

5.4 Conclusions and Suggestions for Further Work

This chapter has provided a survey of evidence from the life sciences, particularly neuroscience and psychology, for the existence of the sequential and hierarchical structuring of behaviour in animals. This evidence serves, along with the experimental evidence presented in the previous chapter, as further proof by demonstration that having sequential and hierarchical structuring underlying behaviour does not necessarily make that behaviour brittle or unresponsive. Rather, these chapters have shown evidence that such strategies support the mastery and combination of complex behaviours which can in turn provide greater robustness and flexibility in an animal or artifact.

In addition, this chapter has provided a review of the agent architecture Edmund as a psychological hypothesis. It first supported the main claims differentiating Edmund from other AI architectures, then pointed out limitations in the architecture as a psychological research platform. Most of these limits reflect not so much short-comings in the architecture as a specialisation to the particular work for which the model was built. In each case, alternative models or architectures were suggested for exploring particular psychological phenomena outside

Edmund's parametric space. This was by no means a complete review of alternatives. In particular, there was no discussion of the extensive number of packages available for neuron-level simulation. Neither was there any discussion of several of the dominant cognitive modelling architectures, such as SOAR (Newell 1990, Laird & Rosenbloom 1996) or ACT-R (Anderson & Matessa 1998). These architectures were discussed in Chapter 3 as part of the review of AI architectures. They were not reviewed in the context of cognitive models because of their homogeneous structure and representations, and the dominance of learning underlying their theory, results in an approach so substantially different as to be not very relevant to the evidence discussed here. Another useful tool for cognitive modelling, COGENT (Cooper et al. 1998), was not mentioned because it is a tool for designing specific cognitive architectures, rather than an architecture for building a variety of agents.

Several limitations of Edmund were found which suggest directions for future work on the architecture. For one, it splits all of the responsibility for coherent behaviour between two very different systems, the central action selection and the control of the individual behaviours. Mammals have significant further assistance in smoothing and assimilating their various actions in the form of their cerebellum. Adding such a mechanism would probably be necessary for stringing together fine motions if Edmund were to be used to control something as intricate as a many-jointed robotic limb or a complex animated face.

Another interesting area for developing Edmund would be to automate the learning or adaptation of some of Edmund's basic components, particularly the action selection routines. As was suggested earlier in the text, this could be interesting if conducted on line, so that an agent might learn new competencies over its lifetime, either by experience or imitation. Alternatively, for psychological research, it might be more interesting to model the behaviour of living animals. Dawkins (1976) suggests several mechanisms for determining the hierarchies of animal behaviour control; these could probably be made even more powerful by using modern statistical and mathematical techniques for clustering and machine learning. Indeed, there is some considerable work being done in this area using Markov models, since they are more compliant to statistical approach (e.g. Baumberg & Hogg 1996, Brand et al. 1997). Dawkins (1976) argues against Markov representations, an argument that ethologists are still supporting, on the basis that the hierarchical representation with its emphasis on chains of decisions as well as trajectories within search spaces, seems a more natural expression of animal behaviour.

If a library of the primitive behaviours of these hierarchies could be adequately built into a simulation of the animal, then comparing the behaviour derived from these learned hierarchies with that of the real animal could be not only a strong test of this thesis, but also lead to a valuable experimental platform for researching animal behaviour.

There is, of course, much further work that could be done with Edmund as it exists, particularly in terms of modelling various types of theories of modular learning, such as episodic memory, but these are more ancillary to the main thesis of this dissertation. Discovering and modelling convincingly the hierarchical structure underlying the behaviour of real animals would be by far the richest direction for pursuing the results of this dissertation.

Chapter 6

Conclusion

This dissertation has presented a body of research supporting the claim that hierarchy and sequence are integral to intelligence, both natural and artificial. This has been argued from three perspectives:

- a literature review of agent architectures indicating a need for both distributed and hierarchical control in the same system,
- a body of laboratory research demonstrating that such a system can meet and exceed the standards of fully distributed, dynamic control, and
- a further literature review indicating that not only *can* hierarchical and sequential control be integrated into intelligent dynamic systems, but they *are* so integrated in animal intelligence.

Although none of this evidence might in itself be considered conclusive, the main contribution of this dissertation is to strengthen the claim that hierarchy and sequence are integral to intelligent systems by integrating these several sources of support.

This dissertation has made several additional contributions besides the direct support of its thesis. It provides a framework for considering evidence from artificial intelligence and psychology together, it examines methodological issues of performing artificial intelligence research in psychology laboratories, and it provides a detailed examination of one model from artificial intelligence in terms of psychological plausibility. These contributions support the main thesis indirectly by supporting the methodological approach of the dissertation.

The first chapter introduced the above thesis, and described its history and current controversy. It also provided a brief introduction to the methodological issues of the use of the cross disciplinary approach taken in establishing the thesis.

The second chapter examined an essential premise for the relevance of the dissertation: that some aspects of animal behaviour can be explored via their modelling on artificial platforms, and that further, since this approach to studying intelligence has been actively pursued for some time, at least some of the research already done should be applicable to psychology. It introduced the terminologies and problems of research in artificial intelligence considered most relevant to the aspects of psychology explored in this dissertation. In doing so it described the difference between the constructive and the descriptive stances taken by AI and psychology, respectively. The second part of the chapter demonstrated the commonality between the concerns of artificial intelligence and psychology by showing that the progress made in artificial intelligence in defining the utility of concepts such as intentionality, rationality, context and complexity does have relevance to current psychological research. This chapter suggested that although the progress in artificial intelligence is not *necessarily* relevant to psychology, it *may* be. Consequently psychologists should review AI results for potentially useful hypotheses.

The third chapter examined a number of such hypotheses on the optimal strategy for the control of intelligent behaviour as they have been encoded in AI agent architectures. This examination showed several trends in the histories of agent architectures:

- the need for pre-coded plans or plan elements in order to solve problems reliably and quickly,
- the need for the provision of reactive and opportunistic behaviour, and
- the need for limiting the attention of action selection mechanisms to a subset of possible behaviours appropriate to the context.

The first and third of these trends are relevant to the thesis of this dissertation, in that they describe sequential and hierarchical structure, respectively. The second is indicative of why older models that made no such allowances have been open to such harsh criticism from dynamic theories.

The third chapter also described in detail a particular architecture, Edmund, which meets all three of these criteria. This served as the basis for the experimental work described in the fourth chapter. This chapter demonstrated Edmund both on a mobile robot, where it was shown to be practical for this benchmark task of reactive intelligence, and in a simulated environment, where it was shown to be superior to several leading dynamic approaches. The fourth chapter also discussed the methodological issues of these two experimental test beds at length. Similarly, the fifth chapter contained further analysis of appropriate domains for using Edmund, suggested other platforms more appropriate for other domains, and suggested future work on both the architecture and the thesis problem.

More importantly to the thesis, the fifth chapter provided a survey of evidence from the life sciences, particularly neuroscience and psychology, for the existence of the sequential and hierarchical structuring of behaviour in animals. This evidence served along with the experimental evidence presented in the previous chapter, as further proof by demonstration that having sequential and hierarchical structuring underlying behaviour does not necessarily make that behaviour brittle or unresponsive. In fact, these chapters have shown evidence that such strategies support the mastery and combination of complex behaviours which can in turn provide greater robustness and flexibility in an animal or artifact. In conclusion, this dissertation has given considerable evidence that hierarchical and sequential structures are integral parts of both natural and artificial intelligence.

Bibliography

Adams, J. A. (1984), 'Learning of movement sequences', *Psychological Bulletin* **96**(1), 3–28.

Agre, P. E. & Chapman, D. (1987), Pengi: An implementation of a theory of activity, *in* 'Proceedings of the Sixth National Conference on Artificial Intelligence', Morgan Kaufmann, Seattle, Washington, pp. 196–201.

Agre, P. E. & Chapman, D. (1988), What are plans for?, AI memo 1050, MIT, Cambridge, MA.

Albus, J. S. (1997), 'The NIST real-time control system (RCS): an approach to intelligent systems research', *Journal of Experimental & Theoretical Artificial Intelligence* **9**(2/3), 147–156.

Anderson, J. R. & Matessa, M. (1998), The rational analysis of categorization and the ACT-R architecture, *in* M. Oaksford & N. Chater, eds, 'Rational Models of Cognition', Oxford University Press.

Appleby, S. & Steward, S. (1994), 'Mobile software agents for control in telecommnications networks', *BT Technology Journal* **12**(2), 104–113.

Arbib, M. A. & House, D. H. (1987), Depth and detours: An essay on visually guided behavior, *in* M. A. Arbib & A. R. Hanson, eds, 'Vision, Brain, and Cooperative Computation', The MIT Press, Cambridge MA.

Bacharach, M. & Hurley, S., eds (1991), *Foundations of decision theory: issues and advances*, Oxford University press. see particularly chapters by Levy and Pettit.

Baerends, G. P. (1976), 'The functional organisation of behaviour', *Animal Behaviour* **24**, 726–738.

Ballard, D. H., Hayhoe, M. M., Pook, P. K. & Rao, R. P. N. (1997), 'Deictic codes for the embodiment of cognition', *Brain and Behavioral Sciences* **20**(4).

Barlow, H. (1994), What is computational goal of the neocortex?, *in* C. Koch & J. L. Davis, eds, 'Large-Scale Neuronal Theories Of The Brain', mitpress, pp. 1–22.

Bates, E. (in press), Plasticity, localization and language development, *in* S. Broman & J. M. Fletcher, eds, 'The changing nervous system: Neurobehavioral consequences of early brain disorders', Oxford University Press.

Bates, J., Loyall, A. B. & Reilly, W. S. (1992), An architecture for action, emotion, and social behavior, Technical Report CMU-CS-92-144, CMU School of Computer Science, Pitsburgh, PA.

Baumberg, A. & Hogg, D. C. (1996), ‘Generating spatio-temporal models from examples’, *Image and Vision Computing* **14**(8), 525–532.

Beach, F. A., Hebb, D. O., Morgan, C. T. & Nissen, H. W., eds (1960), *The Neuropsychology of Lashley*, McGraw-Hill Book Company.

Benson, S. (1996), Learning Action Models for Reactive Autonomous Agents, PhD thesis, Stanford University. Department of Computer Science.

Bizzi, E., Giszter, S. ., Loeb, E., Mussa-Ivaldi, F. A. & Saltiel, P. (1995), ‘Modular organization of motor behavior in the frog’s spinal cord’, *Trends in Neuroscience* **18**, 442–446.

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

Bohm, D. (1980), *Wholeness and the Implicate Order*, Routledge, London.

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

Brady, M. & Paul, R., eds (1984), *Robotics Research: The First International Symposium*, The MIT Press, Cambridge MA.

Braitenberg, V. (1984), *Vehicles: Experiments in Synthetic Psychology*, MIT Press, Cambridge, Massachusetts.

Brand, M. (1995), Explanation-mediated vision: Making sense of the world through causal analysis, PhD thesis, Northwestern University.

Brand, M., Oliver, N. & Pentland, A. (1997), Coupled hidden markov models for complex action recognition, *in* ‘Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR97)’.

Bratman, M. E., Israel, D. J. & Pollack, M. E. (1988), ‘Plans and resource-bounded practical reasoning’, *Computational Intelligence* **4**, 349–355.

Brooks, Jr., F. P. (1975), *The Mythical Man-month: Essays on Software Engineering*, Addison-Wesley Publishing Company, Reading, MA.

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. (1989), ‘A robot that walks: Emergent behavior from a carefully evolved network’, *Neural Computation* **1**(2), 253–262.

Brooks, R. A. (1990), Elephants don’t play chess, *in* P. Maes, ed., ‘Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back’, MIT Press, Cambridge, MA, pp. 3–15.

Brooks, R. A. (1991*a*), Intelligence without reason, *in* ‘Proceedings of the 1991 International Joint Conference on Artificial Intelligence’, Sydney, pp. 569–595.

Brooks, R. A. (1991*b*), 'Intelligence without representation', *Artificial Intelligence* **47**, 139–159.

Brooks, R. A., Breazeal, C., Irie, R., Kemp, C. C., Marjanovic, M., Scassellati, B. & Williamson, M. (1998), Alternate essences of intelligence, *in* 'American Assocation of Artificial Intelligence (AAAI98)', Madison, WI, pp. 961–968.

Brooks, R. A. & Stein, L. A. (1993), Building brains for bodies, Memo 1439, Massachusetts Institute of TechnologyArtificial Intelligence Lab, Cambridge, MA.

Brunelli, R. & Poggio, T. (1993), 'Face recognition: Features versus templates', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(10), 1042–1052.

Bruner, J. S. (1982), The organisation of action and the nature of adult-infant transaction, *in* M. von Cranach & R. Harré, eds, 'The Analysis of Action', Cambridge University Press, pp. 313–328.

Bruner, J. S. (1990), *Acts of Meaning*, Harvard University Press, Cambridge, MA.

Bryson, J. (1992), The subsumption strategy development of a music modelling system, Master's thesis, University of Edinburgh. Department of Artificial Intelligence.

Bryson, J. (1996*a*), The design of learning for an artifact, *in* 'AISB Symposium on Learning in Robots and Animals'. unpublished proceedings, paper is available from http://www.ai.mit.edu/ joanna/publications.html.

Bryson, J. (1996*b*), The use of state in intelligent control. unpublished working paper.

Bryson, J. (1999), Creativity by design: A behaviour-based approach to creating creative play, *in* F. Nack, ed., 'AISB'99 Symposium on Creativity in Entertainment and Visual Art', The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, Sussex, pp. 9–16.

Bryson, J. (n.d.), Specialized learning and the design of intelligent agents.

Bryson, J. & McGonigle, B. (1998), Agent architecture as object oriented design, *in* M. P. Singh, A. S. Rao & M. J. Wooldridge, eds, 'The Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL97)', Springer-Verlag, pp. 15–30.

Byrne, R. W. & Russon, A. E. (1998), 'Learning by imitation: a hierarchical approach', *Brain and Behavioral Sciences* **21**(5), 667–721.

Calvin, W. H. (1996), *The Cerebral Code*, MIT Press.

Carlson, N. R. (2000), *Physiology of Behavior*, Allyn and Bacon, Boston.

Chalmers, M. & McGonigle, B. (1984), 'Are children any more logical than monkeys on the five term series problem?', *Journal of Experimental Child Psychology* **37**, 355–377.

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

Chapman, D. (1989), 'Penguins can make cake', *AI Magazine* **10**(4), 51–60.

Chomsky, N. (1957), *Syntactic Structures*, Mouton, The Hague.

Chomsky, N. (1980), ‘Rules and representations’, *Brain and Behavioral Sciences* **3**, 1–61.

Clark, A. (1996), *Being There: Putting Brain, Body and World Together Again*, MIT Press, Cambridge, MA.

Connell, J. H. (1990), *Minimalist Mobile Robotics: A Colony-style Architecture for a Mobile Robot*, Academic Press, Cambridge, MA. also MIT TR-1151.

Cooper, R., Shallice, T. & Farringdon, J. (1995), Symbolic and continuous processes in the automatic selection of actions, *in* J. Hallam, ed., ‘Hybrid Problems, Hybrid Solutions, Frontiers in Artificial Intelligence and Applications’, IOS Press, Amsterdam, pp. 27–37.

Cooper, R., Yule, P., Fox, J. & Sutton, D. (1998), COGENT: An environment for the development of cognitive models, *in* U. Schmid, J. F. Krems & F. Wysotzki, eds, ‘A Cognitive Science Approach to Reasoning, Learning and Discovery’, Pabst Science Publishers, Lengerich, Germany, pp. 55–82. see also http://cogent.psyc.bbk.ac.uk/.

Correia, L. & Steiger-Garção, A. (1995), A useful autonomous vehicle with a hierarchical behavior control, *in* F. Moran, A. Moreno, J. Merelo & P. Chacon, eds, ‘Advances in Artificial Life (Third European Conference on Artificial Life)’, Springer, Berlin, pp. 625–639.

Cosmides, L. & Tooby, J. (1992), Cognitive adaptations for social exchange, *in* J. H. Barkow, L. Cosmides & J. Tooby, eds, ‘The Adapted Mind’, Oxford University press, pp. 163–228.

Cosmides, L. & Tooby, J. (1994), Origins of domain specificity: the evolution of functional organization, *in* L. A. Hirschfeld & S. A. Gelman, eds, ‘Mapping the Mind: Domain Specificity in Cognition and Culture’, Cambridge University Press.

Dailey, M. N. & Cottrell, G. W. (1997), Task and spatial frequency effects on face specialization, *in* ‘Advances in Neural Information Processing Systems 10’, Lawrence Erlbaum Associates.

Dautenhahn, K. & Nehaniv, C., eds (1999), *Proceedings of the AISB’99 Symposium on Imitation in Animals and Artifacts*, AISB, Edinburgh.

Dawkins, R. (1976), Hierarchical organisation: A candidate principle for ethology, *in* P. P. G. Bateson & R. A. Hinde, eds, ‘Growing Points in Ethology’, Cambridge University Press, Cambridge, pp. 7–54.

Demiris, J. & Hayes, G. (1999), Active and passive routes to imitation, *in* ‘Proceedings of the AISB’99 Symposium on Imitation in Animals and Artifacts’, AISB, Edinburgh.

Dennett, D. C. (1991), *Consciousness Explained*, Little, Brown, Boston, MA.

Dennett, D. C. (1995), *Darwin’s Dangerous Idea*, Penguin.

Dennett, D. C. (1997), Consciousness in human and robot minds, *in* M. Ito, Y. Miyashita & E. T. Rolls, eds, ‘Cognition, Computation, and Consciousness’, Oxford University Press.

Dennett, D. C. & Kinsbourne, M. (1992), ‘Time and the observer: The where and when of consciousness in the brain’, *Brain and Behavioral Sciences* **15**, 183–247.

Dickinson, A. R. (1998), Hierarchical Organisation in Serial Search Tasks by *Cebus Apella* Monkeys, PhD thesis, University Of Edinburgh Department of Psychology.

Donnett, J. & McGonigle, B. (1991), Evolving speed control in mobile robots: from blindness to kinetic vision, *in* 'Proceedings of the Vision Interface Conference', Calgary.

Dorigo, M. & Colombetti, M. (1998), *Robot Shaping: An Experiment in Behavior Engineering*, MIT Press / Bradford Books.

Dreyfus, H. L. (1992), *What Computers Still Can't Do*, MIT Press, Cambridge, MA.

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D. & Plunkett, K. (1996), *Rethinking Innateness. A Connectionist Perspective on Development*, MIT Press, Cambridge, MA.

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

Firby, J. (1996), Modularity issues in reactive planning, *in* 'Proceedings of the Third International Conference on AI Planning Systems', pp. 78–85.

Fodor, J. A. (1983), *The Modularity of Mind*, Bradford Books, MIT Press, Cambridge, MA.

Freud, S. (1900), *The Interpretation of Dreams*, Avon, New York.

Gadanho, S. C. (1999), Reinforcement Learning in Autonomous Robots: An Empirical Investigation of the Role of Emotions, PhD thesis, University of Edinburgh.

Gallistel, C., Brown, A. L., Carey, S., Gelman, R. & Keil, F. C. (1991), Lessons from animal learning for the study of cognitive development, *in* S. Carey & R. Gelman, eds, 'The Epigenesis of Mind', Lawrence Erlbaum, Hillsdale, NJ, pp. 3–36.

Gallistel, C. R. (1980), *The Organization of Action: a New Synthesis*, Erlbaum, Hilldale, NJ.

Gallistel, C. R. (1990), *The Organization of Learning*, MIT Press / Bradford Books, Cambridge, MA.

Gardner, M. & Heyes, C. (1998), 'Splitting lumping and priming', *Brain and Behavioral Sciences* **21**(5), 695.

Gat, E. (1991), Reliable Goal-Directed Reactive Control of Autonomous Mobile Robots, PhD thesis, Virginia Polytechnic Institute and State University.

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

Glasspool, D. W. (1995), Competitive queuing and the articulatory loop, *in* J. Levy, D. Bairaktaris, J. Bullinaria & P. Cairns, eds, 'Connectionist Models of Memory and Language', UCL Press.

Gleitman, H. (1995), *Psychology*, 4 edn, Norton.

Goldfield, E. C. (1995), *Emergent Forms: Origins and Early Development of Human Action and Perception*, Oxford University Press.

Grand, S., Cliff, D. & Malhotra, A. (1997), Creatures: Artificial life autonomous software agents for home entertainment, *in* W. L. Johnson, ed., 'Proceedings of the First International Conference on Autonomous Agents', ACM press, pp. 22–29.

Greenfield, P. M. (1991), 'Language, tools and brain: The ontogeny and phylogeny of hierarchically organized sequential behavior', *Brain and Behavioral Sciences* **14**, 531–595.

Gurney, K., Prescott, T. J. & Redgrave, P. (1998), The basal ganglia viewed as an action selection device, *in* 'The Proceedings of the International Conference on Artificial Neural Networks', Skovde, Sweden.

Hallam, B. E., Halperin, J. R. P. & Hallam, J. C. T. (1995), 'An ethological model for implementation on mobile robots', *Adaptive Behavior* **3**, 51–80.

Hallam, B. & Hayes, G. M. (1992), Comparing robot and animal behaviour, DAI Research Paper 598, University of Edinburgh, Edinburgh, Scotland.

Hammond, K. J. (1990), 'Case-based planning: A framework for planning from experience', *The Journal of Cognitive Science* **14**(3).

Hanks, S., Pollack, M. E. & Cohen, P. R. (1993), Benchmarks, testbeds, controlled experimentation and the design of agent architectures, Technical Report 93–06–05, Department of Computer Science and Engineering, University of Washington.

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

Henson, R. N. A. (1996), Short-term Memory for Serial Order, PhD thesis, University of Cambridge. St. John's College.

Henson, R. N. A. & Burgess, N. (1997), Representations of serial order, *in* J. A. Bullinaria, D. W. Glasspool & G. Houghton, eds, 'Proceedings of the Fourth Neural Computation and Psychology Workshop: Connectionist Representations', Springer, London.

Hexmoor, H. H. (1995), Representing and Learning Routine Activities, PhD thesis, State University of New York at Buffalo.

Hexmoor, H., Horswill, I. & Kortenkamp, D. (1997), 'Special issue: Software architectures for hardware agents', *Journal of Experimental & Theoretical Artificial Intelligence* **9**(2/3).

Hinton, G. E. & Ghahramani, Z. (1997), 'Generative models for discovering sparse distributed representations', *Philosophical Transactions of the Royal Society B* **352**, 1177–1190.

Horn, B. K. P. & Winston, P. H. (1976), A laboratory environment for applications oriented vision and manipulation, Technical Report 365, MIT AI Laboratory.

Horswill, I. D. (1993*a*), Specialization of Perceptual Processes, PhD thesis, MIT, Department of EECS, Cambridge, MA.

Horswill, I. D. (1993*b*), Specialization of Perceptual Processes, PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Horswill, I. D. (1995), Visual routines and visual search, *in* 'Proceedings of the 14th International Joint Conference on Artificial Intelligence', Montreal.

Horswill, I. D. (1997), 'Visual architecture and cognitive architecture', *Journal of Experimental & Theoretical Artificial Intelligence* **9**(2/3), 277–293.

Houghton, G. & Hartley, T. (1995), 'Parallel models of serial behavior: Lashley revisited', *PSYCHE* **2**(25).

Hubel, D. H. (1988), *Eye, Brain and Vision*, Freeman.

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

Hume, D. (1748), *Philisophical Essays Concerning Human Understanding*, Andrew Millar, London.

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

Humphrys, M. (1997), Action Selection methods using Reinforcement Learning, PhD thesis, University of Cambridge.

Jakobi, N. (1997), 'Evolutionary robotics and the radical envelope of noise hypothesis', *Journal Of Adaptive Behaviour* **6**(2), 325–368.

Jorion, P. J. M. (1998), 'A methodological behaviourist model for imitation', *Brain and Behavioral Sciences* **21**(5), 695.

Jusczyk, P. (1985), On characterizing the development of speech perception, *in* J. Mehler & R. Fox, eds, 'Neonate cognition: Beyond the blooming buzzing confusion', Erlbaum, Hillsdale, NJ.

Karmiloff-Smith, A. (1992), *Beyond Modularity: A Developmental Perspective on Cognitive Change*, MIT Press, Cambridge, MA.

Kelso, J. S. (1995), *Dynamic Patterns: The Self-Organization of Brain and Behavior*, MIT Press.

Kitano, H. (1998), 'Special issue: Robocup', *Applied Artificial Intelligence* **12**(2–3).

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I. & Osawa, E. (1997), RoboCup: The robot world cup initiative, *in* 'Proceedings of The First International Conference on Autonomous Agents', The ACM Press.

Kobayashi, T., Nishijo, H., Fukuda, M., Bures, J. & Ono, T. (1997), 'Task-dependent representations in rat hippocampal place neurons', *JOURNAL OF NEUROPHYSIOLOGY* **78**(2), 597–613.

Kolodner, J. L. (1993), *Case-Based Reasoning*, Morgan Kaufmann.

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

Kuniyoshi, Y. & Nagakubo, A. (1997), Humanoid as a research vehicle into flexible complex interaction, *in* 'The Tenth International Conference on Intelligent Robots and Systems (IROS97)'.

Laird, J. E. (1997), ‘Plan execution architecture evaluation’, proposal on the world wide web, http://ai.eecs.umich.edu/people/laird/nsf.html.

Laird, J. E. & Rosenbloom, P. S. (1996), The evolution of the Soar cognitive architecture, *in* D. Steier & T. Mitchell, eds, ‘Mind Matters’, Erlbaum.

Lashley, K. S. (1950), In search of the engram, *in* ‘Physiological Mechanisms in Animal Behavior’, number 4 *in* ‘Society of Experimental Biology Symposium’, Cambridge University Press. reprinted in Beach et al. (1960).

Lashley, K. S. (1951), The problem of serial order in behavior, *in* L. A. Jeffress, ed., ‘Cerebral mechanisms in behavior’, John Wiley & Sons, New York.

Lee, D. N. & Aronson, E. (1974), ‘Visual proprioceptive control of standing infants’, *Perception and Psychophysics* **15**, 529–532.

Lee, D. N. & Lishman, J. R. (1975), ‘Visual proprioceptive control of stance.’, *Jojurnal of Human Movement Studies* **1**, 87–95.

Leon, V. J., Kortenkamp, D. & Schreckenghost, D. (1997), A planning, scheduling and control architecture for advanced life support systems, *in* ‘NASA Workshop on Planning and Scheduling for Space’, Oxnard, CA.

Levison, L. (1996), Connecting Planning and Action via Object-Specific Reasoning, PhD thesis, University of Pennsylvania. School of Engineering and Applied Science.

Levy, W. B. (1996), ‘A sequence predicting CA3 is a flexible associator that learns and uses context to solve hippocampal-like tasks’, *Hippocampus* **6**(6), 579–590.

Livesey, P. J. (1986), *Learning and Emotion: A Biological Synthesis*, Vol. 1 of *Evolutionary Processes*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Lorenz, K. (1950), The comparative method in studying innate behaviour patterns, *in* ‘Symposia of the Society for Experimental Biology’, Vol. 4, pp. 221–268.

Lorenz, K. (1973), *Foundations of Ethology*, Springer, New York.

Mac Aogáin, E. (1998), ‘Imitation without attitudes’, *Brain and Behavioral Sciences* **21**(5), 696–697.

MacDonald, J. & McGurk, H. (1978), ‘Visual influence on speech perception process’, *Perception and Psychophysics* **24**(3), 253–257.

MacKay, D. M. (1956), The epistemological problem for automata, *in* ‘Automata Studies’, Princeton University Press, pp. 235–251.

Maes, P. (1989), How to do the right thing, A.I. Memo 1180, MIT, Cambridge, MA.

Maes, P. (1990*a*), Situated agents can have goals, *in* P. Maes, ed., ‘Designing Autonomous Agents : Theory and Practice from Biology to Engineering and back’, MIT Press, Cambridge, MA, pp. 49–70.

Maes, P. (1991), A bottom-up mechanism for behavior selection in an artificial creature, *in* J.-A. Meyer & S. Wilson, eds, ‘From Animals to Animats’, MIT Press, Cambridge, MA, pp. 478–485.

Maes, P., ed. (1990*b*), *Designing Autonomous Agents : Theory and Practice from Biology to Engineering and back*, MIT Press, Cambridge, MA.

Malcolm, C. (1997), A hybrid behavioural/knowledge-based approach to robotic assembly, DAI Research Paper 875, University of Edinburgh, Edinburgh, Scotland.

Malcolm, C., Smithers, T. & Hallam, J. (1989), An emerging paradigm in robot architecture, *in* 'Proceedings of IAS', Vol. 2, Amsterdam, Netherlands.

Marjanovic, M., Scassellati, B. & Williamson, M. (1996), Self-taught visually-guided pointing for a humanoid robot, *in* P. Maes, M. J. Matarić, J.-A. Meyer, J. Pollack & S. W. Wilson, eds, 'From Animals to Animats 4 (SAB96)', MIT Press, Cambridge, MA.

Matarić, M. J. (1990), A distributed model for mobile robot environment-learning and navigation, Technical Report 1228, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Matarić, M. J. (1997), 'Behavior-based control: Examples from navigation, learning, and group behavior', *Journal of Experimental & Theoretical Artificial Intelligence* **9**(2/3), 323–336.

Matarić, M. J. & Cliff, D. (1996), 'Challenges in evolving controllers for physical robots', *Robotics and Autonomous Systems* **19**(1), 67–83.

Matheson, T. (1997), 'Hindleg targeting during scratching in the locust', *Journal of Experimental Biology* **200**, 93–100.

McClelland, J. L., McNaughton, B. L. & O'Reilly, R. C. (1995), 'Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory', *Psychological Review* **102**(3), 419–457.

McClelland, J. L. & Rumelhart, D. E., eds (1988), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press. two volumes.

McDougall, W. (1923), *Outline of Psychology*, Methuen, London.

McFarland, D. (1985), *Animal Behaviour*, Benjamin/Cummings, Menlo Park, California.

McFarland, D. (1987), *The Oxford Companion to Animal Behaviour*, Oxford University Press, Oxford, United Kingdom.

McGonigle, B. (1991), Incrementing intelligent systems by design, *in* J.-A. Meyer & S. Wilson, eds, 'From Animals to Animats', MIT Press, Cambridge, MA, pp. 478–485.

McGonigle, B. (in press), Robotic experiments and cognition, *in* O. Holland & D. McFarland, eds, 'Artificial Ethology', Oxford University Press.

McGonigle, B. & Chalmers, M. (1977), 'Are monkeys logical?', *Nature* **267**.

McGonigle, B. & Chalmers, M. (1996), The ontology of order, *in* L. Smith, ed., 'Piaget: A Critical Assessment', Routledge.

McGonigle, B. & Chalmers, M. (1998), Rationality as optimised cognitive self-regulation, *in* M. Oaksford & N. Chater, eds, 'Rational Models of Cognition', Oxford University Press.

McGonigle, B. & Johnston, B. S. (1995), Applying staged cognitive growth principles to the design of autonomous robots, Technical report, Laboratory for Cognitive Neuroscience and Intelligent Systems, Department of Psychology, University of Edinburgh.

McGurk, H. & MacDonald, J. (1976), 'Hearing lips and seeing voices', *Nature* **264**, 746–748.

Michaud, F. & Matarić, M. J. (1999), 'Representation of behavioral history for learning in nonstationary conditions', *Robotics and Autonomous Systems* **29**(2).

Minsky, M. (1985), *The Society of Mind*, Simon and Schuster Inc., New York, NY.

Mithen, S. (1996), *The Prehistory of Mind*, Phoenix, London.

Mussa-Ivaldi, F. A., Giszter, S. F. & E.Bizzi (1994), Linear combinations of primitives in vertebrate moto control, *in* 'Proceedings of the National Acadamy of Sciences', Vol. 91 of *Neurobiology*, pp. 7534–4538.

Neely, J. H. (1991), Semantic priming effects in visual word recognition: A selective review of current findings and theories, *in* D. Besner & G. W. Humphreys, eds, 'Basic Processes in Reading: Visual Word Recognition', Lawrence Erlbaum Associates, chapter 9.

Nehmzow, U. & McGonigle, B. (1993), Robot navigation by light, *in* 'European Conference on Artificial Life (ECAL93)'.

Nehmzow, U. & McGonigle, B. (1994), Achieving rapid adaptations in robots by means of external tuition, *in* 'From Animals to Animats 3 (SAB94)', mitpress.

Nehmzow, U., Recce, M. & Bisset, D. (1997), Towards intelligent mobile robots - scientific methods in mobile robotics, Technical Report UMCS-97-9-1, University of Manchester Computer Science. Edited collection of papers, see also related special issue of *Journal of Robotics and Autonomous Systems*, in preparation.

Nehmzow, U., Smithers, T. & McGonigle, B. (1993), Increasing behavioural repertoire in a mobile robot, *in* 'From Animals to Animats 2 (SAB92)', MIT Press, pp. 291–297.

Nelson, K. (1990), 'Hierarchical organization, revisited', *Netherlands Journal of Zoology* **40**(4), 585–616.

Newell, A. (1982), 'The knowledge level', *Artificial Intelligence* **18**(1), 87–127.

Newell, A. (1990), *Unified Theories of Cognition*, Harvard University Press, Cambridge, Massachusetts.

Newell, A. & Simon, H. A. (1972), *Human Problem Solving*, Prentice-Hall.

Nicolelis, M. A. L., Baccala, L. A., Lin, R. C. S. & Chapin, J. K. (1995), 'Sensori-motor encoding by synchronous neural ensemble activity at multiple levels of the somoatosensory system', *Science* **268**, 1353–1358.

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.

Norman, D. A. (1998), *The Invisible Computer*, MIT Press.

Norman, D. A. & Shallice, T. (1986), Attention to action: Willed and automatic control of behavior, *in* R.Davidson, G. Schwartz & D. Shapiro, eds, 'Consciousness and Self Regulation: Advances in Research and Theory', Vol. 4, Plenum, New York, pp. 1–18.

Parker, L. E. (1998), 'ALLIANCE: An architecture for fault tolerant multi-robot cooperation', *IEEE Transactions on Robotics and Automation* **14**(2).

Pebody, M. (1995), Learning and adaptivity: Enhancing reactive behaviour architectures in real-world interaction systems, *in* F. Moran, A. Moreno, J. Merelo & P. Chacon, eds, 'Advances in Artificial Life (Third European Conference on Artificial Life)', Springer, Berlin, pp. 679–690.

Perkins, S. (1998), Incremental Acquisition of Complex Visual Behaviour using Genetic Programming and Shaping, PhD thesis, University of Edinburgh. Department of Artificial Intelligence.

Perrett, D. I., Hietanen, J. K., Oram, M. W. & Benson, P. J. (1992), 'Organisation and functions of cells responsive to faces in the temporal cortex', *Philosophical Transactions of the Royal Society of London* **335**, 25–30.

Perrett, D. I., Mistlin, A. J. & Chitty, A. J. (1987), 'Visual neurones responsive to faces.', *Trends in Neuroscience* **9**, 358–364.

Piaget, J. (1954), *The Construction of Reality in the Child*, Basic Books, New York.

Plaut, D. C. (1999), Systematicity and specialization in semantics, *in* D. Heinke, G. W. Humphreys & A. Olson, eds, 'Connectionist models in cognitive neuroscience: Proceedings of the Fifth Annual Neural Computation and Psychology Workshop', Springer-Verlag, New York.

Pöppel, E. (1994), 'Temporal mechanisms in perception', *International Review of Neurobiology* **37**, 185–202.

Port, R. F. & van Gelder, T., eds (1995), *Mind as Motion: Explorations in the Dynamics of Cognition*, MIT Press, Cambridge, MA.

Premack, D. & Premack, A. J. (1983), *The Mind of an Ape*, W.W. Norton and Company.

Prescott, T. J., Redgrave, P. & Gurney, K. (1999), 'Layered control architectures in robots and vertebrates', *Adaptive Behavior* (7), 99–127.

Ram, A. & Santamaria, J. C. (1997), 'Continuous case-based reasoning', *Artificial Intelligence* **90**(1–2), 25–77.

Ramachandran, V. S. & Blakeslee, S. (1998), *Phantoms in the brain: Human nature and the architecture of the mind*, Fourth Estate, London.

Rao, R. P. N. & Ballard, D. H. (1996), The visual cortex as a hierarchical predictor, Technical Report 96.4, University of Rochester. National Resource Laboratory for the Study of Brain and Behavior, Computer Science Department.

Redgrave, P., Prescott, T. J. & Gurney, K. (1999), ‘The basal ganglia: a vertebrate solution to the selection problem?’, *Neuroscience* **89**, 1009–1023.

Reeves, R. & Hallam, J. C. (1995), Control of walking by central pattern generators, *in* ‘Intelligent Autonomous Systems Four (IAS95)’, Karlsruhe. also Edinburgh DAI RP-726.

Reilly, W. S. N. (1996), Believable social and emotional agents, Technical Report CMU-CS-96-138, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. Ph.D. thesis.

Rhodes, B. (1995), Pronemes in behavior nets, Technical Report 95-01, MIT. Media Lab, Learning and Common Sense.

Rhodes, B. (1996), PHISH-nets: Planning heuristically in situated hybrid networks, Master’s thesis, MIT. Media Lab, Learning and Common Sense.

Roper, T. J. (1983), Learning as a biological phenomena, *in* T. R. Halliday & P. J. B. Slater, eds, ‘Genes, Development and Learning’, Vol. 3 of *Animal Behaviour*, Blackwell Scientific Publications, Oxford, chapter 6, pp. 178–212.

Rosenblatt, K. & 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’.

Russell, S. J. & Norvig, P. (1995), *Artificial Intelligence : A Modern Approach*, Prentice Hall, Englewood Cliffs, NJ.

Sengers, P. (1998), Do the thing right: An architecture for action expression, *in* K. P. Sycara & M. Wooldridge, eds, ‘Proceedings of the Second International Conference on Autonomous Agents’, ACM Press, pp. 24–31.

Sharkey, N. E. (1998), ‘Learning from innate behaviors: a quantitative evaluation of neural network controllers’, *Machine Learning* **31**, 115–139.

Sherman, S. M. & Koch, C. (1990), Thalamus, *in* G. M. Shepherd, ed., ‘Synaptic organization of the brain’, 3 edn, Oxford University Press, pp. 246–278.

Sillito, A., Jones, H., Gerstein, G. & West, D. (1994), ‘Feature-linked synchronization of thalamic relay cell firing induced by feedback from the visual cortex’, *Nature* **369**, 479–482.

Simon, H. A. (1969), *The Sciences of the Artificial*, MIT Press, Cambridge, MA.

Simon, H. A. (1995), Explaining the ineffable: AI on the topics of intuition, insight and inspiration, *in* ‘Proceedings of the 14th International Joint Conference on Artificial Intelligence’, Morgan Kaufmann, Montreal.

Singh, M. P., ed. (1998), *The Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL97)*, Springer, Providence, RI.

Sipser, M. (1997), *Introduction to the Theory of Computation*, PWS Publishing Company.

Sirosh, J. & Miikkulainen, R. (1996), 'Self-organization and functional role of lateral connections and multisize receptive fields in the primary visual cortex', *Neural Processing Letters* .

Skinner, B. F. (1935), 'The generic nature of the concepts of stimulus and response.', *Journal of General Psychology* **12**, 40–65.

Sloman, A. & Logan, B. (1998), Architectures and tools for human-like agents, *in* 'Proceedings of the Second European Conference on Cognitive Modelling', University of Nottingham Press, Nottingham, UK, pp. 58–65.

Smart, W. D. (1992), Location recognition wih neural networks in a mobile robot, Master's thesis, University of Edinburgh. Department of Artificial Intelligence.

Smirnakis, S. M., Berry, M. J., Warland, D. K., Bialek, W. & Meister, M. (1997), 'Adaptation of retinal processing to image contrast and spatial scale', *Nature* **386**, 69–73.

Steels, L. (1994), Building agents with autonomous behavior systems, *in* L. Steels & R. Brooks, eds, 'The 'artificial life' route to 'artificial intelligence'. Building situated embodied agents.', Lawrence Erlbaum Associates, New Haven.

Tanji, J. (1996), Involvement of motor areas in the medial frontal cortex of primates in temporal sequencing of multiple movements, *in* R. Caminiti, K. Hoffmann, F. Lacquaniti & J. Altman, eds, 'Vision and Movement: Mechanisms in the Cerebral Cortex', Vol. 2, Human Frontier Science Program, Strasbourg, pp. 126–133.

Tanji, J. & Shima, K. (1994), 'Role for supplementary motor area cells in planning several movements ahead', *Nature* **371**, 413–416.

Teyler, T. J. & Discenna, P. (1986), 'The hippocampal memory indexing theory.', *Behavioral Neuroscience* **100**, 147–154.

Thórisson, K. R. (1996), Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills, PhD thesis, MIT Media Laboratory.

Thórisson, K. R. (1999), 'A mind model for multimodal communicative creatures & humanoids', *International Journal of Applied Artificial Intelligence* **13**(4/5), 519–538.

Thrun, S., Büchen, A., Burgard, W., Fox, D., Fröhlinghaus, T., Henning, D., Hofmann, T., Krell, M. & Schmidt, T. (1998), Map learning and high-speed navigation in RHINO, *in* D. Kortenkamp, R. P. Bonasso & R. Murphy, eds, 'Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems', MIT Press, Cambridge, MA, chapter 1, pp. 21–52.

Tinbergen, N. (1951), *The Study of Instinct*, Clarendon Press, Oxford.

Trullier, O., Wiener, S., Berthoz, A. & Meyer, J. A. (1997), 'Biologically-based artificial navigation systems: Review and prospects.', *Progress in Neurobiology* **51**, 483–544.

Turing, A. M. (1950), 'Computing machinery and intelligence', *Mind* **59**, 433–460.

Turing, A. M. (1970), Intelligent machinery, *in* B. Meltzer & D. Michie, eds, 'Machine Intelligence 5', American Elsevier Publishing, New York, NY, pp. 3–23.

Tversky, A. & Kahneman, D. (1981), 'The framing of decisions and the psychology of choice', *Science* **211**, 453–58.

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

Ullman, S. (1984), 'Visual routines', *Cognition* **18**, 97–159.

van Gelder, T. (1998), 'The dynamical hypothesis in cognitive science', *Behavioral and Brain Sciences* **21**(5), 616–665.

Vereijken, B. & Whiting, H. T. A. (1998), 'Hoist by their own petard: The constraints of hierarchical models', *Brain and Behavioral Sciences* **21**(5), 695.

Walter, W. G. (1950), 'An imitation of life', *Scientific American* **182**(5), 42–45.

Warnett, L. (forthcoming), Intelligent systems: towards a new synthetic agenda, PhD thesis, The University of Edinburgh Department of Psychology.

Webb, B. (1996), 'A robot cricket', *Scientific American* **275**(6), 62–67.

Wennekers, T. & Palm, G. (to appear), Cell assemblies, associative memory and temporal structure in brain signals, *in* R. Miller, ed., 'Time and the Brain.', Vol. 2 of *Conceptual Advances in Brain Research*, Harwood Academic Publishers.

Whitehead, S. D. (1992), Reinforcement learning for the adaptive control of perception and action, Technical Report 406, University of Rochester Computer Science, Rochester, NY.

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.

Wilson, M. & McNaughton, B. (1994), 'Reactivation of hippocampal ensemble memories during sleep', *Science* **261**, 1227–1232.

Winston, P. H. (1992), *Artificial Intelligence*, third edition edn, Addison-Wesley, Reading, Massachusetts.

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

Wyatt, J., Hoar, J. & Hayes, G. (1998), 'Design, analysis and comparison of robot learners', *Robotics and Autonomous Systems* **24**(1–2), 17–32.

Zelinsky, A. (1991), 'Mobile robot map making using sonar', *Journal of Robotic Systems* **8**(5), 557–577.

Appendix A

The Edmund Mouse

The code for Tyrrell's Simulated Environment is available from his web page: http://www.soc.soton.ac.uk/SOES/STAFF/tt/index.html

The additional code needed for running the simulations documented in Chapter 4 is available here: ftp://ftp.ai.mit.edu/pub/users/joanna/edmund-mouse.tar.gz

For quick perusal, some of the critical files associated with those documents are available below.

A.1 Edmund Mouse Control Script

This file is read by edmund. It has comments, and some of the competences and action patterns are not used, because they are not children of the drive "life". I leave them in here as they provide some record of the development process.

```
# animal grooms towards sunset, if near home but not in it
# (or nothing else is up!)
(life () (nil) DRIVE ((flee 80) (mate 70) (home 60 40 10) (triangulate
  60) (check 55 50 10) (forage 50) (hurry_home 45) (sleep_at_home 50)
  (sneak 45) (clean 20) (sit 10)))
(flee ((sniff_predator t)) () ((freeze 50) (run_away 60)
  (look_for_predator 40)))
(sleep_at_home ((at_home t) (day_time nil)) (sleep))

# not covered if predator in square!
(freeze ((see_predator t) (predator1 nil) (covered t)) (hold_still))
(run_away ((see_predator t)) (pick_new_sheltered_dir go_fast))
(look_for_predator () (observe_predator))
(check ((at_home nil)) (look_around))

# cleaning also happens in exploit
(clean ((at_home nil)) (groom))
(sit () (rest))

# go home until get there if late!
(home ((late t))) ((at_home t)) (go_home))
```

```
(triangulate ((lost t)) (go_home))
(hurry_home ((day_time nil) (at_home nil)) (go_home))
(cruise ((day_time t)) (nil) ((mate 90) (exploit 60)))
(cruising () (nil) ((mate 90) (exploit 60)))
(mate ((sniff_mate t)) (nil) ((inseminate 90) (court 80) (pursue 70)))

# can’t see anything in the dark, this is useless
(night_mate ((see_mate t)) (nil) ((inseminate 90) (court 80) (pursue 70)))
(inseminate ((at_home nil) (courted_mate_here t)) (copulate))
(court ((at_home nil) (mate_here t)) (strut))
(pursue ((sniff_mate t)) (pick_mate_dir go))

# use this fit_to... only if full cruise calls for it
# (exploit ((fit_to_exploit t)) ((use_resource 80) (leave 60)))
(exploit ((day_time t)) (nil) ((use_resource 80) (leave 60)))
(use_resource ((at_home nil) (needed_resource_available t)) (exploit_resource))
(leave () (pick_dir go))

# these replace exploit -- mostly stay inconspicuous (and groom),
# forage only when needed
(forage ((day_time t) (hungry t)) (nil) ((use_resource 80) (seek_resource 60)))
(fix_cold ((day_time t) (cold t)) (pick_sheltered_dir go_fast))
(fix_hot ((day_time t) (hot t)) (nil) ((chill 80) (seek_cold 60))
(chill ((cool_spot t)) (rest))
(seek_cold () (pick_new_sheltered_dir go))
(seek_resource () (pick_forage_dir go))
(sneak ((day_time t)) (nil) ((get_safe 80) (preen 70) (warm_up 60)
  (relax 50) (wander 40)))
(get_safe ((covered nil)) (pick_new_sheltered_dir go))
(preen ((dirty t) (at_home nil)) (groom))
(relax ((tired t)) (rest))
(warm_up ((cold t)) (pick_new_sheltered_dir go_fast))
(wander () (pick_new_sheltered_dir go))
```

A.2 The Primitive Interface

This file is the interface between the behaviours (in this case provided mostly by the simulation) and the control hierarchy coded above.

```
/* em-prims.cc -- the primitives (senses and actions) you find
   called in the lap script files.  These mostly just call other libraries
   of functions in other files...

   $Log: em-prims.cc,v $
   Revision 1.5  1997/11/15 15:36:05  jo
   the "final" version for the paper

   Revision 1.1  1997/11/03 23:34:20  jo
   Initial revision


*/

#include <stdio.h>
```

```
#include <stream.h>
#include "em.hh"                // c++ stuff, see asm_decs.h below
#include "brain-objs.hh"

#include "asm_mydecs.h"         // tyrrell's code, can't include twice!
#include "animal_defs.h"        // #defs for actions, directions

#include "em-dirs.hh"           // must be after tyrrell's defs

extern int action;

static char rcsid[] = "$Id: em-prims.cc,v 1.5 1997/11/15 15:36:05 jo Exp jo $";

/*

When you add a primitive --
   1) add it to this list so it can be identified by people
   2) add the code (don't seem to need declarations, just definitions)
   3) add it to initprims at the bottom, so it gets set up

Remember that if a primitive action does not return SUCCESS, the rest
of a fap will not execute!  FAILURE indicates something _very_ drastic
happened, which should abort the next element.  Actually, it may not
make sense to abort except on sensing!

Notice that things that don't change "action" won't result in a time
step being taken, so we can use perceptual actions as an abstraction
without incurring cost (see also em.cc)

*/

bool res; // commonly used...

// This was just my first test...

/*  ************  Basic Actions  ************ */
// # Since "life" has no goal, need an action that always fails
lapres nil () {
  return (FAILURE);}

lapres sleep () {
  action = SLEEPING;
  return (SUCCESS);
}

int getting_lost () {
  res = (variance > LIKELY)
    ? true : false;
  return((int) res);
}

int lost () {
  res = (variance > VERY_LIKELY)
    ? true : false;
  return((int) res);
}

// slight info on predator
int sniff_predator () {
```

```
  res = (max_p1_perc_stimulus > P1SNIFF * HALF
         || max_p2_perc_stimulus > P2SNIFF * HALF ||
         (max_irr_perc_stimulus > TRIVIAL))
    ? true : false;
  return((int) res);
}

// slight info on predator
int predator1 () {
  res = (max_p1_perc_stimulus > max_p2_perc_stimulus)
    ? true : false;
  return((int) res);
}

// more certain of predator
int see_predator() {
  res = (max_p1_perc_stimulus > P1SEE * HALF ||
         max_p2_perc_stimulus > P2SEE * HALF ||
         (max_irr_perc_stimulus > LIKELY))
    ? true : false;
  return((int) res);
}

// action = look_around if biggest trace of predator is in our square
// -- assume that doesn’t take careful discrimination to sort out and
// we’re just confused
lapres observe_predator () {
  action = Directions::observe_predator();
  return (SUCCESS);
}


int at_home() {
  res = (den_dir == SAME_SQUARE && max_den_perc_stimulus > CERTAIN)
    ? true : false;
  return((int) res);
}

// you can’t get sheltered from cats or hoofs...
int covered() {
  res = (shelter_perc_stim[SAME_SQUARE] > VERY_LIKELY  &&
         max_p1_perc_stimulus < TRIVIAL && max_irr_perc_stimulus < TRIVIAL)
    ? true : false;
  return((int) res);
}

lapres hold_still () {
  action = FREEZING;
  return (SUCCESS);
}

lapres pick_safe_dir () {
  Directions::pick_safe();
  return(SUCCESS);
}

lapres go_fast () {
  action = Directions::go_fast();
  return(SUCCESS);
```

```
}

lapres go () {
  action = Directions::go();
  if (action == RESTING && cleanliness < LIKELY)
    action = CLEANING;
  return(SUCCESS);
}

lapres look_around () {
  action = LOOKING_AROUND;
  return(SUCCESS);
}

lapres groom () {
  action = CLEANING;
  return(SUCCESS);
}

// should prob. not be the same as TRIVIAL, TWEAK! (but value is cubed!)
int late () {
  res = (night_prox > 0.75) ? true : false;
  return((int) res);
}

lapres go_home () {
  action = Directions::go_home();
  return(SUCCESS);
}

int day_time () {
  res = (night_prox < 0.90) ? true : false;
  return((int) res);
}

// mating will fail if health is < .1, am I being conservative?  TWEAK
// NB this was never really calibrated, it just got ignored
int fit_to_mate () {
  res = (animal_health > 0.2 && carbo_minus < 0.8 && water_minus < 0.8 &&
         fat_minus < 0.8 && protein_minus < 0.8  && an_temp_plus < 0.8 &&
         cleanliness > 0.3) ? true : false;
  return((int) res);
}

// TWEAK -- I don't understand the code for this -- don't think need to
// check if mate's in square, but not sure about courtedness value
int courted_mate_here() {
  res = (mate_courtedness > TRIVIAL) ? true : false;
  return((int) res);
}

lapres copulate () {
  action = MATING;
  return(SUCCESS);
}

int mate_here() {
  res = (mate_dir == SAME_SQUARE && max_mate_perc_stimulus > TRIVIAL2 * HALF)
    ? true : false;
```

```
  return((int) res);
}

lapres strut () {
  action = COURT_DISPLAYING;
  return(SUCCESS);
}

// should probably play with this parameter TWEAK!
int sniff_mate() {
  res = (max_mate_perc_stimulus > TRIVIAL * HALF)
    ? true : false;
  return((int) res);
}

int see_mate() {
  res = (max_mate_perc_stimulus > LIKELY * HALF)
    ? true : false;
  return((int) res);
}

lapres pick_mate_dir () {
  Directions::pick_mate();
  return(SUCCESS);
}

lapres sit_or_clean () {
  if (cleanliness < VERY_LIKELY)
    action = CLEANING;
  else
    action = RESTING;
  return(SUCCESS);
}

int safish() {
  res = (shelter_perc_stim[SAME_SQUARE] > LIKELY)     ? true : false;
  return((int) res);
}

// A bunch of cases...  more specific stuff below!
int needed_resource_available() {
  static double food_need;
  // don’t hang out if going to get trodden
  if (max_irr_perc_stimulus > TRIVIAL)
    return ((int) false);
  food_need = carbo_minus + protein_minus + fat_minus / 3;
  res = (( food_perc_stim[SAME_SQUARE] > TRIVIAL && food_need > LIKELY) ||
         (water_perc_stim[SAME_SQUARE] > TRIVIAL && water_minus > LIKELY) ||
         (an_temp_minus > TRIVIAL && sqr_temp_plus > TRIVIAL) ||
         (an_temp_plus > TRIVIAL && sqr_temp_minus > TRIVIAL) ||
         (((cleanliness < VERY_LIKELY || animal_health < LIKELY) &&
           shelter_perc_stim[SAME_SQUARE] > LIKELY)))     ? true : false;
  return((int) res);
}

// Similar needed_resource_available! assumes they’ve been found true
// (pick_dir also is a bit like this)
lapres exploit_resource() {
  static double food_need;
```

```
  food_need = carbo_minus + protein_minus + fat_minus / 3;

  //most transient && biggest reward...
  if (prey_in_animal_square && animal_health > 0.1)
    action = POUNCING;
  else {
    if (food_perc_stim[SAME_SQUARE] > TRIVIAL  && food_need > LIKELY) {
      if (cf_in_animal_square)
        action = EATING_CF;
      else
        action = EATING_FF;
    }
    else if (water_minus > LIKELY && water_perc_stim[SAME_SQUARE] > TRIVIAL)
      action = DRINKING;
    else if ((an_temp_minus && sqr_temp_minus) ||
             shelter_perc_stim[SAME_SQUARE] < TRIVIAL) {
      // move to get warm! (or if exposed)
      Directions::pick();  action = Directions::go_fast();
    }
    else if (cleanliness < CERTAIN)
      action = CLEANING;
    else
      action = RESTING;
  } // else not pouncing

  return(SUCCESS);
} // exploit_resource


// hungry or thirsty, actually.  May want to latch this like in
// real animals -- variable that sets higher than it releases.
int hungry() {
  static double food_need;
  food_need = carbo_minus + protein_minus + fat_minus / 3;
  res = ((food_need > LIKELY) || (water_minus > LIKELY));
  return((int) res);
}

int cold() {
  res = (an_temp_minus > LIKELY);
  return((int) res);
}

int hot() {
  res = ((an_temp_plus > LIKELY));
  return((int) res);
}

int cool_spot() {
  res = ((sqr_temp_minus > TRIVIAL));
  return((int) res);
}

int dirty() {
  res = (cleanliness < VERY_LIKELY);
  return((int) res);
}

// there isn't really an independent tired factor...
```

```
int tired() {
  res = (animal_health < TRIVIAL);
  return((int) res);
}

lapres pick_sheltered_dir () {
  Directions::pick_sheltered();
  return(SUCCESS);
}

lapres pick_new_sheltered_dir () {
  Directions::pick_new_sheltered();
  return(SUCCESS);
}

lapres pick_forage_dir () {
  Directions::pick_forage();
  return(SUCCESS);
}

lapres pick_dir () {
  Directions::pick();
  return(SUCCESS);
}

lapres rest () {
  action = RESTING;
  return(SUCCESS);
}

/*---------*/


void
initprims() {
  // basic life
  Act::add_act("nil", nil);
  Act::add_act("sleep",sleep);
  Act::add_act("observe_predator",observe_predator);
  Act::add_act("hold_still",hold_still);
  Act::add_act("pick_safe_dir", pick_safe_dir);
  Act::add_act("go_fast",go_fast);
  Act::add_act("go",go);
  Act::add_act("groom",groom);
  Act::add_act("look_around",look_around);
  Act::add_act("go_home",go_home);
  Act::add_act("copulate",copulate);
  Act::add_act("strut",strut);
  Act::add_act("pick_mate_dir", pick_mate_dir);
  Act::add_act("sit_or_clean", sit_or_clean);
  Act::add_act("exploit_resource", exploit_resource);
  Act::add_act("pick_sheltered_dir",pick_sheltered_dir);
  Act::add_act("pick_new_sheltered_dir",pick_new_sheltered_dir);
  Act::add_act("pick_forage_dir",pick_forage_dir);
  Act::add_act("pick_dir",pick_dir);
  Act::add_act("rest", rest);
  Sense::add_sense("getting_lost",getting_lost);
  Sense::add_sense("lost",lost);
  Sense::add_sense("predator1",predator1);
```

```
  Sense::add_sense("see_predator",see_predator);
  Sense::add_sense("sniff_predator",sniff_predator);
  Sense::add_sense("at_home",at_home);
  Sense::add_sense("covered",covered);
  Sense::add_sense("late",late);
  Sense::add_sense("day_time", day_time);
  Sense::add_sense("fit_to_mate", fit_to_mate);
  Sense::add_sense("courted_mate_here", courted_mate_here);
  Sense::add_sense("mate_here", mate_here);
  Sense::add_sense("sniff_mate", sniff_mate);
  Sense::add_sense("see_mate", see_mate);
  Sense::add_sense("needed_resource_available", needed_resource_available);
  Sense::add_sense("hungry", hungry);
  Sense::add_sense("cold", cold);
  Sense::add_sense("hot", hot);
  Sense::add_sense("cool_spot", cool_spot);
  Sense::add_sense("dirty", dirty);
  Sense::add_sense("tired", tired);
  Sense::add_sense("safish", safish);
} // initprims
```

A.3 The Behaviour

This is the only behaviour I required for the Tyrrell Simulated Environment task beyond those provided by his thesis. It is for choosing the direction to go when the animal is moving.

```
class Directions {
public:
  Directions(){};
  ˜Directions(){};

  // class
  static int choice() {return(picked);}
  inline static void pick_safe();
  inline static void pick_sheltered();
  inline static void pick_new_sheltered();
  inline static void pick_forage();
  inline static int go_home();
  inline static void pick_mate();
  inline static void pick();
  inline static int go_fast();
  inline static int go();
  inline static int observe_predator();

private:
  static int picked;     // the next direction

  static int opposite[9];
  static int opleft[9], opright[9];
  static int left[9], right[9];
  inline static double dangerof(int direction);
}; // class Directions
```

```
// go the safest aprox. of opposite the most scary predator
// (we assume this was called to flee)  sets the variable "picked"
inline void Directions::pick_safe() {
  static int bad;
  static double op, or, ol;

  // small chance but...
  if (den_dir == SAME_SQUARE && max_den_perc_stimulus > CERTAIN) {
    picked = SAME_SQUARE; return;}

  if (max_p2_perc_stimulus > max_p1_perc_stimulus)
    bad = p2_dir;
  else
    bad = p1_dir;
  op = dangerof(opposite[bad]);
  or = dangerof(opright[bad]);
  ol = dangerof(opleft[bad]);
  if (op > or) {
    if (or > ol)
      picked = opleft[bad];
    else
      picked = opright[bad];
  } else {
    if (op > ol)
      picked = opleft[bad];
    else
      picked = opposite[bad];
  }
} // Directions::pick_safe


// go the safest aprox. of home -- assume nothing to run away
// from (we assume then we'd be fleeing)  sets the variable "picked"
inline int Directions::go_home() {
  static double dval, lval, rval;
  int den;

  if (max_den_perc_stimulus > TRIVIAL ||
      (max_den_perc_stimulus > max_den_memory_stimulus))
    den = den_dir;
  else
    den = remembered_den_dir;

  dval = dangerof(den) - shelter_perc_stim[den];
  rval = dangerof(right[den]) - shelter_perc_stim[den];
  lval = dangerof(left[den]) - shelter_perc_stim[den];
  if (dval > rval) {
    if (rval > lval)
      picked = left[den];
    else
      picked = right[den];
  } else {
    if (dval > lval)
      picked = left[den];
    else
      picked = den;
  }

  // we wouldn't have called this if we were home, so if we aren't
```

```
  // moving our perception is off.
  if (picked == SAME_SQUARE)
    return (LOOKING_AROUND);
  else
    return (go());

} // Directions::go_home

// go the safest aprox. of nearest mate -- assume nothing to run away
// from (we assume then we'd be fleeing)  sets the variable "picked"
// don't take a "big" risk here -- TWEAK
inline void Directions::pick_mate() {
  static double mval, lval, rval;

  mval = dangerof(mate_dir);
  if (mval < TRIVIAL)
    picked = mate_dir;
  else {
    rval = dangerof(right[mate_dir]);
    lval = dangerof(left[mate_dir]);
    if (lval > rval && rval < TRIVIAL)
      picked = right[mate_dir];
    else if (lval < TRIVIAL)
      picked = left[mate_dir];
    else
      pick(); // try to stay out of the way of moving animals...
  }
} // Directions::pick_mate


// just combine stuff -- world doesn't look that hard that we have
// to be very clever, so take tyrrell's word for the combined solution stuff
// (sort of!)  note no mate -- we handle them seperately
inline void Directions::pick() {
  static int iii, max_dir;
  static double food_need, max, res;
  food_need = carbo_minus + protein_minus + fat_minus / 3;

  max = 0; max_dir = 0;
  // start from current square to avoid oscillating
  for (iii = 0; iii < 9; iii++)
    if (dangerof(iii) < TRIVIAL * AVOID) { // TWEAK
      res =  food_perc_stim[iii] * food_need * 1.1
        + water_perc_stim[iii] * water_minus
        + mate_perc_stim[iii]
        - an_temp_minus * shade_perc_stim[iii]
        + an_temp_plus  * shade_perc_stim[iii]
        +shelter_perc_stim[iii] * shelter_perc_stim[iii] * SHELTER // TWEAK
        // can't do anything interesting at home!
        - den_perc_stim[iii] * den_perc_stim[iii];
      if (res > max) {
        max = res;
        max_dir = iii;
      }
    } // if danger (for iii)
  picked = max_dir; // will rest if surrounded by danger
} // Directions::pick()
```

```
// like pick but bias towards shelter and mate
inline void Directions::pick_sheltered() {
  static int iii, max_dir;
  static double food_need, max, res;
  food_need = carbo_minus + protein_minus + fat_minus / 3;

  max = 0; max_dir = 0;
  // start from current square to avoid oscillating
  for (iii = 0; iii < 9; iii++)
    if (dangerof(iii) < TRIVIAL * AVOID) { // TWEAK
      res = mate_perc_stim[iii] +
        +shelter_perc_stim[iii]
        - den_perc_stim[iii];
      if (res > max) {
        max = res;
        max_dir = iii;
      }
    } // if danger (for iii)
  picked = max_dir; // will rest if surrounded by danger
} // Directions::pick_sheltered()

// like pick-sheltered, but ignore present square (want to move)
inline void Directions::pick_new_sheltered() {
  static int iii, max_dir;
  static double food_need, max, res;
  food_need = carbo_minus + protein_minus + fat_minus / 3;

  max = 0; max_dir = 0;
  // start from current square to avoid oscillating
  for (iii = 1; iii < 9; iii++)
    if (dangerof(iii) < TRIVIAL * AVOID) { // TWEAK
      res = mate_perc_stim[iii] +
        +shelter_perc_stim[iii]
        - den_perc_stim[iii];
      if (res > max) {
        max = res;
        max_dir = iii;
      }
    } // if danger (for iii)
  picked = max_dir; // will rest if surrounded by danger
} // Directions::pick_new_sheltered()


// like pick but food or water sought
inline void Directions::pick_forage() {
  static int iii, max_dir;
  static double food_need, max, res;
  food_need = carbo_minus + protein_minus + fat_minus / 3;

  max = 0; max_dir = 0;
  // start from current square to avoid oscillating
  for (iii = 0; iii < 9; iii++)
    if (dangerof(iii) < TRIVIAL * AVOID) { // TWEAK
      res =  food_perc_stim[iii] * food_need * 1.1
        + water_perc_stim[iii] * water_minus
        +shelter_perc_stim[iii] * shelter_perc_stim[iii] * SHELTER // TWEAK
        // can’t do anything interesting at home!
        - den_perc_stim[iii] * den_perc_stim[iii];
      if (res > max) {
```

```
        max = res;
        max_dir = iii;
      }
    } // if danger (for iii)
  picked = max_dir; // will rest if surrounded by danger
} // Directions::pick_forage()


// sum all the scary things (maybe scale these things later as TWEAK)
inline double Directions::dangerof(int dir) {
  return (p2_perc_stim[dir] + p1_perc_stim[dir] +
          irr_perc_stim[dir] + edge_perc_stim[dir] +
          dp_perc_stim[dir] - den_perc_stim[dir]);
} // Directions::dangerof

inline int Directions::go_fast() {
  switch(picked) {
  case (N): return(MOVE2_N);
  case (NE): return(MOVE2_NE);
  case (E): return(MOVE2_E);
  case (SE): return(MOVE2_SE);
  case (S): return(MOVE2_S);
  case (SW): return(MOVE2_SW);
  case (W): return(MOVE2_W);
  case (NW): return(MOVE2_NW);
  default:
    return (FREEZING);  // could throw error...
  } // switch
} // Directions::go_fast()

inline int Directions::go() {
  switch(picked) {
  case (N): return(MOVE_N);
  case (NE): return(MOVE_NE);
  case (E): return(MOVE_E);
  case (SE): return(MOVE_SE);
  case (S): return(MOVE_S);
  case (SW): return(MOVE_SW);
  case (W): return(MOVE_W);
  case (NW): return(MOVE_NW);
  default:
    return (RESTING);  // could throw error...
  } // switch
} // Directions::go()

inline int Directions::observe_predator() {
  static int dir;

  dir = (max_p2_perc_stimulus < max_p1_perc_stimulus) ? p1_dir : p2_dir;
  switch(dir) {
  case (N): return(LOOKING_N);
  case (NE): return(LOOKING_NE);
  case (E): return(LOOKING_E);
  case (SE): return(LOOKING_SE);
  case (S): return(LOOKING_S);
  case (SW): return(LOOKING_SW);
  case (W): return(LOOKING_W);
  case (NW): return(LOOKING_NW);
  default:
```

```
    return (LOOKING_AROUND);  // if there’s a predator here we’d know it!
  } // switch
} // Directions::go()
```

A.4 The Main Routines

The version of Edmund used in this thesis is in C++, but Tyrrell’s simulated environment was in C. These languages can link, but the main() function must be in the C++ code. Consequently I also had to make a minor modification of Tyrrell’s code – I changed his ”main” to be ”asmain”, which is called from here.

```
/* em.cc --- the routines tyrrell's simulated enviroment call

Joanna Bryson, Nov 1 1997

$Log: em.cc,v $
Revision 1.1  1997/11/03 23:34:20  jo
Initial revision


*/

#include <stdio.h>
#include <stream.h>
#include <GetOpt.h>
#include <signal.h>
#include "em.hh"                // c++ stuff, see asm_decs.h below
#include "brain-objs.hh"
#include "brain.def"            // define the static/class variables

#include "asm_defs.h"
#include "asm_mydecs.h"         // tyrrell's code, includes defs for this file
#include "animal_defs.h"        // #defs for actions, directions

#include "em-dirs.hh"
#include "em.def"

static char rcsid[] =  "$Id: em.cc,v 1.1 1997/11/03 23:34:20 jo Exp jo $";

char *LOG_PATH = "../emfiles";
char *IOF_PATH = "../emfiles";        // lap and landmark files

int GlobalTickClock;  // how we tell time... set here!

Drive * life;  // has to be a global since we aren't the main loop

double P1SEE, P2SEE, P1SNIFF, P2SNIFF, AVOID, SHELTER;
extern "C" void asmain (int argc, char * argv[]);
extern "C" double rnd (double range);

main (int argc, char *argv[])
{
  char *lapfilename = "intercourse.lap";  // argh, have to hard code!

  if (!(lapfilename)) {
      cerr << "Error:  in main, no lapfile provided (argument -l) \n";
    }


  // now we need to read in the LAPs...

  cerr << "\n LAP reading brains... ";
  initprims();                          // add the primitives (senses and acts)
  read_brains(lapfilename, IOF_PATH);  // does its own error checking

  cerr << "\n Control to slave. ";
  asmain(argc,argv);
}
```

```
void em_initialize() {

  // generate our parameters we're testing
  /*
     P1SEE = .55 + rnd(0.2);
     P2SEE = .55 + rnd(0.2);
     P1SNIFF = .5 + rnd(0.2);
     P2SNIFF = .4 + rnd(0.2);
     AVOID = .45 + rnd(0.2);
     SHELTER = .6 + rnd(0.2);
     */
  P1SEE = .44;
  P2SEE = .72;
  P1SNIFF = .51;
  P2SNIFF = .45;
  AVOID = .30;
  SHELTER = .78;

  cerr << "\n getting life... ";
  life = Drive::get_drive("life");
  life = (Drive *) life->instantiate();

  GlobalTickClock = 0;  // start time as well as life :-)

  cerr << "Starting up... life =>\n";

}  // em_initialize()


int em_select_action() {
  static int res;
  action = -1;

  GlobalTickClock++;
  while ( (action < 0) && ((res = life->execute()) == CONTINUE) );

  if (action >= 0)
    return (action);

  cerr << "Life ended!!!\n";             // should never happen...

  switch (res) {
  case SUCCESS:
    cout << "Yaaay, we won!\n"; break;
  case FAILURE:
    cout << "Boooo, we lost!\n"; break;
  default:
    cout << "Huh?";
  } // switch res
  exit (-4);
} // em_select_action()
```