GOAL PROCESSING IN AUTONOMOUS AGENTS

by

LUC BEAUDOIN

A thesis submitted to the
Faculty of Science
of the
University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
University of Birmingham
Birmingham B15 2TT
England

August 1994

Synopsis

The objective of this thesis is to elucidate goal processing in autonomous agents from a design-stance. A. Sloman's theory of autonomous agents is taken as a starting point (Sloman, 1987; Sloman, 1992b). An autonomous agent is one that is capable of using its limited resources to generate and manage its own sources of motivation. A wide array of relevant psychological and AI theories are reviewed, including theories of motivation, emotion, attention, and planning. A technical yet rich concept of goals as control states is expounded. Processes operating on goals are presented, including vigilational processes and management processes. Reasons for limitations on management parallelism are discussed. A broad design of an autonomous agent that is based on M. Georgeff's (1986) Procedural Reasoning System is presented. The agent is meant to operate in a microworld scenario. The strengths and weaknesses of both the design and the theory behind it are discussed. The thesis concludes with suggestions for studying both emotion ("perturbance") and pathologies of attention as consequences of autonomous goal processing.

"The problem is not that we do not know which theory is correct, but rather that we cannot construct any theory at all which explains the basic facts" (Power, 1979 p. 109)

"I think that when we are speculating about very complicated adaptive systems, such as the human brain and social systems, we should especially beware of oversimplification—I call such oversimplification “Ockham's lobotomy”. " (Good, 1971a p. 375)

To my wife, children, and parents

Acknowledgements

I wish to express my deep gratitude to Aaron Sloman for his energetic, consistent, and insightful involvement in my research.

Thanks to Claude Lamontagne for showing me the importance of Artificial Intelligence for psychology; for demonstrating to me the poverty of empiricism; and for making me aware of the British tradition in computational psychology.

Thanks to everyone who was involved in the Attention and Affect Project.

Special thanks to Margaret Boden and Mike Harris for useful suggestions and encouragement for future research.

My doctoral studies were funded by a Commonwealth Scholarship from The Association of Commonwealth Universities in the United Kingdom, and scholarships from The Fonds pour la formation de chercheurs et l'aide à la recherche, and the Natural Sciences and Engineering Research Council of Canada.

Thanks to Patty for waiting in Canada while I indulged in research in Britain.

Contents

Illustrations

Tables

Procedures

Equations

Chapter 1. Introduction

1.1 Human scenario

> In the following scenario, consider the tasks and abilities of a nursemaid in charge of four toddlers, Tommy, Dicky, Mary, and Chloe. One morning, under the nursemaid's supervision the four children are playing with toys. Mary decides that she wants to play with Dicky's toy. So she approaches him and yanks the object out of his hands. Dicky starts to sob, as he cries out "mine! mine!" The nursemaid realises that she ought to intervene: i.e., to take the toy away from Mary, give it back to Dicky, and explain to Mary that she ought not to take things away from others without their permission. This task is quite demanding because Dicky continues crying for a while and needs to be consoled, while Mary has a temper tantrum and also needs to be appeased. While this is happening, the nursemaid hears Tommy whining about juice he has spilt on himself, and demanding a new shirt. The nursemaid tells him that she will get to him in a few minutes and that he should be patient until then. Still, he persists in his complaints. In the afternoon, there is more trouble. As the nursemaid is reading to Mary, she notices that Tommy is standing on a kitchen chair, precariously leaning forward. The nursemaid hastily heads towards Tommy, fearing that he might fall. And, sure enough, the toddler tumbles off his seat. The nursemaid nervously attends to Tommy and surveys the damage while comforting the stunned child. Meanwhile there are fumes emanating from Chloe indicating that her diaper needs to be changed, but despite the distinctiveness of the evidence it will be a few minutes before the nursemaid notices Chloe's problem.

Fortunately, human life is not always as hectic as that of a nursemaid. Nevertheless, this little scenario does illustrate some important human capabilities, and the "motivational" processes that they evince. (We are focusing on the nursemaid, not the children.) While directing the planned activities of the day, the nursemaid is able to detect and respond to problems, dangers and opportunities as they arise, and to produce appropriate goals when faced with them. For instance, when Mary violates Dicky's rights, the nursemaid needs to produce a collection of goals including one to comfort Dicky, to instruct Mary, and to comfort her too. The nursemaid is able to prioritise and schedule goals that cannot be executed simultaneously. Thus she decides that cleaning Tommy's dirty shirt can wait until Dicky and Mary are sufficiently calm. Although very resourceful, the nursemaid is, of course, neither omniscient nor omnipotent. When she is involved in a crisis, she might fail to notice other problems (such as Chloe's diapers). The nursemaid might even have to abandon some of her goals (though this scenario did not illustrate this). This nursemaid scenario is referred to throughout the thesis, and a technical version of it is described.

The objective of this thesis is to elucidate goal processing in autonomous agents such as the nursemaid: to try to give an account of the functions, constraints, and kinds of goal processes, and to investigate the cognitive architectures that can support these processes. This objective is expounded in this chapter. Understanding this objective requires a preliminary notion of autonomous agency, which is given in the following section along with the objectives of the thesis. Design-based methodology is described in detail by A. Sloman (1993a) and summarised below. The introduction also summarises the accomplishments of the thesis, describes a technical version of the nursemaid scenario, and gives an overview of the thesis.

1.2 Requirements of autonomous agents and of the theory

It is assumed that an agent is autonomous to the extent that it is capable of producing its own objectives but has limited resources with which to satisfy them. Some of these objectives are "top-level", meaning that they are not ontogenetically derived as means to some end, such as through some planning process; or if they are derived, that they have achieved "functional autonomy" (Allport, 1961) in as much as the agent treats them as good in themselves. Similarly, some top-level goals are derived from an evolutionary process even though the agent treats them as non-derivative. There is a large and controversial literature on what are the "true" objectives of human life. For instance, Aristotle (1958) has argued that there is only one non-derivative goal in humans: happiness. For behaviourists, the objectives of behaviour (if any) are to seek reinforcement and avoid punishments. A few stimuli are innately reinforcing (or punishing); but most reinforcing (or punishing) stimuli have that status through association with other reinforcing stimuli. For Freud, the ego seeks a compromise between an id that works according to a "pleasure principle" and the superego that incorporates versions of parental values. There are many theories of the ends of action. This thesis is not concerned with specifying the innate objectives of human life. It merely assumes that an autonomous agent has some number of top-level goals and a greater number of derivative ones.

The word "autonomous" is used as a technical term, in order concisely to refer to a class of agents. There is a long history of debate concerning what autonomy "really" means. However, the current thesis is not meant to contribute to this debate. An arbitrary new term could have been used instead of "autonomy", but since this term has a colloquial meaning that is close to the one referred to here, it has been adopted. Normally one would not include the concept "resource-bounded" in one's definition of autonomy (for in principle an agent whose resources surpassed its desires might still be called autonomous). However, it is expedient to do so in this document since all the agents it discusses are resource-bounded in some sense (and "autonomous resource-bounded agents" is too wordy an expression for one that is used so frequently).

In order to explain goal processing in autonomous agents, one needs to understand what requirements they satisfy. Doing this is an objective of this thesis and should be read as a theoretical contribution, since the requirements are falsifiable, or in principle can be shown to be deficient in number or organisation. Requirements analysis is roughly analogous to the notion of "computational theory" discussed by D. Marr (1982). Here follows an overview of the requirements of autonomous agents.

As mentioned above autonomous agents have multiple sources of motivation. They do not merely have one top level goal. These sources of motivation will lead them to produce particular goals, either as means to some end, or as an instantiation of the motivational source. The sources of motivation can be triggered asynchronously to the agent's other mental processes. For example, the (top-level) goal to eat can be triggered asynchronously to one's process of planning how to get from one place to another. Triggering of motivational sources can either be through internal or external events. For example, if the nursemaid had a desire to eat, it might have been triggered by an internal event (a drop in her blood sugar levels) or an external one (e.g., seeing palatable food). The multiplicity of motivation implies that the agents have many different tasks that they must perform.

There are important temporal constraints acting on autonomous agents. They need asynchronously to be responsive to the very sources of motivation that they activate. That is, motive processes should be able to interrupt other process. For example, when the nursemaid produced a goal to comfort Dicky, this interrupted her process of reading to Mary. The agent needs to be able to discover, set, and meet deadlines for its goals. This implies that some of the algorithms that it uses should be "anytime algorithms" (Dean & Boddy, 1988; Dean & Wellman, 1991; Horvitz, 1987). An anytime algorithm is one that can produce a result the quality of which is a function of the time spent processing. S. Russell and E. Wefald (1991) distinguish between two kinds of anytime algorithms. A contract anytime algorithm is one which before it starts to execute is given an amount of time that it can use before it must produce a response, and arranges to produce the best solution that it can within this time frame (e.g., it might select a method that requires the specified amount of time). An interruptable anytime algorithm is one that can be interrupted as it is going and yet still emit a sensible response. Engineers have devised many anytime algorithms, but not all devices use them. For instance, a typical calculator is not interruptable—it either gives a response or it does not. Many chess playing computer programs use contract anytime algorithms—the user can set the amount of time which the machine uses to make its move. Anytime performance is a form of graceful degradation, or graceful adaptation. Further temporal constraints are discussed in the core of the thesis.

There are various limits in the resources that autonomous agents have with which to deal with their goals. In particular, their beliefs are incomplete and may contain errors. They have limited abilities to predict the consequences of actions. Their processors work at a finite (though possibly variable) speed and have a finite set of mechanisms (though this set might increase and diversify with time). They have limited external resources of all kinds (principally effectors, tools, etc.). Temporal constraints have already been noted.

The strategies of autonomous agents must be robust, in the sense that they must operate in a wide variety of settings under various constraints. Autonomous agents must be adaptable, in that if they do not immediately have strategies that they can apply to generate the right goals and satisfy those goals in a new environment, they can adapt their strategies at some level to function in the new environment. This implicates requirements for learning. However, although requirements of robustness and adaptability are important, they are not examined closely in this thesis.

As B. Hayes-Roth (1993) points out, autonomous agents have to deal with complex contextual conditions. That is, there are usually many variables that are relevant to the control of their behaviour, some of which are internal, some external, and some both.

As will become increasingly obvious throughout the thesis, autonomous agents integrate a wide range of capabilities. Thus the computational architectures that model autonomous agents will be "broad" (Bates, Loyall, & Reilly, 1991). Many architectural components are active simultaneously, implying parallelism at a coarse grained level. For example, their perceptual mechanisms operate in parallel with motor processes, and processes that trigger sources of motivation (e.g., new goals) and that deal with the sources of motivation (e.g., planning processes).

There are many other requirements besides those listed here that can be derived from them e.g., the importance of directing belief revision as a function of the utility of inferences produced (Cawsey, Galliers, Logan, Reece, & Jones, 1993). The requirements are expanded in Ch 4. Other requirements will not be addressed here, such as social communication with others. Some of these other requirements will be easier to study once theories account for the main requirements.

An increasing number of researchers in computational psychology and Artificial Intelligence are addressing the requirements of autonomous agents (though usually in isolation). It is therefore a very exciting time to be performing research in this area. The requirements do not appear to be very controversial; however, it is not clear that everyone realises the difficulty of explaining how the requirements could be met (let alone how they are actually met by humans). (For more on requirements, see Boden, 1972; Hayes-Roth, 1990; Hayes-Roth, 1992; Oatley, 1992; Simon, 1967; Sloman, 1985a; Sloman, 1987).

1.3 Methodology—the design-based approach in context

The foregoing discussion of "requirements" and "architectures", as well as the title of the thesis foreshadowed the current section, in which the design-based approach is described and contrasted with related methodologies.

Much has been written about the different ways to conduct science. Cognitive science is a particularly rich area in that many methodologies are used. Here the taxonomy of methodologies related by Sloman (1993a) is given. Phenomena-based research proceeds either in a positivist or falsificationist (Popper, 1959) manner by collecting empirical data which either support or refute theories. In cognitive science, these data are supposed to shed light on cognitive systems through correlational or causal links between observable states, processes, and events. See (Keppel, 1982) for prescriptions concerning empirical research methodology. Phenomena-based research is mainly concerned with the "actual" rather than what is possible or necessary. In contrast, the current thesis does not present new phenomena-based research. However, in order to specify what needs to be explained, it occasionally refers to fairly obvious facts about humans (as opposed to very detailed empirical findings). Historically, institutional psychology (including theoretical psychology and cognitive psychology) has almost exclusively been concerned with empirical research (Green, 1994).

There is also semantics-based research in which scientists study concepts and relations between them. This involves techniques of "conceptual analysis" used chiefly (but not only) by philosophers. (See Sloman, 1978 Ch. 4; Warnock, 1989). For example, A. Ortony, G. L. Clore, and M. A. Foss (1987) have analysed the concept of emotion, and proposed a taxonomy of emotion concepts. Psychologists and linguists often carry out a related kind of research in which they try to specify what people actually mean by colloquial terms. Conceptual analysis can use empirical data about what people mean by terms as a starting point, but not as a final criterion for the validity of their analyses. Analysing concepts can be useful in the design-based approach, as well. In Ch. 3 some of the results of a conceptual analysis of goals are presented.

The design-based approach, used chiefly in AI, involves taking an engineering scientist methodology for studying real or possible systems. It has five main steps some of which can be executed recursively or in parallel. (1) Specify the requirements of the system in question. That is, what capabilities does or should the system have? What are its tasks, and why does it have them? A ply of requirements analysis of autonomous agents was presented in the previous section. This is extended throughout the thesis. (2) Propose designs which can satisfy the requirements. A design comprises an architecture and its mechanisms. An architecture comprises modules (components) that have causal links between them (e.g., data transmission, control, inhibition, etc.) The architecture need not be described at a physical level, i.e. its components can exist in a virtual machine. (3)

Implement designs (which can be prototype designs) in a computer simulation or in hardware. This helps to uncover lacunas and inconsistencies in a theory. (4) Analyse how, and the extent to which, the design meets the requirements, and how the simulation embodies the design. The analysis can be both mathematical and based on experimental tests of the implementation. (5) Study the space of possible designs surrounding the proposed model: How could the model have been different? What are the trade-offs that are implicated in the design? How would slight changes in the requirement impact on the design? What further capabilities could the system have if its design were slightly different? A complete understanding of a design requires that one can characterise it in relation to other designs in the space of possible designs (Sloman, 1984; Sloman, 1993a; Sloman, 1994c).

Although the design-based approach is distinguished from the phenomena-based methodologies, that does not imply that it cannot yield theories about humans (or other species). In fact quite the contrary is true, for in order to understand how individuals of some species really operate, one needs to have cogent theories about how they could operate. In other words, one can only understand actual systems through reference to possible systems (if a model could not possibly be implemented to satisfy the requirements, then it cannot empirically be correct). The kind of autonomous agency studied here involves such a sophisticated set of capabilities that it will take many years (perhaps centuries) before we have plausible working conjectures about how they can be realised. Once we have such theories, we will be in a good position to suggest an empirical theory, and then try to refute it. This is not to say, however, that phenomena-based research is useless. There is a need for many different types of research to be pursued in parallel, with some interaction between them.

There are many different ways in which design-based research can be conducted. See Sloman (1993a) for a number of variables. One dimension of variation of research is the "breadth" of the requirements that are studied and of the architectures that are proposed. Most research in cognitive science focuses on a very narrow set of capabilities, such as how visual perception of motion is possible, how one can identify speakers solely on the basis of acoustic input, what is responsible for spatial Stroop effects, etc. These questions can lead to the production of very detailed models. Even someone who is interested in autonomous agents does not necessarily try to provide a broad picture of the agents (e.g., she can focus on one of the requirements, such as time dependent planning). In this thesis, however, a very broad set of capabilities is addressed (compare previous section). This makes the task more difficult, and implies that the solutions that are proposed will be more sketchy for a longer period of time. J. Bates, A. B. Loyall, and W. S. Reilly (1991) have suggested a useful way of representing the distinction between the resultant architectures. Some will be very narrow (looking at a very specific task) but very deep (giving plenty of detail about the mechanisms underlying the task). Others will be very broad, but shallow. In practice, depth and breadth are traded-off. Of course, ultimately broad and deep architectures are most desirable.

This section has briefly expounded the design-stance not for the purpose of convincingly defending it—that would require more space than is available—but in order to set the framework for the rest of this thesis, which can be read as a case study in design-based methodology.

1.4 Summary of the contributions of the thesis and the importance of its objectives

I have approached the objectives of this thesis by applying and improving an existing theory of motive processing in autonomous agents proposed by Sloman in various publications. Through conceptual analysis and design exploration, this thesis directly builds upon Sloman's work, and it relates it to other theories. In this research I have

- systematically addressed the issue of how goals are processed in autonomous agents from a design-based perspective.
- collected and reviewed a number of theories from a wide range of research areas that bear on the issue of autonomous agency. These theories had never been considered together before. I have shown how these theories contribute pieces to the puzzle of autonomous agency, and how they can benefit from one another;
- further elaborated requirements for autonomous agents;
- provided a conceptual analysis of goals that views them as rich control structures with a variety of attributes and dimensions. This analysis generalises and clarifies previous work;
- proposed a new taxonomy of goal processes that distinguishes between vigilational processes and management processes;
- described important unsolved problems in the control of goal processes;
- proposed new concepts, terminology, and conceptual distinctions, e.g., "busyness", "management" processes, "deciding" goals, "generactivation", "surfacing", "criticality", and a distinction between the intentional and propensity interpretations of insistence;
- addressed the question, "Can some processing limitations be shown to be useful or necessary design features?"
- analysed, adapted, and improved a promising extant architecture for autonomous agents, (Georgeff & Ingrand, 1989);
- analysed the proposed architecture's strengths and weaknesses, thereby setting the scene for future research;

- made a number of specific proposals for new research following on the work in this thesis;
- indicated a conceptual resemblance between emotion (as "perturbance") and a psychopathology (obsessive compulsive disorder).

Contributions such as these stand as progress towards a deeper understanding of goal processing in autonomous agents. Such an understanding is extremely important for theoretical and engineering reasons. It will help to explain human motive processing mechanisms, by situating them within a space of possible designs. A deep understanding of goal processing should help to explain emotion-like phenomena which are referred to as "perturbance" (cf. Ch. 3 and 7). An understanding of normal goal processing should also help to characterise pathologies of goal processing and attention, such as are supposed to occur in affective and anxiety disorders (American Psychiatric Association, 1987). It is hoped that this understanding, in turn, will help to propose intervention schemes to deal with such disorders, as well as with less severe problems. Finally, one will be in a better position to build autonomous systems (robots, programs, etc.) that can take on progressively more responsibilities (e.g., security systems, unmanned space craft systems, emergency response systems). However, these benefits will only fully be reaped after many iterations of the slow and difficult cycles of design-based, semantic, and empirical research.

1.5 The technical nursemaid scenario

The human nursemaid scenario described above is useful for expounding the problems of autonomous agency. However, in order eventually to give an account of a human nursemaid (or any other human autonomous agent) first one needs to design models of simpler agents—as research progresses, the models will become increasingly sophisticated. For this reason, a technical version of the nursemaid scenario has been developed. (Hereafter, this is referred to as the "nursemaid scenario" or simply "the scenario".) The scenario was originally proposed by Sloman (1986), and was adapted for this thesis (Beaudoin & Sloman, 1991; Beaudoin, 1991). The scenario was created to require of an agent capabilities that are similar (at some level of abstraction) to human—autonomous—agents while ignoring other problems that are best left to other researchers, including 3-D vision, motor control, and naive physics. Hence the agent faces multiple (sometimes independent) problems that can occur and develop in overlapping time intervals and that need to be detected currently with and asynchronously to the agent's other activities. The problems differ in their urgency and importance profiles. Some problems get worse at a faster rate than others. Some problems have terminal urgency, others do not. Some problems only have derivative importance; whereas others are intrinsically aversive and some states are intrinsically good. (If the agent could learn, some of the things that were extrinsically aversive could become intrinsically aversive to it, and similarly for the good things.) However, the domain is biased in that there is an over-representation of aversive sources of motivation in relation to positive sources. The agent's cognitive and physical behaviour execute in parallel. The agent's perceptual focus is limited, and hence (unlike a typical program playing chess) it does not know all of the facts about its world. Many events in this world are unpredictable from the agent's perspective.

The "physics" and "psychology" of the domain can be extended indefinitely as required for testing later more complex versions of the theory.

The scenario is intended for a computer simulation, not primarily a robot implementation. The scenario involves a "robot" nursemaid whose function is to care for "robot" babies that roam around in a nursery, preventing problems and responding to them when they occur. Babies arrive at intervals, have to be protected from various dangers, and can eventually be discharged when they have reached a certain age. To discharge its function, the nursemaid has a single camera that can see a limited portion of the nursery at a time, and it has a claw with which it can pick up and transport one baby at a time. (For pragmatic reasons, it is assumed that the nursemaid's computer exists outside the nursery, and that it has remote control of its claw and camera, which can be moved independently.)

The nursery comprises a set of rectangular rooms separated by walls and connected by open doors. The rooms are bounded by deadly ditches. One of the rooms contains a recharge point, another an infirmary machine, and another a baby dismissal point. The claw and babies are considered as shapeless points. (See Figure 1.1).

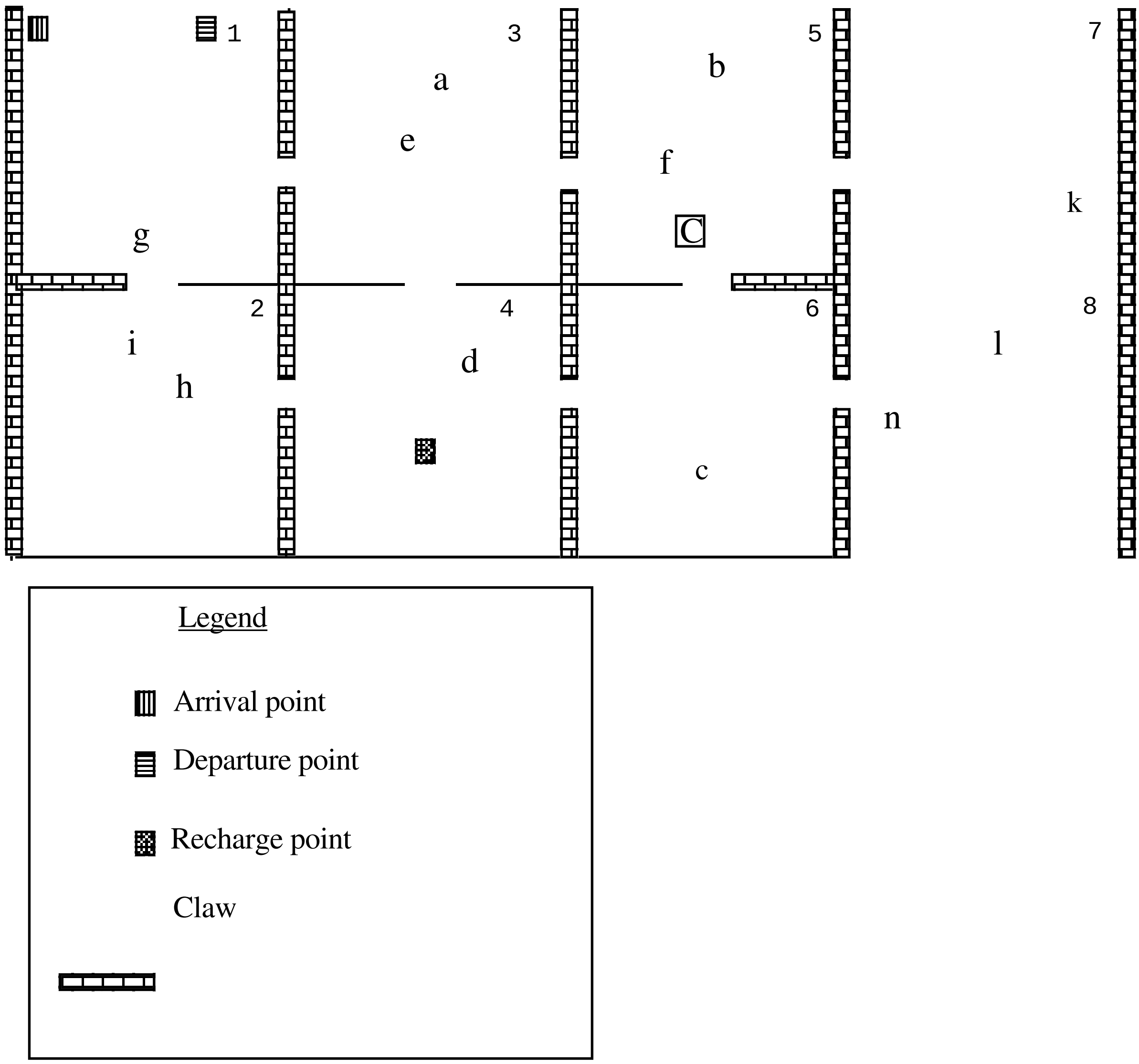

Figure 1.1. The Nursery. Room numbers are given in the upper right corners of the rooms.

There is a variety of problems and other contingencies to which the nursemaid must respond. Babies grow older, and when they reach a certain age or, if they die, they need to be removed from the nursery by being brought through the dismissal point. Babies die if they fall into ditches. Therefore, the nursemaid needs to keep them away from ditches. Babies also die if their battery charge expires; therefore, the nursemaid needs to recharge them in due course. It can do this by connecting the babies to the recharge point. Babies can also die if they contract certain illnesses. Ill or injured babies can be healed at the infirmary. Babies cannot all be put in the same room; for if the population density surpasses a certain threshold in one room, then the likelihood that some babies become thugs increases. Thug babies tend to attack and injure others. Thug babies should be isolated in order for them to lose their malicious tendencies. New babies can arrive in the nursery. Dead babies emit a magnetic field that can corrupt the memories of other babies; babies with corrupt memories can die. Therefore, it is important to dismiss dead babies. Corrupt memory is the only fatal "illness"; however, it can be cured in the infirmary. Other diseases that can develop are: the "shakes", and the "melts". They are all intrinsically bad; the shakes are cyclical, whereas the melts get monotonically worse. Injuries are not fatal. They can be incurred to either arm or leg of a baby, or to the head. The domain also affords a number of potential opportunities (e.g., in some circumstances it will be possible to solve two problems at once, or prevent a problem at a lower cost than at other junctures). Thus, there are a variety of potential problems and opportunities that are causally related, and have temporal patterns.

There are a few domain rules which the nursemaid should follow in case of a conflict between its goals. If the nursemaid can only save one of two babies, it should prefer faster babies to slower ones, healthier babies to less healthy ones, older babies to younger ones, and innocent ones to those that have been thugs. But since it should preserve the health and well being of as many babies as possible, if the nursemaid has to choose between saving two low value babies and one high level value, it should save the former. Notice that the domain does not explicitly quantitatively specify values for outcomes, instead the required preferences are stated in terms of rules and partial orders. There is no objective notion of "utility" (compare the discussion of utility in Ch 6). The given preferences are not sufficiently extensive for the nursemaid (or a human being, for that matter) to be able to infer for every pair of outcomes which one is preferable. This is so even when the outcomes are completely known. The designer of the nursemaid must invent a more specific decision-making scheme. (It would be insightful to observe the kinds of preferences that a human being playing a game version of the nursemaid scenario would invent.) This "invention" will not be totally arbitrary, since there are causal relations amongst problems and objective constraints in the domain, and there are some preference rules which in practice will usually preserve the objective domain preference rules. As an example of a potentially useful rule which the nursemaid could follow is that isolating a thug is usually more pressing than fixing the babies which it has injured. This is because injuries are intrinsically bad, and the longer a thug is on the loose, the more injuries are likely to occur. There is a potential for the rate of injuries caused by the thug to be greater than the rate at which they are fixed in the infirmary; however, this depends on parameters of the domain, such as the speed of travel of the thugs, the number of hits that are required for an injury, the frequency with which thugs tend to attack babies. Therefore, this rule can be invalidated if the parameters change. Moreover, the rule breaks down in some situations, e.g., if all the other babies in the room are dead.

The main task is to design the nursemaid. This is not a study of multiple co-operating and communicating intelligent agents. That is, the babies are considered as very simple automata, whereas the nursemaid is supposed to be a proper autonomous agent. The nursemaid requires cognitive abilities for detecting, prioritising, resolving problems, etc., according to the requirements described in Section 1.1. A design of a nursemaid (called "NML1") is given in Ch. 5. Prototype computer simulations were performed to help improve the design, but the final design was not implemented by the author, although Ian Wright of the University of Birmingham is implementing his design. The implementations are not reported here.

The domain is not designed to have any political, social, or economic significance. It is simply meant to embody a set of high level design requirements of autonomous agents. Furthermore, it can be extended in order to test the proposed design and better show how the design ought to be improved. For instance, one could require that the nursemaid needs to be recharged too, give the nursemaid auditory-like perception (to hear babies screaming, or judge population densities on the basis of wave-forms), allow robot "ogres" to snatch babies, give the nursemaid additional claws, or replace the babies by workers in a workshop factory.

1.5.1 A scenario in the nursemaid domain

In a typical scenario, the initial state of which is depicted in Figure 1.1, the nursemaid detects that babyA has a low charge. Having no other pressing problem to solve, the nursemaid decides to recharge it. As it is moving its claw toward babyA, the nursemaid notices that babyB is perilously close to a ditch. It decides that it had better interrupt its current endeavour and rescue babyB. As it starts to execute its plan to rescue babyB, it perceives babyC which is now sick; however, with the two other problems demanding attention, the nursemaid fails to "realise" that there is a problem with babyC. Later, babyC dies of its fatal illness.

A model of how the nursemaid's behaviour in this scenario could be achieved is given in Ch. 5.

1.6 Overview of the thesis

The thesis provides a literature review, a conceptual analysis of goals, a process specification of goals, an architecture for goal processing, a critique of the architecture, and a conclusion which outlines future research.

Chapter 2 reviews relevant psychological and AI theories. The thesis objectives implicate a very wide range of theories, which themselves involve a broad range of psychological functions. The review is necessarily selective. One theory from each of four areas of psychology is reviewed. In the area of goal theory, which examines psychometric factors involving goals for predicting behaviour, the theory of Thomas Lee and Edwin Locke is examined. In the area of emotion, Keith Oatley & Philip Johnson-Laird's Communication theory is selected. This is classified as an "autonomy theory of emotion". Richard Shiffren & Walter Schneider's theory of attention is reviewed. From the AI literature, Robert Wilensky's model of multiple motive agency is presented. Two AI models of autonomous agency are also reviewed: B. Hayes-Roth's Adaptive Intelligence System and M.

Georgeff's Procedural Reasoning System. Each model contributes something useful and can benefit from the others.

Chapter 3 expounds the concept of goal in terms of "control states". The conceptual structure of goals is presented. Nearly a dozen features of goals are analysed, including their importance, rationale, insistence, commitment-status, and intensity. The use of intentional terminology, such as "beliefs" and "goals", is quite controversial. It is therefore important to justify the mechanistic interpretation of these terms. One of the most persuasive anti-mechanistic views on the issue—D. Dennett's "intentional stance"—is summarised and criticised.

Chapter 4 gives a process specification of goals. A distinction between "high level" management processes and "lower level" vigilational processes is drawn, the functions of these categories are described, and the categories are subdivided. The state-transitions of goals are very flexible—this raises the issue of how to control them. Sloman's notion of insistence based goal filtering is explained as a vigilational function. A distinction is drawn between two interpretations of insistence: an intentional interpretation and a propensity interpretation. New functions for filtering are supposed. Part of the rationale for filtering is that there is a limit to the amount of concurrency that management processes can accommodate. This assumption is discussed. The process specification contributes to the requirements of autonomous agents.

Chapter 5 describes a design of a nursemaid, called "NML1", which will display some (but not all) of the processes described in Ch. 4, and which will be a procedural reasoning system. The architecture assumes a number of modules that execute in parallel (though some of them are partly synchronised), including goal generactivators, insistence filters, an interpreter, a collection of management processes, and perceptual and effector devices. Algorithms for some of these modules are presented, but further research is required to explore a wider variety of algorithms, and better select amongst them.

Chapter 6 presents a critical examination of NML1 and extant theory of autonomous agents. It describes the strengths and weaknesses of the design, and points at areas where more research is needed. It is suggested that an autonomous agent should separate its problem description from its goals, and be capable of representing valenced information. Some of the difficulties with reasoning about procedures are identified. The need for theories to help design mechanisms for controlling management processing is identified. There is also a need for a qualitative theory of decision-making, given a criticism of utility-based decision-making.

Chapter 7 concludes the thesis by summarising it, situating it within the context of a broader project concerned with Attention and Affect, and suggesting fruitful areas for future research.

Chapter 2. Literature Review

There is an enormous amount of disparate literature in psychology and AI that is potentially relevant to the topic of goal processing in autonomous agents. Thousands of articles have been published on the topics of motivation, emotion, "self-regulation", and attention. Rarely are these topics considered together. Affective processes are rarely considered by cognitive psychologists; however, when they are, the cognitive psychologists are usually concerned with the effects of these processes on "cognition" (e.g., as biasing decision-making, or speed of information processing), but affective processes are often not considered as cognitive processes or information processes themselves. In the AI literature, goal processing has been examined, but usually does not use the terms of motivation, emotion, self-regulation, and attention. There are, of course, exceptions to this rule, (e.g. Boden, 1972; Simon, 1967; Sloman, 1978; Sloman & Croucher, 1981). It is fitting for a thesis on broad architectures to take a look at a broad spectrum of research.

Although many areas of research are examined here, only one theory per area will be considered. This survey has three main objectives. One is to demonstrate that many leading theories in different areas can benefit from each other: each has strengths that are lacking in the others. The second is to indicate good ideas to build upon, and pitfalls or problems to overcome (e.g., limitations of existing designs). The third is to illustrate the design-based approach to evaluating psychological and AI literature on autonomous agents.

The first part of this chapter examines some psychological literature. The second part examines AI literature on autonomous agents. The conclusion shows how the various theories complement one another but do not provide a complete account of autonomous goal processing. Later chapters attempt to integrate the contributions—but it will be years before such an integration is complete.

2.1 Psychology

Four main areas of psychological research are reviewed. Firstly, a theory of motivation based on the notion of "goal setting" is presented. Secondly, a category of theories of emotion is described as viewing emotion as a consequence of requirements of autonomous agents. The communicative theory of affect of Keith Oatley and Philip Johnson-Laird, which is a member of this category, is discussed. Thirdly, two theories which divide mental capabilities into attentional and automatic processes are discussed—namely the theory of Walter Schneider, Susan T. Dumais, and Richard M. Shiffrin, and the theory of Donald Norman and Tim Shallice.

2.1.1 Goal theory of motivation

The social sciences have witnessed a proliferation of research on the determinants of goals, and the impact of goals on behaviour (Bandura, 1989; Lee, Locke, & Latham, 1989). These determinants are seen as "factors". An important goal of this research has been to determine what factors there are, and the correlations and causal relations amongst them. However, the ultimate aim of this research is to be able to predict and control performance on the basis of motivational measurements. (In contrast, the current thesis is concerned with explaining possibilities. (See Ch. 2 of Sloman, 1978). The theories that are proposed differ slightly in their definition of factors, and in the exact relations that are hypothesised to hold amongst the variables.

In this section, the "goal setting" theory of T. W. Lee, E. A. Locke, and G. P. Latham (1989) is discussed. The discussion aims to underscore some of the contributions of goal theory, and to distinguish goal theory from the theory proposed in this thesis. Although, goal theory is a phenomena-based theory, it is discussed here without direct reference to the empirical research that led to its postulates.

Goal theory is supposed to provide a "specification of goal processes". The theory emphasises the positive effect on performance of an individual "setting" specific and difficult goals. The main assumptions are that (1) the content of a goal determines the mental profile of behaviour towards the goal, and this profile in turn impacts on performance; (2) these causal relations are subject to moderating influences, as described below. Goals have four components: (1) The goal level is the difficulty of the state to be achieved. For instance, a student might aim to be in the top 5th or 10th percentile—the higher the percentile, the higher the goal level. (2) There is the degree of quantitative specificity of the goal. For example, the goal to be within the 10th percentile is more specific than the goal to "do well academically". (The fact that the authors focus on quantitative specificity may be due to a general prejudice against qualitative formulae in science, for in principle qualitative objectives can be just as precise as quantitative ones.) (3) There is the "complexity" of the goal; by this they mean the number of subgoals that are required to satisfy it. (The term "complexity" as used here is slightly misleading, because whereas they say that it is a predicate of goals their concept pertains to plans. In fact, a goal may be non-complex (in the logical sense) while triggering a complex plan, and vice-versa. A more adequate notion of complexity is proposed in Section 3.2.2.) (4) There is the conflict between a target goals and other goals. Goal level and goal specificity are assumed to combine additively to affect the behaviour profile. However, goal specificity only has its effect if the goal level is high. The "complexity" of the goal and goal conflict negatively affect the behaviour profile. Goal specificity is assumed to affect the profile of behaviour so long as the goal is difficult.

The behaviour profile comprises direction of effort. This simply means that behaviour is selectively directed towards the goal. There is a quantitative dimension of amount of effort, and one of persistence in the face of external difficulties. And "task strategy" represents the plans that are used to execute the task.

There is a moderating factor between goal content and behaviour profile: goal commitment. (See Hollenbeck & Klein, 1987) for a competing model of goal commitment). That is, no matter how "complex" or difficult the goal, the agent will only work for it if he is committed to it. Although this variable is assumed on empirical grounds, it is apparent that there is a conceptual constraint operating here as it is part of the logic of the colloquial concept of not being committed to a goal that one will not work towards it. (Admittedly, commitment and intention are slippery constructs that have been hotly contested at least since the ancient times (Aristotle, 1958). In particular, there is a sense in which one can be "not committed" to a goal whilst pursuing it.) Goal commitment is said to be affected by a number of factors: the legitimacy of the authority of the person who sets the goals (the authors are interested in social settings where goals trickle down an organisational hierarchy); peer and group pressures; expectancy that one's endeavours will be successful; the extent of one's perceived general and task specific self-efficacy; the value of the goal and its instrumentality (in achieving super-goals).

A collection of other variables is proposed which affect the link between the behaviour profile and performance, such as the person's knowledge, feedback, tools available, etc. It is noteworthy that these constructs are represented quantitatively.

Characterising goal theoretic research is useful in order to put the present thesis in a distinctive context. Goal theory underscores a number of variables that need to be considered in goal processing. Some of these factors are social (e.g., peer and group pressures) and the present research will ignore them because it is believed that before characterising social agency one needs to characterise non-social agency. There are some people who believe that intelligence, intentionality, consciousness, etc. are only possible for social agents, but this is contrary to the author's assumptions. Goal theory also usefully describes motivation as a multi-dimensional phenomenon—motivation is not simply the amount of effort that a person is willing to exert for a goal. A similar tenet is expressed in the following chapter. The factors of goal theory, however, are specified at a high level, and not in information processing terms. Thus the notion of "processes" that is used is different from the engineering notion used in this thesis—e.g., the "process" diagrams of Lee et al. (1989) are not state-transition diagrams or petri graphs. Lee et al. (1989) do not model the states of goals as low and high level decisions are taken about them. Their "process specifications" are really specifications of statistical relations between variables (or factors). This is useful for Lee et al. (Lee, et al., 1989), to the extent that they can measure the variables, manipulate them, and thereby exercise some control over behaviour. Furthermore, the "processes" are not meant to be embodied in a computational architecture, let alone an architecture that has to solve problems in an environment. Moreover, all of the variables that are considered by goal theory are quantitative, whereas in a design-based framework many of these "variables" would actually translate into mechanisms or structured data, such as descriptions of states to be achieved or prevented. For example, there would be a mechanism for feedback rather than just a quantitative causal link. Furthermore, the theory does not provide an account of multiple goal processing. In Chapter 3, a concept of goal is proposed that is richer than the one presented here.

2.1.2 Autonomy theories of emotion[1]

In the very early years of AI and computational psychology, affect (e.g., motivation and emotions) was a prominent area of investigation (e.g., Taylor, 1960; Tomkins, 1963; Toda, 1962; see Boden 1972, 1987 for reviews). Perhaps because of the difficulty of the task, interest in affect waned in the 1960's and 1970's. However, since circa 1985 affect has been studied by a growing number of computationally minded scientists. Whether by coincidence or not, this growth coincides with the growing interest in autonomous agents in AI. Thus there are a number of theories of emotion that claim that emotions are a consequence of the requirements of autonomous agency: i.e., in order to design an agent which meets these requirements, evolution (or any other designer) must produce a system with emotion producing mechanisms (e.g. Frijda, 1986; Oatley & Johnson-Laird, 1987; Simon, 1967; Sloman & Croucher, 1981; Toda, 1962). Some of these theories are "functionalist" in the sense that they view emotions either as being a process or system (Frijda, 1986) or as a state resulting from a special purpose system specifically designed to deal with emotional situations (Oatley & Johnson-Laird, to appear); others are afunctionalist (Sloman & Croucher, 1981) in that they view emotions as being an emergent property of a system made of components each of which has a function, but none of which is specifically designed to produce an emotional state. (The issue of functionalism is briefly discussed in Ch. 7.)

Rather than review the whole literature, this section focuses on one theory, the communicative theory of emotions. Sloman's theory is briefly described in Ch. 7.

2.1.2.1 A communicative theory of emotion

Keith Oatley and Philip Johnson-Laird have proposed an empirical (but partly design-based) communicative theory of affect. The theory was originally published in (Oatley & Johnson-Laird, 1987) but it has recently been revised in (Oatley & Johnson-Laird, to appear). This theory stems from a recognition that autonomous agents need to be able globally to redirect attention when faced with

---

[1]These theories have not been thus categorised before. They are sometimes called "cognitive theories", but this is a different category, since not all cognitive theories are autonomy theories.

significant junctures regarding their plans, such as changes in probability of goal satisfaction. When an agent detects that a goal is "significantly" more likely to be achieved than it previously believed, this leads to a particular class of positive emotion (happiness). Decrease in probability of goal satisfaction leads to negative emotions. These emotions serve to communicate this change both between processors within the individual's mind, and between individuals. (Here we will ignore social requirements, however.) Emotional communication is supposed to be both rapid and usually effective. The communication leads to an interruption of processing and an adjustment in the system's plans.

The communicative theory assumes that the mind comprises a hierarchy of parallel processors where the parallelism is coarse, and not necessarily neural. See Johnson-Laird (1988 Part VI). At the highest level of the hierarchy there is a processor corresponding to "consciousness" which exercises control over the lower levels, and uses semantic messages as well as control messages. (The distinction between these terms is theirs, not mine. See Oatley (1992 Ch. 3). Semantic messages have specific addresses or referents, whereas control signals do not. Control signals propagate in parallel throughout the mind in a manner analogous to diffusion.

The theory assumes mechanisms that communicate control signals. There are two main aspects to this: one is the detection of control conditions, the other is the production of control actions in response to the control conditions. Some modules are concerned with the appraisal of events as being relevant to the system's goals. These mechanisms encode control conditions. For instance, one mechanism might detect that a situation implies that a plan is very likely to succeed. Each control condition has associated with it a distinct control action. The control actions are known as "emotion modes". When a control condition is detected, the detecting module sends a global broadcast throughout the system that affects many processors and thereby triggers an emotion mode. Each emotion mode is responsible for a distinct form of action readiness, cognitive organisation, and can provoke "conscious preoccupation" with the event that caused it.

The most recent version of the communicative theory assumes four innate basic types of emotion (i.e. control dispositions consisting of pairs of control conditions and control actions) (Oatley & Johnson-Laird, to appear). These four emotions can be actualised without the agent knowing their cause or without them having a particular object. The authors claim that the emotion modes involve a broadcast of control signals that are devoid of semantic content. These basic emotions are as follows.

1. Happiness is generated when subgoals are being achieved. The control action is to continue the plan in question, modifying it if necessary.

2. Sadness occurs when there is a failure of a major plan toward a goal, or an active goal needs to be abandoned. The control action here leads to a search for a new plan.

3. Fear occurs when a self-preservation goal is threatened, or when there is a goal conflict. The control action is to arrest the current plan, pay attention, freeze, and/or flee.

4. Anger occurs at the juncture where an active plan meets with some interference. (This overly-broad assumption is criticised in the next subsection.) The control action is to work harder, and/or attack.

Besides these four basic emotions, there are five other innate control states that necessarily involve a semantic object: attachment, parental love, sexual attraction, disgust, and rejection.

The communicative theory supposes that these control states usually inhibit each other but occasionally can be active in parallel. Complex evaluations of a situation can lead to the simultaneous activation of control states. "With loss of a loved one [one] may feel both sad at the loss and angry at those whose negligence was responsible for it" (Oatley & Johnson-Laird, to appear). In this circumstance, both happiness control signals and anger control signals are propagated in parallel throughout the mind. Thus, different action tendencies will be elicited.

2.1.2.1.1 Critique

The theory can be evaluated in relation to three different questions. (1) Can the class of systems it describes actually meet the requirements of autonomous agents? Or to what extent does it? The next question is most interesting if the first question is answered affirmatively. (2) Is there an empirical correspondence between the described system and what happens in virtual or physical machines within the human mind? (3) To what extent does the account map onto folk theory? Like Sloman (1988), Oatley and Johnson-Laird assume that emotional terms implicitly refer to internal mental states and processes. Unlike Sloman and Beaudoin, however, Oatley and Johnson-Laird are very interested in providing a theory that maps onto folk psychology. This is why they unabashedly use the terms they do. Of course, such a mapping will never be perfect, because there is so much variability (and continual evolution that is partly based on scientific theories) in usage of intentional idioms. Whether or not the theory gives an adequate account of "emotions" will partly depend on this folk psychological criterion—many accounts fail because they do not map onto what people think they mean by the terms. In contrast, we are content to introduce a new term instead of "emotion" (Cf. Chapters 3, 4, 7). It is nevertheless important to separate (2) and (3) because even if the theory fails on the third count, it may be successful on the second. In view of this, one could replace the terms "emotion", "happiness", etc. with technical analogues.

Oatley and Johnson-Laird (to appear) review data supporting the communicative theory according to empirical criteria (2) and (3). Although they claim it does well on both counts, it is weak on criterion (3). For example, their definition of "anger" does not restrict it to frustration due to the actions of a cognizant agent who should have known better (cf. Ortony, Clore, and Collins, 1988 Ch. 7.) From a purely empirical perspective (2) the communicative theory is one of the best cognitive theories of "emotions", given the variety of phenomena it encompasses. (Many of the empirical components of the theory were not described here.) From the design stance (1), the recognition of the requirement that an autonomous agent must be able to redirect its attention when faced with significant junctures in plans is important. Whether this always requires global signalling is a different question. The theory does provide principles that are worth investigating for designing an architecture. Although these principles have at least face validity and do seem plausible, it has not yet been demonstrated that the theory describes a design which can meet the difficult requirements of autonomous agency or be implemented. In particular, stronger analytical arguments are required to demonstrate that coherent shifts in behaviour can be achieved on the basis of a diffusion of control signals. A more specific design and implementation (e.g., of a nursemaid based on the theory) would be useful in this respect. This would require that such questions as "How are the processors to be designed?", "How many communication ports can a processor have?", "What specific examples of control messages are there?", and "Precisely how does a processor decide what do on the basis of control signals?" be addressed.

A burden of explanation lies on the notions of the top level processor and the lower level processors. However, even before proposing specific mechanisms for these modules, one ought to provide a more systematic analysis of the tasks of the system that is non-committal regarding which modules are selected to execute the tasks or how they do so. For example, whereas the communicative theory supposes that a process of "evaluation" is to be executed it seems that the concept of evaluation is complex and subtle in ways not reflected by the theory. There are different kinds of evaluation that ought to be distinguished systematically. For instance, in Ch. 3 different forms of evaluation of goals are expounded: e.g., concerning the importance, urgency, intensity, and insistence of goals. And each of these dimensions of assessment is itself complex: there are different kinds of importance, and different forms of urgency. Moreover, the dimensions of assessment can have separate effects on the internal or external behaviour of an agent. Once these tasks are clarified, it becomes possible to assign them to specific modules or interactions between modules.

Finally, the focus on a small set of junctures is also worth investigating. It is possible that there is a larger set of junctures to which an autonomous agent must be sensitive than the communicative theory posits. It would be useful for a handful of AI researchers who are unaware of the communicative theory to attempt to produce a taxonomy of plan junctures. What would they find? V. R. Lesser, J. Pavlin, and E. H. Durfee (1989) investigate a similar issue and propose six types of goal relationships. They suggest control actions that should be taken on the basis of these control conditions (junctures). These actions are solely concerned with increasing the efficiency of processing, whereas the communicative theory postulates a wider variety of actions. Still, it is left to future research to answer the above question and integrate the aforementioned theories.

When these issues have been addressed it may be possible to produce a model of a nursemaid which effectively processes goals in a manner that is consistent with the design principles of the communicative theory. In sum, the communicative theory fares well empirically, and is likely to generate useful design-based research.

2.1.3 Attention

In psychology the issue of internal resource boundedness has been studied in terms of limitations on "attention". Definitions of attention differ, but most of them imply the selection (or suppression) of information for (or from) higher order processing (Christ, 1991). Why is there a need to select some information? This is usually (but not always) said to be because there is one (or many) processor(s) that has (have) "limited capacity". With regard to these general issues, psychologists have asked many questions, some of which were fairly misguided[1], others insightful. Among the better questions are "What are the limits on contemporaneous mental processing?", "Which mental processes go on in series, which go in parallel?" and "What is the ordering of processes that are serial in relation to each other?" In order for these questions to be answered, models need to be proposed in which there are stages of information processing, and possibly parallel processes.

R. M. Shiffrin and W. Schneider (1984; 1977) provide a controversial explanation of a body of literature on attention. This model is dated, but it serves to illustrate the points concisely. They suggest that there are two qualitatively different sets of mental processes: automatic and controlled. Automatic processes are supposed to be quick, parallel, "effortless", and "uncontrollable"; and they do not use a capacity limited short term memory. In contrast, controlled processes are supposed to be slow, serial, "effortful", and largely controllable. Both processes are assumed to have their effects by varying the degree of activation of memory structures in a short term store. They argue that tasks can become automatic if they involve a consistent mapping between stimuli and responses, whereas if the mapping is variable then control is needed for successful execution. They explain Stroop interference (Stroop, 1935) by supposing that both colour identification (or spatial judgements) and reading are automatic processes that vary in speed. Hence, in order for the correct response to be given on incompatible trials, the faster, incompatible automatic process needs to be inhibited, which is something that requires "attention".

---

[1]For instance Allport (1989) demonstrates that the huge debate on whether "selection of information for attentional processing is early or late" is based on a conceptual muddle.

This kind of theory is appealing because it seems parsimoniously to map onto a distinction which is familiar in folk psychology. Most of us believe that there are things which we do "automatically" and things which require "attention". And, because of its simplicity, it appeals to scientists who value the parsimony edge of Occam's razor. Table 2.1 shows how Schneider et a. (1984) distinguish between 11 dimensions on the basis of their two-fold distinction.

Table 2.1

Some characteristics of Automatic and Control processes according to (Schneider, et al., 1984).

Characteristic	Automatic processes	Control processes
Central capacity	Not required	Required
Control	Not complete	Complete
Indivisibility	Holistic	Fragmented
Practice	Results in gradual improvement	Has little effect
Modification	Difficult	Easy
Seriality dependence	Parallel Independent	Serial Dependent
Storage in LTM	Little or none	Large amounts
Performance level	High	Low, except when task is simple
Simplicity	Irrelevant	Irrelevant
Awareness	Low	High
Attention	Not strictly required	Required
Effort	Minimal	Great

However, there are important criticisms to be made against the model, not the least of which is that it buys parsimony at the cost of blurring important distinctions. Three comments are in order.

The first concerns the complexity of attention. Although it is tempting to reduce mental phenomena to two distinct categories of dimensions, the reality of the situation is much more complex. Conceptual analysis reveals attention and automaticity to be polymorphous concepts (White, 1964), i.e., concepts that are multiply instantiated by different activities. Moreover, they have "neighbours" in conceptual space along dimensions that are not captured by the authors, such as the concepts of attending, noticing, realising, desiring and intending (White, 1964), and various failures thereof (Austin, 1968).

Even if sense could be made of attention in terms of control and automatic processes, J. D. Cohen, K. Dunbar (1990), and J. L. McClelland and G. D. Logan (1989) show that these processes are not as empirically distinct as is supposed. For instance, Logan reports that subjects in the Stroop paradigm who are informed of the probability of compatibility and incompatibility of information cancel out Stroop effects. This and other evidence is used by Logan to conclude that automatic processes can indeed be controlled by subjects.

The second comment concerns the supposed autonomy of control processing. A. Allport, E. A. Styles, and S. Hsieh have recently empirically criticized this model (and (Norman & Shallice, 1986)'s theory that there is a "supervisory attentional system" (or SAS)[1]). They take it as a defining feature of the controller that it is "autonomous". "An underlying theoretical distinction is made between a "controlled" system, that is essentially stimulus driven, and an autonomous system that does not depend on stimulus triggering." (p. 8). They find that the time to switch tasks is much higher when the switch occurs at a memorised pre-specified juncture than when a stimulus cue for the new task is given. From these and similar data they conclude that the controller is also triggered into action by sensory stimuli, and hence is not autonomous. However, one might counter that Allport and his colleagues are attacking a straw-man, since those who propose a controller do not really believe (or need not believe) that it is autonomous, only that it controls other parts of the system in which it is embedded. Indeed it would be foolish to believe that the controller does not itself respond to events in the world. Yet, unfortunately, there are some foundations to Allport's claim. For instance, Schneider et al. (1984) write:

> We suggest a two-part definition that is sufficient to establish the presence of a large class of automatic and control processes. It may be stated as follows:
> 1. Any process that does not use general, non-specific processing resources and does not decrease the general, non-specific capacity available for other processes is automatic.
> 2. Any process that demands resources in response to external stimulus inputs, regardless of subjects' attempts to ignore the distraction, is automatic. (p. 21).

In this passage, automatic processing is distinguished from control processing on the basis of its being "stimulus driven", or prompted by stimuli. Nevertheless, even if this clause is removed, Schneider's theory does not immediately crumble. However, if a distinguished expert on attention, such as Allport is, is wrong in believing that environmental autonomy is a critical feature of Schneider's controller or Norman and Shallice's SAS, then perhaps this is partly because the working of these modules is not specified clearly enough. In any case, as mentioned in the introduction, the definition of "autonomy" varies widely according to theoretical persuasion.

The third comment concerns the specification of the control module. Perhaps the main problem with these models is that we are given purported characteristics of control processing without being presented with a breakdown of its components. The controller is essentially a black box. It is said to be "unitary", but this does not make sense: it cannot perform its complex tasks if it does not have components. And those components are not sufficiently obvious from the specification that they can be omitted.

One might ask "Why are the controllers so poorly specified?" Perhaps this is because the authors think that there are few data to go on. (Allport and colleagues claim that there are few data to go on.) Rather than make detailed theories which are likely to be false, they prefer more abstract and

---

[1]The models of Schneider and Norman and Allport are different in many respects. However, they are treated together for the remainder of this section because they both distinguish between a black-box like supervisory (or control) mechanism and automatic mechanisms.

non-committal theories. However using a design-based approach allows one to overcome this difficulty by sketching the space of possible designs without (in the short run) committing oneself to any specific design as the one that is really representative of human nature.

There are other psychological theories that do break down the "control module" and in this respect they fare better than these theories of attention. Two examples are the theories of Nico Frijda (1987) and Julius Kuhl (1988). Indeed work in AI, to be reviewed in the next section, does even better in this respect.

Whereas it seems that almost all of the research on human attention has directly or indirectly tried to further our understanding of the nature of the constraints on human ability to process information, very few researchers have systematically tried to answer the question: What constraints should there be on a person's ability to process information? That is, one can ask "What purpose can limiting processing resources serve for an agent?" Trying to explain attention in these terms does not require that one make a "reification move" of proposing a module for attention. If attention is viewed as selective processing then it can be viewed as an aspect of processing rather than a module. Attempts to frame or answer analogous questions include (Allport, 1989; Boden, 1988 166-8; Heuer & Sanders, 1987; Simon, 1967; Sloman, 1978 pp. 138 and 251-2). In Ch. 4 a variant of this question is formulated as "What should be the constraints on an autonomous agent's ability to manage goals in parallel?" This question is not an empirical one and it cannot adequately be answered without reference to possible designs, design options, and environmental requirements. In order properly to answer that question, therefore, design options need to be proposed first. This is done in the next section and in Chapters 5 and 6.

2.2 AI and autonomous agents

Since the inception of AI, many architectures for problem solving and action have been produced. (For reviews see Boden, 1987 Ch. 12; Chapman, 1987; Cohen & Feigenbaum, 1982 Ch. XV; Georgeff, 1987). The space of possible designs is extremely large, and although many designs have been produced, only the tip of the iceberg has been studied. Producing design taxonomies is an important but arduous task.

Building agents that meet the requirements of autonomy has recently become a more prominent research goal of AI. The resultant systems are often called "reactive planners", because they are capable of directing their behaviour on the basis of intentions that might be long standing or recent. In this section, three of the main research projects in this area are reviewed. The first project is headed by Robert Wilensky. It focuses on systems with multiple goals. The second project, headed by B. Hayes-Roth, adapts a type of architecture that is prominent in AI, namely blackboard architectures, to the task of autonomous agents. The third project, headed by M. Georgeff, investigates systems that are based on procedural knowledge.

2.2.1 Wilensky on agents with multiple goals

Wilensky was one of the first people in AI to work specifically on systems with multiple top level goals (Wilensky, 1980). His is a distinguished contribution to AI, providing insightful requirements, scenarios, and an architecture. He notes that human-like agents need to be capable of generating their own goals, and that there are many conflicts which arise in systems with multiple goals such that the agents need to know how to notice and resolve them. Moreover autonomous agents have to form plans that solve many goals. (M. Pollack 1992 later referred to this as "overloading intentions".)

Wilensky proposes an architecture to meet these requirements. It has a Goal Detector[1] which generates goals on the basis of changes in the state of the world or through means-ends reasoning, or in order to solve a planning problem. (The latter alternative involves "meta-goals".) The Plan Generator suggests candidate plans for goals, and expands them to the point where they can be passed on to the executor. It has three components. (1) The Proposer: suggests possible plans. (2) The Projector predicts the effects of executing plans, and stores them in a Hypotheses database. Interestingly, goal detectors are sensitive not only to changes in the model of the world, but also to hypothetical changes in the world ensuing from plans. (3). The Revisor edits plans that might have problematic effects (as signalled by goal detectors responding to data in the Hypotheses databases). These can either be pre-stored plans or "fairly novel solutions" (Wilensky, 1990). The Executor carries out plans and detects execution errors.

The Projector can detect that two goals conflict, e.g., because the effects of a hypothetical plan to satisfy one goal interfere with another goal (which is not necessarily a means to a common top level goal). When the system detects a conflict, it will have to come to a decision that involves a meta-planning process (i.e., planning the planning). The Plan Generator has a number of meta-plans. (1) RE-PLAN involves trying to find a plan of action in which the goals can be satisfied without a conflict. However, it is not always possible to find one. (2) CHANGE-CIRCUMSTANCE is a meta-plan that involves changing the situation which led to the conflict. It is not always possible to eliminate a conflict between goals, so it is sometimes necessary to abandon a goal. The (3) SIMULATE-AND-SELECT meta-plan involves simulating courses of action that favour one goal or the other, and selecting between them. An attempt is made to violate both goals as little as possible. However, this raises an important theoretical question, "How to select amongst future states on some (not necessarily explicit) basis of value and certainty?" In Ch. 6 it is argued that existing theoretical

[1] This concept has had many names and close conceptual cousins, including "monitors" (Sloman, 1978), "motivator generators" (Sloman & Croucher, 1981), "opportunity analyzers" (Bratman, Israel, & Pollack, 1988), "relevance detectors" (Frijda, 1986).

principles for selecting amongst alternate actions and conflicting states are inadequate and need to be improved.

Although Wilensky's model relies on the ability to simulate the outcome of a plan, it does not have a theory for how this should be done. Moreover, it does not represent effects of possible actions in a temporal manner. This implies that it cannot characterise the time course of the importance of the effects of actions (hence that it cannot compute generalised urgency, described in Section 3.2.2.1). Nowadays, it is fashionable to argue that planning itself is intractable (Agre, 1988; Agre & Chapman, 1990). Some of the sceptics appeal to the frame problem, others to the recent argument that combinational planning is NP complete (Chapman, 1987). However, this claim actually is only demonstrated for certain classes of planners. It has not been demonstrated that it is impossible to produce a mechanism which heuristically proposes possible behaviours and heuristically predicts effects of these actions. Since Wilensky is not committed to any specific form of planning, his system is immune to the formal arguments. It will be necessary for these details to be spelt out in future work. My design also makes use of predictive abilities, but I do not have an adequate theory about how these abilities are realised.

Unfortunately, the requirements of autonomous agents are even more complicated than the ones Wilensky's model was designed to address. For instance, goals need to be generated and managed in interaction with dynamic environments the temporal features of which impose temporal constraints on the design. In particular, an architecture needs to be able to produce goals asynchronously to its other mental processes, and respond to them. Hence it needs to be able (1) to store (descriptions of) reasoning processes, (2) to interrupt these processes, (3) to resume these processes while being sensitive to changes that happened since they were last run (e.g., the basis for decisions and conclusions might be invalidated). This requires a much more sophisticated set of mechanisms than those used by contemporary computer operating systems—one cannot simply freeze a process descriptor at one moment and resume the next, expecting it to be still valid. Moreover, representation of time and truth maintenance also need to be considered. These requirements are not addressed by Wilensky's work. But one person cannot solve all of the problems. Recent developments in blackboard systems and procedural reasoning systems have looked at some of these other requirements.

2.2.2 Blackboard systems

In this section attention is directed at a class of architectures known as blackboard systems (BBSs)—a particular kind of rule-based system (Hayes-Roth, 1987). Systems labelled as BBSs admit great variety: there is probably no statement which applies to all blackboard systems and which distinguishes them from systems not labelled as such. (D. Corkill, 9 Jun 1993, makes a similar point.) For reviews of literature on blackboard systems the interested reader is referred to (Jagannathan, Dodhiawala, & Baum, 1989; Nii, 1986a; Nii, 1986b). The approach taken in this section is to focus on a particular lineage of blackboard systems proposed by B. Hayes-Roth. I start by discussing a standard blackboard system for problem solving. Then I explain why this system was not suitable for autonomous agency. Then I present a design that improved upon the former for the purpose of achieving autonomous behaviour.

It is worth noting that the problems addressed by Hayes-Roth as well as the methodology she uses are extremely similar to those of this thesis.

2.2.2.1 A standard blackboard system

A blackboard system developed by B. Hayes-Roth, the Dynamic Control Architecture (DCA) (Hayes-Roth, 1985), is worth discussing here, amongst other reasons, because (1) it is insightful to see the additions that need to be made to them to address the particular issues of autonomous agents; and (2) autonomous agents might have mechanisms in common with systems that do not have temporal or epistemic constraints. The DCA is quite complex and the following presentation is by necessity a simplification.

The DCA has a global database (known as the blackboard), procedures (known as Knowledge Sources), and a scheduler. Knowledge Sources have conditions of applicability which determine whether on any given cycle they should be considered as candidates for execution. The scheduler verifies which Knowledge Sources are applicable, creates Knowledge Source Activation Record (KSARs) out of applicable Knowledge Sources, rates every KSAR, finds the preferred KSAR on the basis of the current rating preference policy, and executes it. When a KSAR executes, it records its results on the blackboard.

Blackboard systems such as the DCA have the following features (1) they solve problems incrementally, in that solution elements are gradually added to the blackboard, and (2) their solutions implicate a parallel decomposition of the main task, in that multiple aspects of the problem can be worked on in an interleaved fashion (through the sequential interleaved execution of different KSARs), (3) they activate solution elements "opportunistically" when required; that is, Knowledge Sources can be executed when their conditions of activation are met (unless the scheduler decides not to choose them).

The use of non-interruptable KSARs and a global blackboard is an important feature of DCA. Since KSARs are not interruptable, there is no need for a KSAR to be aware of another's intermediate computations: from the perspective of one KSAR the execution of another is instantaneous. Thus the values of local variables of KSARs are not open for inspection by other processes. And this holds even if a KSAR incrementally adds values to the blackboard.

These features of opportunistic interleaved execution of tasks can be quite useful. By design, they allow the system to take advantage of computational opportunities, and hence potentially to use its reasoning time effectively. Moreover, its capacity to work on multiple tasks makes it possible to process multiple goals.

However, DCA in itself is not suitable as a model of autonomous problem solving. (Laffey et al. 1988 make a related point regarding AI systems in general.) Communication between KSARs makes use of a global database. This is tedious and can cause unwanted interactions: direct communication between KSARs is sometimes preferable. Moreover, if KSARs and the blackboard need to be implemented on a distributed architecture, then it will sometimes be faster for two adjacent KSARs to communicate directly than via a blackboard. The DCA is slow and not highly interruptable. This is partly because (1) the scheduler operates sequentially (and exhaustively); (2) there is single write-access to the blackboard, implying a communication bottle-neck; (3) KSARs execute in a non-interruptable fashion; (4) KSARs execute one at a time; (4) the scheduler reassesses each KSAR after the execution of every Knowledge Source. Even Hayes-Roth admits:

> Given today's hardware and operating systems, if one's goal is to build high performance application systems, the blackboard control architecture is probably inappropriate (Hayes-Roth, 1985 p. 299).

However in Hayes-Roth's view the performance problem with DCA is not mainly the lack of parallelism but its exhaustive scheduling. See Section 2.2.2.2. Like many models in AI, DCA can partly be judged on the basis of its improvableness. And in this respect it has faired well, evolving over the years with the demands of the times. For instance, in such an architecture, adding facilities for interruptability is not very difficult since by nature the focus of reasoning shifts dynamically. And there are many ways in which the standard architecture can be modified to allow parallel processing. (See Corkill, 1989, for a cogent exploration of the design options in the parallelisation of blackboard models.) DCA's scheduler has provisions for ordering KSARs on the basis of ratings; therefore, it is easy to modify it to include such values as urgency and importance ratings. Indeed, the model in the next section did this. Therefore, blackboard systems have not only benefited from improvements in hardware but also from design changes. This brings us to an exposition of a blackboard system designed specifically for requirements of autonomous agents.

2.2.2.2 AIS: A blackboard model of autonomous agents

B. Hayes-Roth has developed an "Adaptive Intelligent System" (AIS)[1] which is one of the most advanced autonomous agent architectures in the AI literature (Hayes-Roth, 1990; Hayes-Roth, 1992; Hayes-Roth, 1993; Hayes-Roth, Lalanda, Morignot, Pfleger, & Balabanovic, 1993). AIS has three modules executing in parallel, each one of which has an input-output buffer for communication with the others. The perception module takes in information from the environment, abstracts, filters, and annotates it. The action module receives commands for actions (in its input buffer) and "translates" it into sequences of commands. The cognition module performs all of the high level reasoning. It is the cognitive system that uses a blackboard architecture (adapted from the DCA).

The cognitive system has four main procedures which are cyclically executed. (1) A dynamic control planner edits a blackboard entry known as the "control plan" which contains decisions about the kinds of reasoning and domain tasks to perform, and how and when to do so. (2) An agenda manager, identifies and rates applicable KSARs. (3) A scheduler selects KSARs for execution simply based on their ratings and scheduling rules (e.g., most important first). (4) An executor simply runs the executable KSAR.

A few features distinguish AIS from its ancestors. A major difference lies in its sensitivity to temporal constraints. Its agenda manager (which prioritises KSARs) follows a "satisficing cycle" rather than an exhaustive one. That is, it keeps prioritising the KSARs until a termination condition is met. This condition is not necessarily that all KSARs have been evaluated, but can be that a certain amount of time has elapsed. When the condition has been met, the scheduler then selects the best candidate KSAR for execution, and passes it to the executor. This is an anytime algorithm (cf. Section 1.2). Moreover, it is capable of reflex behaviour. Perception-action arcs are not mediated by the cognitive system. Furthermore the control planner makes plans that are adjusted as a function of deadlines, which are computed dynamically. Guardian, an implementation of AIS (Hayes-Roth, et al., 1992), uses a novel anytime algorithm for responding to external problems (Ash, Gold, Seiver, & Hayes-Roth, 1992). This algorithm hierarchically refines its theory about the nature of a problem, i.e. its diagnosis. At any time in the process, it can suggest an action based on the current diagnosis. The system delays its response until the best diagnosis is produced, or until something indicates that an immediate response is necessary. (There is a general problem in the control of anytime algorithms concerning when a response should be demanded.) This algorithm could be used by other architectures (e.g., by PRS, which is described in Section 2.2.3).

---

[1] Hayes-Roth doesn't provide a consistent name for this design. Since she sometimes refers to it as an "Adaptive Intelligent System", that's what it is called here. A partial implementation of the design is called "Guardian" (Hayes-Roth, Washington, Ash, Hewett, Collinot, Vina, et al., 1992).

2.2.2.2.1 Assessment of AIS

Although there are ways to improve AIS, it is on the whole an advance on the state of the art in blackboard systems. It is worth noting that although AIS is not proposed as a model of human cognitive architectures, it fares well in its ability to meet the requirements of autonomous agents. Again, this is probably because of the tendency in the psychological literature to favour non-committal models over false ones. Moreover, AIS has been implemented.

There are some features of the design that could be improved upon.

- One could argue that greater efficiency can be obtained by increasing AIS's macro-parallelism. More specifically, one could improve its responsiveness if the cognitive system had multiple KSARs executing in parallel. The blackboard, which is identified as the major bottleneck of blackboard systems, in AIS is still a single write data structure. However, B. Hayes-Roth (1990) argues against the parallelisation of the cognitive system. She says:

> Although we have considered distributing cognitive tasks among parallel processes [...], our experience with Guardian suggests that cognitive tasks have many important interactions, including sequential constraints, and associated needs for communication. Operating on a single processor in the context of a single global data structure supports these interactions, so we would favour distribution of cognitive tasks only in a shared-memory architecture (p. 121)

B. Hayes-Roth's claim that cognitive tasks should not be parallelised because of their "important interactions" is unjustified: she does not provide arguments for it in the cited publications. Decker et al. (1991) and R. Bisiani and A. Forin (1989) have reported successful parallelisation of blackboard systems. Moreover, it appears that whereas some tasks have important interactions, others do not: e.g., there might be no important interaction between playing chess and verbally describing a previous event. It is easier to demonstrate lack of interaction in some well defined domains than in very abstract terms. However, the theoretically important issue of limits on cognitive parallelism is moot and deferred to Ch. 4.

- The modularity assumption that motor, sensory, and cognitive systems do not overlap, although popular, seems to be inconsistent with human cognition. For example, there is clear evidence that visual information does not merely output its results to a cognitive system, it is also part of a posture control mechanism involving effectors as well (Lee and Lishman, 1975). Moreover, for many purposes overlapping systems are more useful. Sloman (1989) discusses many examples of this.

- Sensitivity to temporal constraints is a critical requirement of the system. Although D. Ash, G. Gold, A. Seiver, and B. Hayes-Roth (1992) have presented an anytime algorithm, their notion of deadlines is too simple. A more general notion of urgency is required that considers graded "deadlines". (See Ch. 3.) Incorporating this might not require an architectural change.

- Although the blackboard data structures are quite rich—particularly the KSARs and the decisions (Hayes-Roth, 1985)—there are some important types of information about tasks (i.e., goals) that are not represented. For instance, there is no way to express that a task has been rejected. Also, it is not possible to express that the acceptance (not the scheduling) of a task or decision is conditional upon some proposition being true. For example, such an agent could not express "I'll only try to solve the problem of fixing this valve if I manage to solve the problem of fixing the two gauges." (It could probably do this if the tasks were generated synchronously as subgoals of a task, rather than asynchronously to planning on the basis of perceptual information.)

- The provisions for preventing the distraction of the agenda manager are meagre. This is especially problematic since the agenda manager follows a "satisficing cycle". If the number of KSARs is very high, then it could happen that important/urgent KSARs do not even get considered because the satisficing cycle terminates before they are rated. Hayes-Roth might respond that this is a price that needs to be paid in order to obtain quick responses. However, whereas it is true that the requirements preclude an optimal agenda manager, it would nevertheless be possible to decrease the risk by using additional attentional mechanisms, such as one which rates and orders the KSARs asynchronously to the rest of the cognitive operations, or that produces and uses heuristic measures of the ratings (e.g., "insistence" (Sloman & Croucher, 1981), as discussed below).

However, this is not to say that the architecture cannot be changed to allow for these improvements. Moreover, regardless of whether the architecture can be improved, it serves as a reference point in design space for autonomous agents. NML1 improves on some of AIS's shortcomings, although unfortunately it does not match AIS in every respect.

2.2.3 Procedural reasoning systems

M. Georgeff and his colleagues have developed a system to meet the requirements of autonomous agents. These researchers were particularly concerned with the system being able to change its intentions and goals rapidly as events unfold. The communicative theory has similar aims. It is worth discussing PRS in detail here because NML1 is based on PRS.

PRS is based on procedures (Georgeff & Lansky, 1986). Procedures are essentially plans, or instructions denoting sequences of actions that achieve a goal state or lead to some other state if the procedures fail or are aborted. Procedures have applicability conditions, which are expressions in a temporal logic. They can be executed when their conditions are met. They can be invoked either as subroutines or as top level responses to world or self knowledge. That is, their applicability conditions can be unified with goal expressions or beliefs. Procedures' instructions are either goal expressions or primitive actions. Procedures are executed by an interpreter that either causes the performance of the primitive action (if the next instruction is a primitive action) or pushes the goal on a goal stack and later selects a procedure whose conditions of applicability unifies with the goal. If many procedures are applicable to a goal then a meta-procedure is invoked to select amongst them. PRS goals can be complex temporal instructions, and they may include rich control structures such as conditionals, iterators, and recursive calls.

Procedures differ from KSARs (and productions) in many ways: e.g., some procedures are required to be active for long periods of time. Moreover, their execution can be interleaved with the interpreter's other activities, including the execution of other procedures, whereas KSARs execute uninterruptedly and typically during short periods of time.

Georgeff provides many justifications for basing a system on procedures, as he defines them. One purported advantage over combinational planning systems such as that of Wilkins (1985) is that procedures conveniently allow (quick) run time expansion. Moreover, Georgeff claims that procedural systems are better than production systems at encoding and executing solutions to problems:

> [...] much expert knowledge is already procedural in nature [...] In such cases it is highly disadvantageous to "deproceduralize" this knowledge into disjoint rules or descriptions of individual actions. To do so invariably involves encoding the control structure in some way. Usually this is done by linking individual actions with "control conditions," whose sole purpose is to ensure that the rules or actions are executed in the correct order. This approach can be very tedious and confusing, destroys extendibility, and lacks any natural semantics (Georgeff & Lansky, 1986 p. 1384)

The rule-based system is less efficient because it needs to include tests in more rules, whereas a procedural system can make assumptions about a procedure's context of execution based on the previous goals that must have been satisfied. This implies that sensing needs are relaxed in procedural systems. These are advantages that PRS has over the more recent system, AIS.

A few terminological issues need to be flagged. Georgeff refers to procedures as "knowledge areas". But this term is misleading since it suggests something whose function is primarily denotational rather than operational. He refers to active procedures as intentions rather than processes. In this thesis, the term "knowledge area" is not used, but procedures are distinguished from processes. And the term "procedure activation record" is used to refer to the information about an active procedure. (This is standard computer science terminology, and also used in blackboard systems.) The concept of a procedure activation record is elaborated in Ch. 5. Georgeff refers to goals as behaviours; but this is confusing, especially since procedures are also referred to as behaviours. In this thesis, goals are not behaviours.

2.2.3.1 The PRS architecture

Procedures cannot execute outside an architecture. Georgeff provides a PRS architecture with a view to meeting the requirements of autonomous agents (Georgeff & Ingrand, 1989; Georgeff & Lansky, 1986; Georgeff & Lansky, 1987; Georgeff, Lansky, & Schoppers, 1987). The PRS architecture has an internal and an external component. (See Figure 2.1.) The external component is made of sensors, a monitor, effectors and a command generator. The internal component has a number of modules. Procedures are stored in a procedure library. Facts about the world or the system are either built-in or produced either by processes (i.e., procedure activations) or the monitor and are stored in the database. The monitor translates sensor information into database facts. Goals can either be generated as subgoals by processes or by the user. PRS does not allow goals to be triggered directly by beliefs in the database. Goals are stored in the goals structure. The process structure is a list of process stacks. A process stack is a stack of procedure activation records. (Processes can be active, unadopted, or conditionally suspended.) Each process stack occupies a particular slot of the process structure. An interpreter selects procedures for execution and pushes procedure activation records on, and removes them from, the appropriate process stacks on the process structure. The command generator translates atomic efferent procedure instructions into commands usable by effectors.

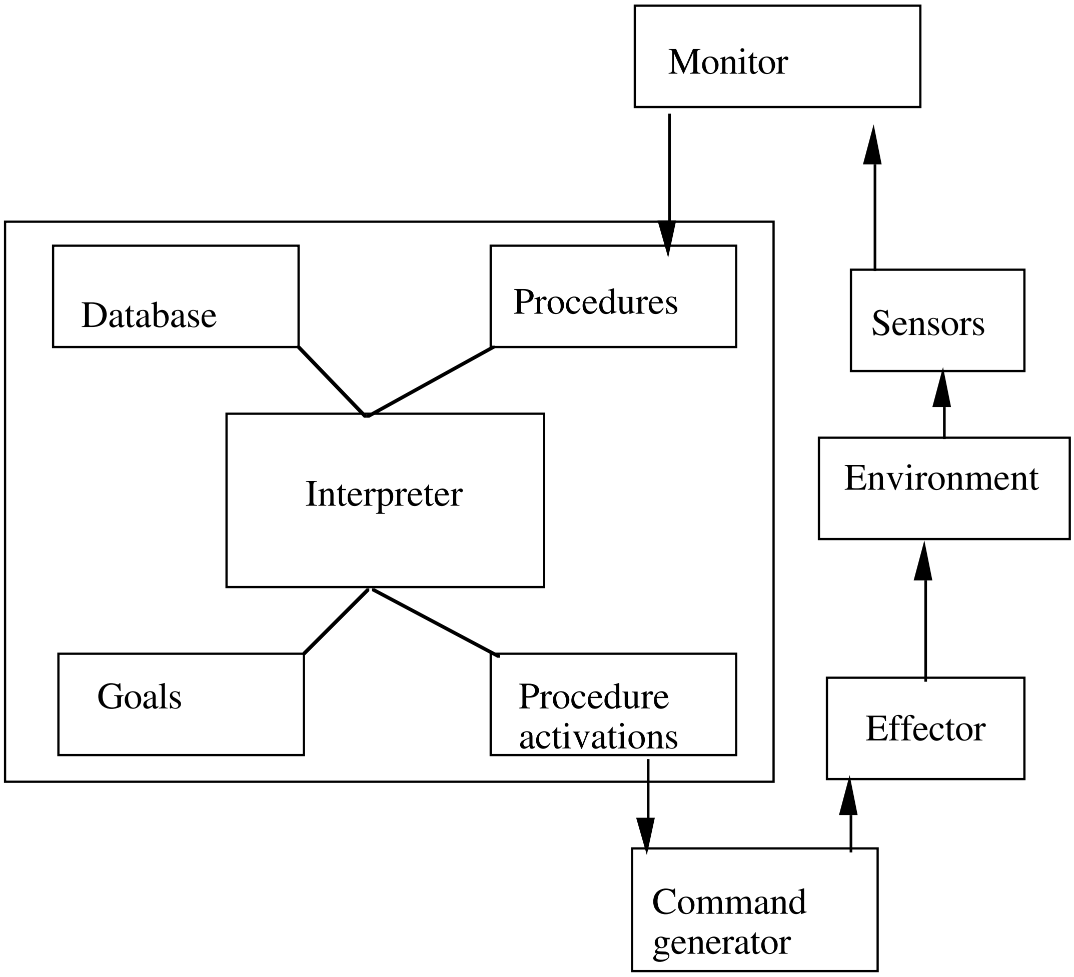

Figure 2.1. Georgeff's Procedural Reasoning System.

The interpreter runs PRS. It goes through the following cycle when new facts are asserted or new goals appear. (1) It runs through the procedure library verifying for each procedure whether it is applicable to the new facts or goals. Conditions of applicability are stored in the procedures and they can refer to facts in the world and/or current goals. If a procedure applies to a fact rather than a goal, then a new process record is created and the procedure is put at the root of the process record's process stack. If only one procedure is applicable to a given goal, this procedure is put on top of the process stack that pushed the goal. If more than one procedure is applicable to a goal, the interpreter invokes a meta-process to select amongst these procedures; if more than one meta-process is applicable, a meta-process will be selected to select amongst meta-processes, and so on recursively (compare Sloman, 1978, Ch. 6); otherwise, a new process record is formed and put in the process structure. Having determined and selected applicable procedures, the interpreter moves on to the next step. (2) The interpreter selects a process for execution. (3) The interpreter executes the selected process until a relevant asynchronous event occurs. This processing works as follows. The interpreter reads the next instruction from the procedure activation record on top of the selected process's process stack. (3.1) If this instruction indicates that an atomic action is required, then this action is performed (this can involve modifying goals or beliefs, or sending an instruction to the command generator.) (3.2) Otherwise the next instruction specifies a goal; in this case this goal is pushed[1] on top of the process's goal stack. The appearance of this goal will cause the interpreter to go to step 1.

One of the main features of PRS is that it does not need fully to expand a plan for a goal (or in response to a fact) when the goal (or fact) arises. PRS only needs to select a procedure, and this procedure will have many nodes that need only be traversed (and possibly expanded) at run time.

2.2.3.2 Assessment of PRS

PRS is a general and promising architecture. It supports shifting of attention, changing intentions and suspending and resuming physical and mental processes. It allows for planning and execution to be performed in an interleaved or parallel fashion. Its use of procedures simplifies control (as noted above). The fact that procedures can be selected in whole without being totally expanded before run-time is useful because, of course, it is often impossible before run time to have a very detailed plan (usually because necessary information is not accessible before run time). However, one of the problems concerning PRS, which will be described in Ch. 6, is that, conversely, it is not possible for it to expand a procedure's sub-goals until they are to be executed. Procedures allow for a task level decomposition, whose virtues have been expounded by Brooks (1986a; 1991a). Unlike the vertical systems explored by Brooks, however, PRS also allows top down control of the vertical systems, and it allows processes to control one another (whereas Brooks sees "behaviours" as protected). PRS's operational and denotational semantics have been studied. It is integrated with a temporal logic. It has been used to program a robot that was supposed to act as an astronaut's assistant, which could execute external commands and respond to problems that it detected. Because of these advantages of PRS, it was selected as a basis for the architecture presented in this Ch. 5.

The purported advantage of PRS over combinational planning systems is lost if the latter also contain a mechanism which can produce plan templates that can be invoked and readily executed. As for the purported advantage of PRS over rule-based systems, it

---

[1] To push a goal is to place it on a goal stack.

depends on the relation between the speed with which control conditions can be verified and the speed with which procedures can be selected. Moreover, as Sloman (1994b) points out:

> One way to get the best of both worlds is to have a more general rule-based system which is used when skills are developed and then when something has to go very fast and smoothly bypass the general mechanism by copying the relevant actions inline into the action bodies of the invoking rules. This change increases speed at the cost of control and subsequent modifiability. Some human skill development feels exactly like that!

One problem with PRS is that it can only deal with goals for which it has pre-formed procedures whose applicability conditions match its goal state directly and which can operate in the current state of the world (i.e., whose preconditions have been satisfied). And it deals with these goals by selecting pre-formed (though unexpanded) procedures. That is, for each goal which it adopts it selects whole plans (procedures) which it expands at run time. In contrast, combinational AI planning systems are capable of considering combinations of operators that might achieve the goal state (e.g., Cohen & Feigenbaum, 1982 Ch. XV; Fikes & Nilsson, 1971). Therefore, in complex domains PRS procedures may have to be very elaborate with many conditionals; or there might need to be many different [initial state|goal] pairs. The latter strategy involves having so-called "universal plans", i.e., a huge collection of plans that map situations onto actions. (The difference between the two methods is that with universal plans the conditionals are verified only once (in the selection of the plan) whereas with highly conditional procedures boolean search occurs both in the process of selecting procedures and as a procedure is executing.) There is a time/space trade-off between performing combinatorial search and anticipating and storing complex (or numerous) procedures (or plans).

In defence of PRS it can be conjectured that one could extend the system to allow it to define meta-procedures that combine procedures into new procedures. Achieving this might require the ability to run procedures in hypothetical mode—as Wilensky's model and NML1 assume. This might be facilitated by the fact that PRS processes already encode preconditions and effects. D. E. Wilkins (1988 Ch. 12) reports that a project is underway to create a hybrid system combining PRS with his combinatorial planner, SIPE. A mechanism might also need to be proposed for automatically generating procedures at compile time. Compare (Schoppers, 1987 section 4).

A related concern is that it is not clear how PRS can be used for learning new procedures—whereas production systems, such as Soar (Rosenbloom, Laird, Newell, & McCarl, 1991) do support learning. Perhaps, the kind of learning exhibited by

Sussman's (1975) Hacker would be available to PRS, and automatic programming techniques should also apply.

Georgeff is committed to represent PRS information in first order logic in a monolithic database. Although this simplifies the interpreter's task, such a restriction is a hindrance to efficiency and accessibility, which requires a structured database, multiple types of representation, and multiply indexed information. (See Agre, 1988; Bobrow & Winograd, 1985; Funt, 1980; Gardin & Meltzer, 1989; Sloman, 1985b)

The goal structure and the process structure do not allow the convenient representation of certain relations between goals. For instance, whereas it can implicitly be expressed that one goal is a subgoal of another, it cannot be stated that one goal serves as a means of achieving two goals at the same time. It might be useful to have a structured database containing a variety of kinds of information about goals.

Although the interpreter's method of verifying whether a procedure is applicable is designed to be efficient because it uses pattern matching of ground literals only, it is inefficient in that it sequentially and exhaustively verifies procedures, and the pattern elements are matched against an arbitrarily large database of beliefs and goals. This is an important drawback because any slow down of the interpreter decreases the whole system's reactivity. The problem is linear in complexity with the number of rules and the size of the database. This situation can be improved by assuming that the applicability of procedures is verified in parallel, that the applicability conditions are unified with elements of a smaller database (e.g., goals only) and that a satisficing cycle (as opposed to an exhaustive one) is performed by the interpreter. One might also assume that applicability detection for a procedure can take place over many cycles of the interpreter, so that more time consuming detection can take place without slowing down the system.

In PRS goals can only be generated by procedure activation records as subgoals or by the user as top level goals. It might be advantageous to have asynchronous goal generators (Sloman, 1978 Ch. 6; Sloman, 1987) that respond to certain states of affairs by producing a "top level" goal. (See Chapters 4 ff.). That is, it is sometimes possible to specify a priori that certain conditions should lead the system to generate certain goals. For instance, a system can be built such that whenever its energy level goes beyond a certain point a goal to replenish its energy supply should be generated. The system's ability to generate its own top level goals is an important feature for its "autonomy", and it also favours modularity.

As in AIS, provisions to prevent the distraction of the PRS interpreter are minimal. (The need for this is discussed in Ch. 4).

NML1 will improve on the last five of these limitations of PRS. Although PRS is an architecture that processes goals, it is not based on a theory of goal processing. This makes it difficult to design processes for PRS. The following two chapters present a theory of goals and goal processing. Other strengths and weaknesses of PRS will be discussed in Ch. 5 and Ch. 6.

2.3 Conclusion

Each theory reviewed in this chapter contributes pieces to the jig-saw puzzle of goal processing. None, however, can complete the picture on its own. The strengths and weaknesses of the work reported in the literature are summarised below along the dimensions of: the concepts of goal that are used; the data structures and processes that are supposed; the overall architectures that are proposed; and the principles of decision-making that are used. Most of the criticism refers to the main articles reviewed here, rather than articles mentioned along the way.

The concepts of goals that have been proposed can be assessed in terms of the amount of relevant information they provide, the rigour of the conceptual analysis, whether they are design-based, and whether they situate goals within a taxonomy of other control states. Of the main papers reviewed here, the concept of goal is analysed most thoroughly by goal setting theorists. That is, these theories provide the most information about the dimensions of variation of goals. However, these theories are still not sufficiently general and systematic. For instance, they do not take into consideration the core qualitative components of goals. Most of the other theories are goal based but do not give much information about goals. The PRS model stands out from the rest in providing a syntax for goal expressions and an interpreter for goal expressions which can cope with sophisticated control constructs. This is useful for designing agents. And the present thesis uses the notation and a similar interpreter. None of the theories reviewed here situate goal concepts in relation to a taxonomy of control states; this is done in Ch. 3 and in Boden (1972), which analyses the work of McDougall. (See also Emmons, 1989; Ortony, 1988; Sloman, 1992b). A theory of goals is required that fares well according to all of these criteria.

A small set of data structures and control processes and processors is posited by most theories. Most architectures suppose the use of explicit goals, although AIS does not (it has "tasks" which are similar). AIS and PRS have specific structures that act as a substrate for process types, namely KSARs and procedures. PRS procedures have fewer fields than KSARs and they can execute for longer periods of time. Procedures can serve as plans in their own right, whereas KSARs usually must be strung together to act as plans (within the unique control plan). These two systems offer the most proven control methods of the papers reviewed, and both could serve as a basis for the nursemaid. Oatley and Johnson-Laird's system supposes a form of control based on non-semantic messages, but it is not yet clear how well that will fare.

The theories reviewed here do not provide a wide variety of goal processes. But see (Heckhausen & Kuhl, 1985; Kuhl, 1986; Kuhl, 1992), which describe how goals can be transformed from wishes, to wants, and intentions, and lead to goal satisfying behaviour. A rich process specification along these lines is given in Ch. 4. A system such as PRS is particularly suitable as a substrate for the execution of such processes. Georgeff does not propose a theory determining which goal processes should take place, he merely proposes mechanisms for selecting and executing processes.

No theory is yet up to the task of specifying both the broad variety of goal processes nor sufficiently detailed principles which should guide decision-making. The decision-making rules provided by goal theory and Kuhl are perhaps the most specific. However, they do not provide a sufficiently broad context: i.e., they specify transition rules for specific goals without considering interactions with other goals, e.g., that the achievement of one goal can be traded-off with another. This is a problem that stands out throughout the current thesis. See especially Ch. 6.

The overall architecture of all reviewed designs is at least slightly hierarchical. (For non hierarchical systems see, e.g., Brooks, 1990; Minsky, 1986). The most deeply hierarchical models are those of the communicative theory and AIS. They all draw a sharp distinction between some sort of higher order processor and lower level processors. Oatley and Johnson-Laird and Sloman (1978 Ch. 10) go so far as to equate the higher order process with consciousness in a substantive sense; Schneider and Shallice speak in terms of a higher order attentional process. Stagnant debates can be circumvented by refusing to map these control concepts onto these colloquial substantive terms. As McDougall (according to Boden, 1972) remarks, the adjectival forms of terms like consciousness are usually sounder than the nominal forms. Only the AIS architecture supports reflexes which can by-pass goal based behaviour. It would be trivial to add this to PRS but not to Wilensky's model.

Chapter 3. Conceptual analysis of goals

As the title of this thesis indicates, the concept of goal figures prominently in the present account of autonomous agency. It is therefore imperative to explicate the meaning of the term and to relate it to other concepts. In section 3.1, a taxonomy of control states is presented, and goals are thereby related to other control states. In section 3.2 the concept of goal is analysed, and its dimensions and structural attributes are presented. This results in a notion of goals that is richer than the one usually presented in AI and psychology. In section 3.3, alternative conceptions of goals are reviewed, including Daniel Dennett's argument against mentalistic interpretation of intentional terminology.

3.1 A provisional taxonomy of control states

The current section summarises and expands Sloman's view of goals and other folk psychological categories as control states (Sloman, 1992b; Sloman, 1993b). The rationale of the exposition is that in order to characterise goals, it is useful to present a taxonomy of related concepts in which goals figure. Since goals are understood as a class of control states, this means relating them to other control states.

Sloman views the mind as a control system. Control states are dispositions of a system to respond to internal or external conditions with internal or external actions. They imply the existence of mechanisms existing at least at the level of a "virtual machine". (A virtual machine is a level of ontology and causation that is not physical, but is based on another level which is either a physical machine or a virtual machine. An example of a virtual machine is Poplog's Pop-11 virtual machine (Anderson, 1989).)

Sloman supposes that a human mind non-exclusively comprises belief- and desire-like control states. These states are not "total" but are actually sub-states of a system. Belief-like control states are relatively passive states that respond to and tend to track external events and states. Desire-like control states are states that initiate, terminate, or moderate processes, typically with a view to achieving some state. Sloman writes:

> Thermostats provide a very simple illustration of the idea that a control system can include substates with different functional roles. A thermostat typically has two control states, one belief-like (**B1**) set by the temperature sensor and one desire-like (**D1**), set by the control knob.
>
> • **B1** tends to be modified by changes in a feature of the environment **E1** (its temperature), using an appropriate sensor (**S1**), e.g. a bi-metallic strip.
>
> • **D1** tends, in combination with **B1**, to produce changes in **E1**, via an appropriate output channel (**O1**)) (I've omitted the heater or cooler.) This is a particularly simple feedback control loop: The states (**D1** and **B1**) both admit one-dimensional continuous variation. **D1** is changed by 'users', e.g. via a knob or slider, not shown in this loop.
>
> Arguing whether a thermostat really has desires is silly: the point is that it has different coexisting substates with different functional roles, and the terms 'belief-like' and 'desire-like'

> are merely provisional labels for those differences, until we have a better collection of theory-based concepts. More complex control systems have a far greater variety of coexisting substates. We need to understand that variety. (Sloman, 1992b Section 6)

Figure 3.1 presents a taxonomy of control states including (at the top level) beliefs, imagination, motivators, moods, perturbance, and reflexes. Here follow provisional definitions for these terms. These definitions are provisional because they need to be refined following design-based research. In this thesis, among these states only goals are examined in more detail.

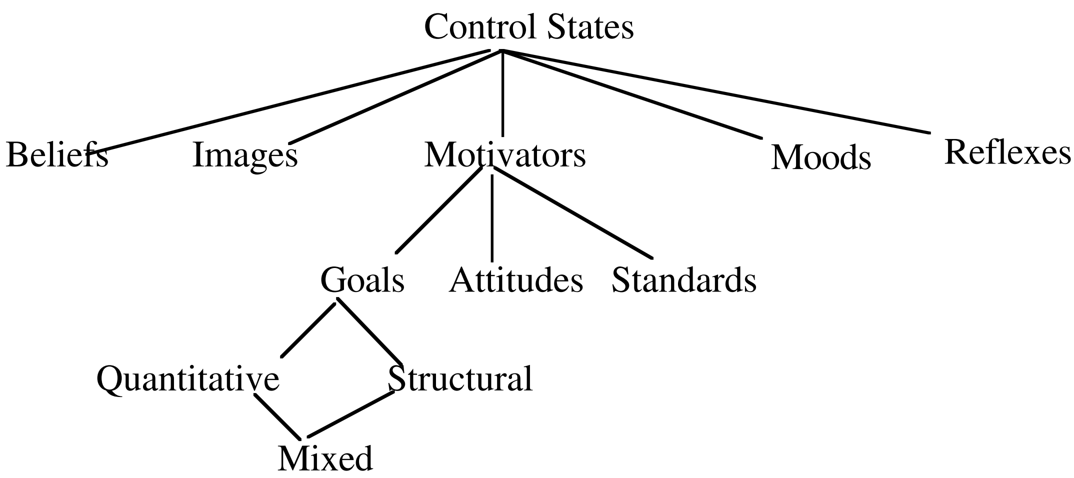

Figure 3.1. Hierarchy of control states.

- Imagination-like control states are similar to belief-like states in both form and content, but their origin and function are different. They are typically used to examine the consequences of possible actions, but they also seem to be used for learning what when wrong in a past endeavour, finding possible causes of events, etc.

- The term "motivator" has been used in two different ways in the literature. In the narrow way (Beaudoin & Sloman, 1993), it is roughly equivalent to the notion of goal which is presented below. In the more general way (Sloman, 1987), it encompasses a wide variety of sub states that have in common the fact that they contain dispositions to assess situations in a certain way—e.g., as good or bad, right or wrong—and that they have the disposition to produce goals. The more general definition is used for this thesis. As Figure 3.1 shows, the main kinds of motivators identified in the theory are: goals, attitudes, and standards.

- A goal can be conceptualised as a representation of a possible state-of-affairs towards which the agent has a motivational attitude. A motivational attitude is a kind of "propositional attitude". The motivational attitude might be to make the state-of-affairs true, to make it false, to make it true faster, prevent it from becoming true, or the like. The representation has the dispositional power to produce action, though the disposition might be suppressed or over-ridden by other factors.

The goal concept used here is similar to other usage of the term in AI and psychology, except that its structure (as given in section 3.2) is richer. There are two main kinds of goals, structural goals and purely quantitative goals. Some goals are combinations of both.

- Structural goals are goals in which the objective is not necessarily described quantitatively. I.e., the objective denotes relations, predicates, states, or behaviours. Most goals studied in AI are structural in this sense.

- Quantitative goals (or "reference conditions") are goals in which the objective is described quantitatively; e.g., the objective might be to elevate the room temperature to 18 degrees Celsius. It is useful to distinguish between structural and quantitative goals because the mechanisms which deal with these kinds of goals can be different. Indeed, there is a branch of mathematics, engineering, AI, and psychology (Powers, 1973) that have evolved specifically to deal with quantitative goals: they have been labelled "control theory". However, the label is misleading because the research it refers to does not study all kinds of control systems, only quantitative ones. A thermostat can be described as a quantitative control system. Such goals have a "reference condition" denoting the desired value of a variable that varies along a certain dimension or set of dimensions. Usually, negative feedback is involved: when an "error" with respect to the reference condition is detected, the system initiates activity which tends to bring the controlled quantity back to the reference condition.

- Attitudes may be defined as "dispositions, or perhaps better, predispositions to like some things, e.g., sweet substances, or classical music or one's children, and to dislike others (e.g., bitter substances, or pop art or one's enemies)" (Ortony, 1988 p. 328). Many attitudes involve intricate collections of beliefs, motivators, likes and dislikes: e.g., the dislike of communists might be combined with a belief that they are out to remove our freedom.

- Standards are expressions denoting what one believes ought to be the case as opposed to what one simply wants—or would like—to be the case. Related terms are prescriptions, norms, and ethical, social, or personal rules. If a person is cognisant that some state, **S**, violates one of his standards, then he is disposed to produce the goal to counteract **S** and/or condemn the agent that brings **S** about.

- Perturbance is an emergent dispositional state in which an agent loses control over some of its management of goals. This technical definition will only make sense to the reader by Ch. 4, once goals and management processes have been described. All that matters for this section is that a difference between goals and perturbance be noted by the reader. A state of perturbance is not a goal, but it arises out of the processing of goals. In Ch. 7, a relation between perturbance and "emotion" is discussed.

- Sloman says of certain <u>moods</u> that they are "persistent states with dispositional power to color and modify a host of other states and processes. Such moods can sometimes be caused by cognitive events with semantic content, though they need not be. [...] Similarly their control function does not require specific semantic content, though they can influence cognitive processes that do involve semantic content." (Sloman, 1992b Section 6). A similar view is taken in (Oatley, 1992). To be more precise, moods are temporary control states which increase the prominence of some motivators while decreasing others. In particular, they affect the likelihood that certain "goal generators" are triggered. Moreover, moods affect the valence of affective evaluations, and the likelihood of affective evaluations (perhaps by modifying thresholds of mechanisms that trigger evaluations). It is not yet clear whether moods as defined here are useful, or whether they merely emerge as side-effects of functional processes.

- A <u>reflex</u> is a ballistic form of behaviour that can be specified by a narrow set of rules based on input integration and a narrow amount of internal state. There are two kinds of reflexes: simple reflexes and fixed action patterns. A simple reflex involves one action, whereas a fixed action pattern involves a collection of actions. Usually, at most only a small amount of perceptual feedback influences reflex action. This would require a definition of action, which is not provided in this thesis.

- Future research may try to elucidate the concept of <u>personality traits</u> as higher order motivators.

This taxonomy is quite sketchy. Every definition by itself is unsatisfactory. This thesis is concerned with goals. Given the controversies surrounding the terms "moods" and "attitudes", these terms could be replaced by technical terms without the theory being worse off because of it. Providing more elaborate distinctions requires expanding the computational architecture that supports the control states.

There are related taxonomies in the literature. Powers (1973) presents a quantitative control-theoretic account of perception and action. Power's framework has been used by C. Carver and M. Scheier (1982). R. A. Emmons (1989) presents a hierarchical theory of motivation, which breaks down the "personal strivings" of individuals into decreasingly abstract categories. M. Boden (1972) reviews William McDougall's theory of psychology, which involves such control states as instincts, sentiments, and emotions. K. J. Holyoak, K. Koh, and R. E. Nisbett (1989) present a mechanistic model of learning in which rules of various degrees of abstraction are generated and subject to a selection process. C. Lamontagne (1987) presents a language for describing hierarchical cognitive systems. It would be a useful experiment to use Lamontagne's language for expressing a hierarchy of control states.

The foregoing taxonomy is clearly oversimplified—but it will do as a sketch for this thesis, which is mainly concerned with goals. It is left for future research to analyse other control states in more detail.

3.1.1 Attributes of control states

In order to distinguish between classes of control state and between instances of classes of control states, one needs to know what their attributes are. Mathematically, there are two types of attributes: dimensional and structural attributes. Dimensions are quantitative attributes. Structural attributes are predicates, relations, and propositions. A. Sloman (1992b; 1993b) discusses some of the attributes of control states: e.g., their duration, the indirectness of their links with behaviour, the variety of control states which they can effect, their degree of modifiability, whether their function requires specific semantic content or not, with what states they can co-exist, the frequency with which the state is generated or activated, the time it takes for the state to develop, how they are brought about, how they are terminated, how they can be modulated, how sensitive they are to run time events, which states do they depend on, etc. Values on dimensions can be explained in terms of the structural attributes—e.g., the duration of a perturbance can be explained in terms of beliefs, mechanisms for activating goals, and the system's ability to satisfy the goals.

Formally speaking, there are many ways of distinguishing between control states. One method for distinguishing amongst classes of control states involves finding whether their typical or mean values on one of the attributes differ. Classes rarely differ by having non-overlapping distributions of attribute values. For example, personality traits are by most definitions more long lasting than moods. Another method involves finding whether one type of control state has a greater variance along one of the dimensions than another. For example, perhaps perturbance tends only to have very direct causal links with behaviour, whereas goals can either have very direct links or very indirect links with behaviour. This is analogous to the kinds of differences detected by ANOVAs in inferential statistics.

A third method is by determining that one category does not have the same attributes as another. A clear example of this from another domain is the case of the difference between lines and points in mathematics or visual perception: lines have a length and an orientation whereas points do not have such a dimension of variation. An example of this in terms of control states is that some attributes of goals that are described below do not apply to other control states: goals differ from attitudes, moods, and standards in that scheduling information can be attached to them. If one decides to be in a bad mood at a certain time, then one is not processing the mood directly; rather, one is processing a goal which has as its object a mood or behaviours that will induce a mood. This method of distinguishing between control states suggests that a useful improvement should be brought to the taxonomy (hierarchy) of control states: it should specify an inheritance hierarchy of attributes. Such a hierarchy would state which attributes are relevant to which class of control states, what the class/subclass relations are, and (implicitly) inheritance of attributes from class to subclass. The author is not primarily suggesting that values along the attributes should be inherited (although that too would be possible), but that the type of attribute should be inherited as in object oriented design. As is typical in the course of object oriented design, this would probably lead to a reorganisation of the hierarchy, the postulation of new classes, new relations amongst classes, and new attributes of classes. If nothing else, this would allow us to provide a much more general characterisation of the features of goals. But this is left for future research.

It is important to distinguish between distinctions between types of control states (e.g., comparing the category of goals with the category of prescriptions) and distinctions between instances of a type of control states (e.g., comparing one goal with another). This section focused on categories. In the next section, the attributes of goals are discussed.

3.2. The conceptual structure of goals

Conceptually, goals are complex structures involving core components which individuate particular goals, and a wide variety of other components that can be associated with them. In this section, the concept of goal is expounded. This can be read as the requirements of purposive control states. An important caveat is in order: there is no implication that the components of this conceptual structure are to be implemented explicitly as fields in a record. A system might be capable of maintaining goal information in an implicit or distributed fashion.

3.2.1 The core information of goals

As was said above, to a first approximation a goal is a "representation of a possible state of affairs towards which the agent has a motivational attitude." This is the core of a goal. The representation of a state-of-affairs can be expressed propositionally (e.g., in predicate calculus), and referred to as the "proposition" of a goal. For instance, if the nursemaid wishes to recharge babyA, it might express the propositional aspect of this goal as

charged(babyA).

A motivational attitude determines the kind of behavioural inclination which an agent has towards a proposition. This can be to make the proposition true, to make it false, to prevent it from being true, to keep it true, to make it true faster, or the like. In the example, the nursemaid's motivational attitude towards this proposition is "make-true". The proposition of a goal has a denotational semantics and can be interpreted as being true or false with respect to the subject's beliefs or objectively. However the motivational attitude when applied to the proposition yields a structure which is neither true nor false: it is an imperative, i.e., it specifies something which is to be done.

- The foregoing specification of goals has the disadvantage that every goal only has one motivational attitude towards one proposition: it does not allow one to express multiple propositions and attitudes within a single goal concept. For instance, it does not allow one to express a goal to "maintain **q** while preventing **p**", which contains an attitude of "maintenance" and one of "prevention". Moreover, standard predicate calculus cannot express a "while" constraint, e.g., "(achieve) **q** WHILE (doing) **p**"—which is not strictly equivalent to "**q** and **p**". A temporal logic is required which can express such propositions. Thus, a more general notion of goals is proposed: a goal is a proposition containing motivational attitudes and descriptors, where the former are applied to the latter. It is left to future research to lend more precision to this definition.

Meanwhile, the PRS notion of goals is provisionally used, since it comes relatively close to meeting the requirements (Georgeff & Lansky, 1986). The PRS goal specification calls propositions "state descriptions", and it uses temporal operators to express "constraints" (which are similar to motivational attitudes). The temporal operators "!" (make true), and "#" (keep true) are used. They are applied to propositions in predicate calculus notation. The core information of the goal to recharge babyA without going through room 5 could be expressed as

!charged(babyA) and #(not(position(claw) = room5))

In the language of PRS, this goal is called a "temporal action description". This is because it is a specification of required behaviour. For brevity, in this thesis, such expressions are simply called "descriptors"; however, the reader should not be misled into believing that descriptors are non-intentional statements. Of course, the interpretation of particular goals requires a system which is capable of selecting appropriate behaviours that apply to goal descriptors; the interpreter described in (Georgeff & Lansky, 1986) and summarised in Ch. 2 fulfils that function.

Unfortunately, the PRS representation of goals (cf. Ch. 2) does not have the expressive power that is ultimately required. That would necessitate temporal operators that stand not only for achievement and preservation attitudes, but the other attitudes listed above as well—e.g., to make a proposition true faster. Furthermore, the interpretation of "while" and "without" in terms of the # operator and negation is impoverished since it does not fully capture the intervals during which the constraints should hold. (See (Allen, 1991; Pelavin, 1991; Vere, 1983) for more comprehensive temporal logics.) However, as a provisional notation it is acceptable for this thesis. Future research should improve upon it.

3.2.2 Attributes of goals

Like other control states, goals have many attributes. The attributes that are enumerated in this section are the knowledge that an agent typically will need to generate with regard to its goals. They are summarised in Table 3.1. Most of this knowledge refers to assessment of goals and decisions about goals. Along with the enumerated attributes, other relevant goal attributes are presented below.

Table 3.1

The conceptual structure of goals

Attribute type	**Attribute name**
Essence:	Goal descriptor
Miscellaneous:	Belief
Assessment:	Importance
	Urgency
	Insistence
	Rationale
	Dynamic state
Decision:	Commitment status
	Plan
	Schedule
	Intensity

One fact emerges from the following analysis of goals: goals are very complex control states with many subtle links to internal processes which influence external actions in various degrees of indirectness. It is possible that some of the historical scepticism about the usefulness of the concept of goal is due to the fact that goal features have not yet been characterised in enough detail and in terms that are amenable to design-based specification. Other reasons are explored by Boden (1972). The author does not claim to have produced such a characterisation; but he does claim to have taken some step towards it.

(1) Goals have a descriptor, as explained in the previous section. This is the essential characteristic of goals, i.e. what makes a control state a goal. The fact that goals have conceptual or propositional components implies that all attributes of propositions apply to goals. Exactly which attributes there are depends on the language used to express goals. For example, if predicates can vary in degree of abstraction, then goals would differ in degree of abstraction. If the language allows propositions to differ in degree of articulation (specificity vs. vagueness) then so will goals (Kagan, 1972; Oatley, 1992). Descriptors along with other knowledge stored in the system implicitly indicate the kind of achievability of a goal. Goals are achievable either in an all-or-none fashion or in a partial (graded) fashion (Haddawy & Hanks, 1993; Ortony, Clore, & Collins, 1988 p. 44). For instance, the goal to charge a baby is a partially achievable goal, because a baby's battery can be more or less charged. The goal to dismiss a baby is an all-or-none goal, because it is not possible merely to satisfy

this goal partly. Even for all-or-none goals, however, it might be possible to take actions which bring one more or less close to satisfying it, in the sense that having performed some of the work toward a goal, less work is now required to satisfy it. Hence achievability can be relative to the end or the means to the end.

(2) Beliefs are associated with goals. They indicate what the agent takes to be the case about the components of the goal's descriptor, such as whether they are true or false, or likely to be true or false, along with information about the certainty of the beliefs—e.g., (Cohen, 1985). Thus far, the theory is non-committal about how beliefs are processed or represented. Beliefs about the goal state together with the goal descriptor determine a behavioural disposition. For example, if the descriptor expresses an (adopted) achievement goal regarding a proposition **P** and **P** is believed to be false then the agent should tend to make **P** true (other things being equal).

3.2.2.1 Assessment of goals

In order to take decisions about goals, a variety of evaluations can be computed and associated with goals, as follows.

(3) importance descriptors represent the costs and benefits of satisfying or failing to satisfy the goal. The notion of importance or value is intentional and linked with complex cognitive machinery for relating goals amongst themselves, and anticipating the outcomes of actions. One cannot understand the importance of a goal without referring to other aspects of the agent's mental life. For instance, one of the main functions of computing importance of goals is determining whether or not the agent will adopt the goal.

In contrast with decision theory (cf. Ch. 5) here it is not assumed that importance ultimately should be represented by a quantitative value or vector. Often, merely noting the consequences of not satisfying a goal is enough to indicate its importance. For instance, someone who knows the relative importance of saving a baby's life will find it a sufficient characterisation of the importance of recharging a baby's battery that the consequence of not recharging the battery is that the baby dies. In other words, if such an agent could speak English and he were asked "How important is it to recharge this baby's battery?" he might answer "It is very important, because if you do not then the baby will die." No further reasoning would be required, because the relative importance of a baby's death is already known: in particular no numeric value of the importance is required. However, if a decision between two goals were required, then the system would compare the consequences of satisfying or not satisfying either goal. For some systems, this could be realised by storing partial orders, such as the rules described in section 1.5. (By definition partial orders are not necessarily total, and hence the system might be unable to make a principled choice between two goals.

Furthermore, goals can have many consequences, and that complicates deciding.) Further discussion of the quantitative/qualitative issue is deferred to Ch. 6.

For partially achievable goals, it might be useful to find the value of different degrees of achievement and non-achievement. The distinction between partial achievement and partial non-achievement is significant with respect to the importance of goals because, for instance, there might be positive value in partially achieving some goal while there might also be adverse consequences of failing to achieve a greater portion of the goal. For example, **X** might have the task of purchasing 10 items. If **X** just purchases eight of them, this might contribute to **X**'s well being. However, this might lead **X**'s partner to chastise **X** for having bought some but not all of the required items. In other words (and more generally), the importance of a goal includes various factors that are consequences of the goal (not) being satisfied. In social motivation there are many examples of adverse side-effects of not satisfying a goal.

There are intrinsic and extrinsic aspects to the importance of goals. The intrinsic aspects are directly implicated in the goal state. These are the goals which are "good in themselves", e.g., producing something aesthetically appealing, performing a moral action, being free from pain, enjoying something pleasant, etc.. What matters for the present purpose is not what humans treat as intrinsically good, but what it means to treat something as intrinsically good. To a first approximation, something is intrinsically good if an agent is willing to work to achieve it for its own sake, even if the agent believes that the usual consequences of the thing do not hold, or even if the agent does not value the consequences of the thing. In other words, (1) intrinsic importance is entirely determined by the propositional content expressed in the goal and not by any causal or other implications of that content; (2) any goal with this same content will always have some importance, and therefore some disposition to be adopted, no matter what else is the case (nevertheless, relative importance will be context dependent). None of this implies that the agent's tendency to work for the object will not eventually weaken if its usual consequences no longer hold. This would be analogous to "extinction" in operant conditioning terms. An ontogenetic theory would need to allow for the possibility that an objective started out as extrinsically important, but then was promoted to being important in itself. Intrinsic importance or value of a goal state is sometimes referred to as functional autonomy of that state (Allport, 1961). (See also (Boden, 1972, Ch. 6, pp. 206-207)). The idea is that even if the motivator is ontogenetically derived from some other motivator, it functions as if its value is inherent. An analogous case holds for phylogenetic derivation of value.

The extrinsic importance of a goal is due to the belief that it preserves, furthers, or prevents some other valued state. This subsumes cases in which one goal is a subgoal of another. Extrinsic importance can be divided into two categories: goal consequences and plan consequences. (a) Goal consequences are the main type of extrinsically valenced consequences of goals. These are the valenced consequences that follow from the satisfaction of the goal regardless of what plan the subject uses to achieve the goal. (b) Plan consequences are valenced side-effects of the plans used to achieve a goal. (Plans are discussed in Section 3.2.2.2.) Different plans can have different consequences. The agent might need to distinguish between consequences that follow from every plan to satisfy a goal—i.e., inevitable consequences—and those that only follow from a subset of plans (i.e. peculiar consequences). Inevitable plan consequences although logically distinct from goal consequences can be treated in the same way as goal consequences (unless the agent can learn new plans). To illustrate, it might be an inevitable consequence of the plans to achieve a goal that some babies risk getting injured (e.g., if we assume that the nursemaid can only depopulate a room by hurling babies over the walls that separate the rooms). Plan contingent consequences can be used to select amongst different plans for a goal.

The concept of importance can be illustrated with an example from the nursery. If the nursemaid detects that a room is overpopulated, it will produce a goal (call it **G**) to depopulate the room. Assume that the nursemaid treats this goal as having a little "intrinsic importance", meaning that even if the usual extrinsically important consequences of **G** were guaranteed not to hold, the nursemaid would still work for **G**. As a colloquial description, "the nursemaid likes to keep the population in a room under a certain threshold". Just how important this is to the nursemaid is really a matter of what other things it is willing to give up in order to satisfy this goal. **G** also has extrinsic "goal consequences", for by preserving **G**, the nursemaid decreases the likelihood of a baby becoming a thug. In fact, no baby will turn into a thug in a room where **G** is preserved (i.e., a non-overpopulated room). In turn, preventing babies from turning into thugs is important because thugs injure babies, and thugs need to be isolated. The importance of these factors in turn can be described: injuries are intrinsically bad, and require that the nursemaid put the injured babies in the infirmary thus using up claw and the infirmary, both of which are limited resources. The nursemaid should be able to identify the importance of the goal in terms of one of these "plies" of consequences, without producing an infinite regress of reasoning about the effects of effects of effects. In an ideal implementation of a nursemaid, it should be easy for the user to manipulate the agent's valuation of **G**.

The valenced factors that an agent considers should be orthogonal, or if there is overlap between them the agent should recognise the overlap. Otherwise, one might overweigh one of the factors. The worst case of a violation of the orthogonality constraint is when two considered factors are actually identical (though they can differ in their names). An example of such an error is if the nursemaid was considering the goal to move a baby away from the ditch. It might correctly conclude that if it did not adopt this goal then the baby would fall into the ditch and die (say from the impact). Then it might also conclude that since the baby is irretrievably in the ditch its battery charge would eventually go down to zero, and die. The error, then, would be to factor in the baby's death twice when comparing the importance of the goal to prevent the baby from falling into the ditch with some other goal. The author conjectures that this kind of error is sometimes seen in human decision making, especially when the alternatives are complex and the relations amongst them can be muddled because of memory load. Another case is when one factor is actually a subset of another. Another case is if two factors partly overlap. For example, when assessing the importance of **G**, the agent might consider the consequence **C1**: "if I do not satisfy **G** then some baby might turn into a thug", **C2**: "babies might get uncomfortable because of overcrowding", and **C3**: "babies might get injured by the thug". **C1** and **C3** are not independent, because part of the reason why it is not good to turn babies into thugs (**C1**) is that this might lead to babies being injured (**C3**).

There are many goal relations and dimensions which are implicit or implicated in the assessment of importance, such as hierarchical relations amongst goals. Some of these relations have been expressed in imprecise or insufficiently general terms in related work. That is corrected here. Many goals exist in a hierarchical network of goal-subgoal relations, where a supergoal has subgoals that are disjunctively and/or conjunctively related.[1] A goal that is a subgoal to some other goal can derive importance from its supergoal. Similarly, a goal that interferes with another can aquire negative importance. There are two complementary pairs of reciprocal dimensions of hierarchical goal relations that are particularly significant. The first pair of dimensions is criticality and breadth of goals. Criticality is a relation between a subgoal, $\mathbf{G}_\partial$, and its supergoal, **G**. The smaller the number of subgoals that are disjunctively related to **G**, the more <u>critical</u> each one of these goals is to **G**. In other words, if a goal **G** can be solved by executing the following plan:

$$\mathbf{G_1} \text{ or } \mathbf{G_2} \text{ ... or } \mathbf{G_N}$$

where $\mathbf{G}_\partial$ is one of $\mathbf{G_1}$, $\mathbf{G_2}$, ... $\mathbf{G_N}$ and $\mathbf{G}_\partial$ is a subgoal of **G**, then the criticality of $\mathbf{G}_\partial$ to **G** is equal to 1/**N**. I.e., $\mathbf{G}_\partial$ is critical to **G** to the extent that there are few other goals besides $\mathbf{G}_\partial$ that can achieve **G**. A more general notion of criticality would also consider the relative costs of the alternative goals as well as their probability of success. With the more general notion, the criticality of $\mathbf{G}_\partial$ to **G** would be inversely proportional to **N**, inversely proportional to the ratio of the cost $\mathbf{G}_\partial$ to the cost of the other subgoals, and inversely proportional to the ratio of the probability of success of $\mathbf{G}_\partial$ to the probability of success of the other goals. Other things being equal, if $\mathbf{G}_\partial$ is more critical to **G** than $\mathbf{G}_ß$ is to **G** then $\mathbf{G}_\partial$ should inherit more of **G**'s importance than $\mathbf{G}_ß$ does. This notion of "criticality" allows one to treat the relation of "necessity" (Ortony, et al., 1988) as a special case of criticality: <u>i.e.</u>, ultimate criticality. Ortony and colleagues claim that a subgoal is necessary to its supergoal if it must

[1] Oatley (1992) points out that humans often lose track of goal subgoal relations (i.e., they have fragmentary plans). From a design stance, this is a fact that needs to be explained and not merely assumed to be the case. Is this fact a necessary consequence of requirements of autonomous agency?

be achieved in order for the supergoal to be achieved. This is the special case of criticality of $\mathbf{G}_\partial$ where **N** is equal to 1.

The <u>breadth</u> of a supergoal **G** is simply the reciprocal of criticality of immediate subgoals of **G**. That is, the breadth of **G** is equal to **N**. Thus the breadth of **G** is the number of goals that can independently satisfy **G**. Thus a goal is wide if it can be satisfied in many ways, and narrow if it can be satisfied in few ways. It appears that children often produce goals that are overly narrow, as Sloman (personal communication), and J. Kagan (1972) suggest. For instance, in the scenario presented in the introduction where Mary took Dicky's toy, one might expect that if Dicky was offered a different instance of the same kind of toy he would not be satisfied, he would want to have <u>that</u> toy back. We might say that Dicky's motive is insufficiently <u>broad</u>, he does not realise that other toys (other possible subgoals) could do just as well. (Of course, the subjectivity of motivation complicates the matter of imputing error upon a person's desires.) The researcher is left with the task of producing a cognitive developmental explanation of the increase in breadth of goals as children get older. (This might somehow be related to variations in abstraction of the goals that are expressed.)

A second pair of dimensions, this time for the conjunction operator, is proposed: sufficiency and complexity. Whereas Ortony, Clore, and Collins (1988) see sufficiency as a categorical notion, it can be viewed as a dimension. Sufficiency is a relation between a subgoal, $\mathbf{G}_\partial$, and its supergoal, **G**. The smaller the number of subgoals that are conjunctively related to **G**, the more <u>sufficient</u> each one of these goals is to **G**. In other words, if a goal **G** can be solved by executing the following plan:

$\mathbf{G_1}$ and $\mathbf{G_2}$ ... and $\mathbf{G_N}$

where $\mathbf{G}_\partial$ is one of $\mathbf{G_1}$, $\mathbf{G_2}$, ... $\mathbf{G_N}$ and $\mathbf{G}_\partial$ is a subgoal of **G**, then the sufficiency of $\mathbf{G}_\partial$ to **G** is equal to 1/**N**. Thus, the categorical notion of sufficiency is a special case, where **N** is equal to 1.

The <u>complexity</u> of a supergoal **G** is the reciprocal of sufficiency of immediate subgoals of **G**. That is, the complexity of **G** is equal to **N**. Thus the complexity of **G** is the number of goals that are required to satisfy **G**.

(4) An agent also must be able to form beliefs about the <u>urgency</u> of goals. In simple cases, the notion of urgency is the same as that of a deadline: <u>i.e.</u>, it indicates the amount of time left before it is too late to satisfy the goal. This is called "deadline urgency" or "terminal urgency". A more general notion of urgency is more complex: here urgency reflects temporal information about the costs, benefits, and probability of achieving the goal (Beaudoin & Sloman, 1991). For instance, urgency information might indicate that the importance of satisfying a goal increases monotonically with time, or that there are two junctures at which action is much less risky or costly. Hence urgency is not necessarily monotonic, and urgency descriptors can be used to characterise some opportunities. An

even more general notion of urgency is not only indexed in terms of quantitative time, but can be indexed by arbitrary conditions: e.g., that executing the goal to recharge a baby will be less costly when a new and more efficient battery charger is installed. In this example, the juncture is a condition denoted by a proposition, not a quantitatively determined juncture.

Urgency can either be conceived in an outcome centred or an action (or agent) centred manner. When urgency is outcome centred, it is computed with respect to the juncture of occurrence of the event in question (e.g., when a baby will fall into a ditch). If it is action centred it is computed with respect to the juncture at which an agent behaves (e.g., the latest time at which the nursemaid can successfully initiate movement toward the baby heading for the ditch).

The achievability of the goal is also relevant to estimates of urgency. In the example of saving the baby, one is faced with an all-or-none goal, as well as circumstantial constraints (the cost and the likelihood of success) depending upon the time at which the action is undertaken. If the goal itself is partially achievable, then the extent to which it is achieved can be a function of the time at which action is commenced. For instance, a baby that is being assaulted might suffer irreversible effects the importance of which are monotonically related to the time at which protective action commences.

(5) For reasons described in the following chapter, it is sometimes useful to associate measures of insistence with goals (Sloman, 1987). Insistence can be conceived as heuristic measures of the importance and urgency of goals. Insistence will be shown to determine whether a goal is considered by "high level" processes. Goals that are insistent over long periods of time are likely to be frequently considered, and hence are said to be "prominent" during that period.

(6) It is also often useful to record the original rationale for a goal. This indicates the reason why the goal was produced in the agent. (Like other information it is often possible to know this implicitly, e.g., because of the goal's position in a goal stack.) Rationale is closely linked to the importance of a goal. The rationale might be that the goal is a subgoal of some other goal; and/or it might be that some motivationally relevant fact is true. For instance, a nursemaid that treats keeping babies' charge above a certain threshold as a top level goal might see the mere fact that babyA's charge is low as the rationale of the new goal to recharge babyA. Issues of recording reasons for goals can be related to the literature on dependency maintenance, e.g., (Doyle, 1979). The task of empirically identifying an agent's top level goals is discussed in (Boden, 1972 pp. 158-198).

(7) There is a record of the goal's dynamic state such as "being considered", "consideration deferred", "currently being managed", "plan suspended", "plan aborted". The kind of dynamic information that is required will depend on the agent's meta-level reasoning capabilities. An important dimension of the dynamic state is the goal's state of activation, this is discussed in the next chapter, once the notions of insistence based filtering and management have been expounded. Many of the

problems of designing an autonomous agent arise out of the fact that many goals can exist simultaneously in different states of processing, and new ones can be generated at any time, potentially disturbing current processing.

Attributes (3) to (7) represent assessments of goals. These measures have a function in the agent—they are used to make decisions about goals. As is explained in Ch. 4, autonomous agents must be able to assess not only goals, but plans and situations as well.

3.2.2.2 Decisions about goals

This section examines the four main kinds of decision about goals.

(8) Goals acquire a commitment status (or adoption status), such as "adopted", "rejected", or "undecided".[1] Goals that are rejected or have not been adopted usually will not be acted upon. The likelihood of commitment to a goal should be a function of its importance: i.e., proportional to its benefits, and inversely proportional to its cost. However these factors can be completely overridden in the context of other goals of high importance. Processes which lead to decisions are called "deciding" processes. The process of setting the commitment status is referred to as "deciding a goal". An example of a commitment status is if the nursemaid decides to adopt the goal to charge babyA.

(9) A plan or set of plans for achieving the goal can be produced. This comprises both plans that have been adopted (as intentions), and plans that are candidates for adoption (Bratman, 1990). Plans can be partial, with details left to be filled in at execution time, or when more information is available. The breadth of a goal is proportional to the size of the set of possible plans for a goal. That is, a wide goal is a goal which can be satisfied in many different ways. A record of the status of execution of plans must be maintained, and the plan must contain a reference to the goal that motivates it (compare the two-way process-purpose index in section 6.6 of (Sloman, 1978), and Ch. 5 below).

(10) Scheduling decisions denote when the goal is to be executed or considered. Thus one can distinguish between physical action scheduling decisions and deliberation scheduling decisions, though many scheduling decisions are mixed (e.g., to the extent that action requires deliberation). Scheduling decisions can be expressed in terms of condition-action pairs, such that when the conditions are satisfied mental or physical actions should be taken. An example of a scheduling decision is if the nursemaid decides "to execute the plan for the goal to recharge babyA when there is enough room in the nursery". The execution condition is partly structural and relative as opposed to being expressed in terms of absolute time. Scheduling is the subject of much recent research in AI

[1]Commitment in social organisms has additional complexity that is not examined here.

(Beck, 1992; Fox & Smith, 1984; Gomes & Beck, 1992; Prosser, 1989; Slany, Stary, & Dorn, 1992).

(11) Finally, goals can be more or less intense. Intensity is a measure of the strength of the disposition to act on the goal, which determines how vigorously it is to be pursued (Sloman, 1987). Intensity is a subtle concept which as yet has not been sufficiently explained. Intensity is not a descriptive measure. In particular, it is not a descriptive measure of current or past performance, nor of the sacrifices that an agent makes to pursue the goal. Rather intensity is a prescriptive measure which is used by an agent to determine the extent of the goal's propensity to drive action to satisfy it. The word "should" here does not denote a moral imperative; instead, it has mechanistic interpretation, in that whatever mental systems drive action will be particularly responsive to intensity measures.

The links between measures of intensity and action are stronger than the links between measures of importance and action, and between urgency and action. An actual design is required to specify more precisely how intensity is computed and the precise way in which it directs action. Still, it can be said that although on a statistical basis the intensity of goals should be highly correlated with their importance and urgency, especially if the cost of achieving them is low, this correlation is not perfect. Sometimes, important goals cannot be very intense, because of a recognition of the negative impact which any behaviour to achieve it might have. Furthermore, it is a sad fact about human nature that some goals can be evaluated as having low or even negative importance and yet be very intense. A person who regretfully views himself as intemperate can usually partly be described as having a goal which is very intense but negatively important. Whereas obsessions, in the clinical sense, involve insistent goals and thoughts that are not necessarily intense, compulsions involve intense goals (Barlow, 1988). (Obsessive-compulsive disorder is described in Ch. 7.) Explaining how intensity can be controlled is a particularly important psychological question, because of the directness of its links with behaviour.

Elaboration of the theory may try to define and explain the terms "pleasure" and "displeasure", which possibly refer to dimensions of goals.

Most of the information about goals can be qualitative or quantitative, conditional, compound and gradually elaborated. For instance, the commitment status of a goal might be dependent on some external condition: e.g., "I'll go to the party if I hear that Veronica is going". And an agent might be more or less committed to a goal (Hollenbeck & Klein, 1987). More information would need to be recorded in an agent that learns. The information about goals will further be discussed below as the goal processes are specified.

In Ch. 4 information about goals is elaborated in the context of management processes that produce these assessments. Simplified examples of goals are provided in Ch. 5, which contains a scenario and a design of an autonomous agent.

3.3 Competing interpretations of goal concepts

There has long been an uneasiness with intentional concepts in general, and with the terms "goal" and "motive" in particular. M. Boden (1972) has dealt convincingly with the arguments of reductionists and humanists who, for different reasons, reject the possibility of a purposive mechanistic psychology.

Readers who are comfortable with the concept of goals provided in the previous sections are advised to skip this section on first reading, as it merely defends the concept in relation to the work of others.

Some authors note the paucity of clear definitions of goals and the diversity of relevant definitions, e.g., (Heckhausen & Kuhl, 1985; Kagan, 1972). For instance, H. Heckhausen and J. Kuhl (1985) write "goal is a notoriously ill-defined term in motivation theory. We define goal as the molar endstate whose attainment requires actions by the individual pursuing it" (1985 pp. 137-138). Although apparently valid, there are a number of problems with this definition. Firstly, the definition does not circumscribe an intentional state, i.e., it is not written in terms of "a representation (or proposition) whose attainment ...". Secondly, it leaves out an essential component which distinguishes goals from representations of other states, such as beliefs, namely a motivational attitude toward the state. Thirdly, it leaves out an important kind of goal namely "interest goals" (Ortony, et al., 1988), i.e., states which the agent cannot bring about but would like to see true (such as wanting a certain team to win a football game, but not being able to help it). This can be allowed in various ways in the goal concept used here. For instance, there could be a motivator with a "make true" attitude, and plan information showing that there was no feasible plan to make it true. There is a fourth problem, which is closely related to the third: the definition excludes those states which an agent wishes to be true but which do not require action on his behalf because someone else will achieve them. Fifth, the definition encompasses some things which are not goals: i.e. all of those things which require action by an agent but which are not his goals. This faulty definition suggests that it is not a trivial task to provide an acceptable definition of goals.

3.3.1 Formal theories of "belief, desire, intention" systems

Many psychology and AI papers use the term "goal" without defining it. For instance, a seminal paper on planning does not even define the term goal, though the implicit definition was: a predicate or relation to be made true (Fikes & Nilsson., 1971). Nevertheless, the most formal attempts to define goals are to be found in recent AI and philosophical literature (Cohen & Levesque, 1990; Rao & Georgeff, 1991). Indeed, H. A. Simon (1993) and M. Pollack (1992) note the apparent "theorem envy" of some AI researchers in recent years.

P. R. Cohen and H. J. Levesque (1990) provide a specification level analysis of belief-desire-intention systems (though not a design). A formal specification comprises a syntax, definitions, and axioms. The Cohen and Levesques specification is meant to provide principles for constraining the relationships between a rational agent's beliefs, desires, intentions, and actions. They cite (Bratman, 1987) as providing requirements of such a specification, such as: (1) Intentions are states which an agent normally tries to achieve (though they will not necessarily intend to achieve all of the side-effects of these attempts) and the agent monitors its attempts to achieve them, retrying if the attempts fail. (2) Intentions constrain what future goals are adopted as intentions: intentions must not be incompatible. (3) Intentions must be states which the agents believe are possible. Although the aim of providing formal specifications is apparently laudable, an unfortunate problem with them is that they are usually overly restrictive. Two of the constraints of (Cohen & Levesque, 1990; Rao & Georgeff, 1991) are particularly troubling. (1) They require that goals be consistent. However, this requirement is too harsh for modelling agents such as human beings, because it is known that not only can goals be inconsistent, but so can intentions. A common experience is to have two different incompatible intentions for a lunch period. (2) In order to be able to propose a universal formula, the authors assume that the agent knows everything about the current state of the world. However, this assumption violates a requirement of autonomous agents, namely that they should be able to cope with incomplete and possibly erroneous or inconsistent world knowledge.

Aristotle's injunction that virtue lies in the mean between a vice of excess and a vice of defect is applicable here. Theories that are too stringent outlaw known possibilities, whereas those that are insufficiently clear fail to distinguish between known possibilities. Formal theories tend to be too stringent, and psychological theories tend to be insufficiently clear.

3.3.2 Arguments against viewing goals as mental states

An old but still contemporary question about the interpretation of "goals" is whether or not they are best characterised as internal states or external attributes of an agent. This debate dates from the early days of behaviourism in psychology. Most AI researchers and cognitive scientists had until recently espoused the view that intentional states (like belief and desire) usefully could be represented in computers. See (Brachman & Levesque, 1985; Dennett, 1978 Ch. 7; Dennett, 1987 Ch. 6). This view contradicted many of the tenets of behaviourism.

Over the last decade, however, there have been renewed criticisms of received notions of representation in general and of goals and plans in particular. For instance, some connectionists have argued that the use of pointer referenced data-structures must be kept to a strict minimum (Agre, 1988 pp. 182-188). However, the author knows of no cogent argument to the effect that no internal representation is used. For instance, although Brooks (1991b) entitled his paper "Intelligence without representation", he later says that he merely rejects "traditional AI representation schemes" and representations of goals. Hence he is merely suggesting different representational schemes. Brooks and his colleagues emphasise the importance of interaction between an agent and its environment in determining behaviour, as if this was not obvious to everyone else. In a similar vein, R. W. White (1959) writes:

> Dealing with the environment means carrying on a continuing transaction which gradually changes one's relation to the environment. Because there is no consummatory climax, satisfaction has to be seen as lying in a considerable series of transactions, in a trend of behavior rather than a goal that is achieved. (p. 322)

These are words that one would expect to find in recent texts on so called "situated activity". (But see (Maes, 1990b) for apostasy within this community.)

A stance needs to be taken in relation to such arguments, since they do bear on the representation of goals. However, this thesis is not the place for a survey of these fundamental arguments. Instead, one of the clearest positions on these matters is described and evaluated: (Dennett, 1987). Dennett's work is chosen instead of that of AI researchers such as (Agre, 1988; Agre & Chapman, 1987; Brooks, 1991a), because in my opinion his arguments are much more sophisticated than theirs. However, his work and that of his philosophical sparring partners (e.g., Fodor, Churchland, and Clark) are very technical and intricate. Dennett himself characterises the literature as follows: "the mix of contention and invention in the literature [on propositions] [...] puts it practically off limits to all but the hardy specialists, which is probably just as well. Others are encouraged to avert their gaze until we get our act together." (Dennett, 1987 p. 205). I nevertheless succumb to the temptation of having a cursory glance at this literature.

The Intentional Stance contains a particular class of arguments concerning (1) the interpretation of intentional terminology and (2) the different ways information can be stored, manipulated, and used in a system. It is important to view these two classes of argument as potentially standing separately. G. Ryle (1956) argued that motives are a particular sort of reason for acting (based on a kind of disposition), and neither an occurrence nor a cause of action. Dennett (1987) has developed Ryle's arguments.

Dennett claims that intentional terms in general are simply used by people as tools to predict and interpret behaviour on the basis of knowledge of their beliefs and desires, and not as terms referring to internal mental states, events, or processes. His claim is partly based on the belief that people do not have access to (nor, presumably, theories about) the design of each others minds, and hence that lay people cannot adopt a "design stance" with respect to one another. It is also based on analogies between intentional terms and physical "abstracta", things that are not real but useful for prediction (e.g., gravity). Just as to be five foot tall is not to be in a particular internal state, to believe that Jon is happy is not to be in a particular state either. Yet either concept can be used predictively.

Dennett further argues that (propositional) representations should not be used to model psychological mechanisms, but to model the worlds in which they should operate. One of Dennett's main justifications of this claim is that he believes that representationalist theories cannot cope with inconsistencies in beliefs. In particular, he thinks it is difficult for them to explain behaviour when it breaks down, when it appears irrational. For in such cases, it often seems as if a person believes things which are inconsistent. Some of Dennett's more technical arguments have to do with philosophical difficulties in specifying the relationship between intentional structures—which are in the mind—and their referents—which may be external to the mind (Dennett, 1987). Dennett takes the example of a calculator which though it embodies rules of mathematics, it does not refer to them or use symbols (except in the input and output stages). He claims that much mental processing might be of "that nature".

Dennett's arguments provide a useful reminder that one should not assume that there is no problem in using intentional representations when designing cognitive systems. A related but distinct thesis, which is in some respect more general than Dennett's, is that the concepts of ordinary language are often both imprecise and inconsistent and that they must be used with caution. For instance, our concepts of personal identity and life do not permit us to decide whether tele-transportation—the process of copying a person's molecular composition, destroying it, and building a "new" one—involves killing the individual or not. However, this does not imply that we cannot benefit from progressively reformulating these terms. The reformulations can be judged more on the basis of practical scientific usefulness than consistency with previous terminology (compare Kuhn). Dennett is well aware of the role of conceptual analysis; nevertheless, as is argued below, his proposal to eradicate intentional constructs from designs of systems seems premature.

In principle, Dennett could give up his proposal to eradicate intentional constructs from designs while maintaining the thesis that intentional constructs can be interpreted behaviouristically, on the basis that they buy one predictive power, and even that they have some measure of "reality". (Dennett, 1988 pp. 536-8, argues that his view is not strictly instrumentalist.) For, it does not follow from the fact that behaviouristic interpretation of terms is very useful and that it is in a sense real ("abstracta") that representationalist interpretations are empirically false, philosophically untenable, or that they lead to poor designs: i.e. the two tenets need not be mutually exclusive.

R. S. Peters (1958) critically notes that Ryle lumps together a multifarious compilation of concepts under the dispositional umbrella term "motive". Dennett posits an even broader category of "intentional idioms". Dennett motivates his intentional stance not only as an account of beliefs, desires, and intentions, but of folk psychology in general, including preferences, goals, intentions, interests "and the other standard terms of folk psychology (Dennett, 1987 p. 10). What regroups these terms together? Previous philosophical work answered this question by saying that they (or at least some of them) were intentional in that they had components that referred to something. Dennett does not allow himself the luxury of grouping these terms in the conventional way, yet he refers to a category that is co-extensive with the traditional one, and it does not seem clear that he has a proper category which encompasses them. Intentional now means "folk psychological", which means "useful for predicting and interpreting behaviour". But what about moods, attitudes, personality traits, and other categories classified above? Although Dennett does not provide an analysis of these categories, he boldly assumes that they are all to be distinguished strictly in terms of how they are used to predict behaviour. Yet, conceptual analysis suggests that some of these terms are not even "intentional" in the sense of previous philosophers. For example, currently some researchers believe that moods have little or no semantic content but can best be understood in terms of the control they effect (Oatley, 1992; Sloman, 1992b).[1] As was suggested by Sloman (1992b) and noted above, control states differ in the precision or extent of their semantic content.

Moreover, although Dennett claims that taking the intentional stance buys one predictive power, he does not provide us with rules to make these predictions, nor does he list this as a topic for future research.

It is not evident that models which use intentional constructs cannot account for inconsistencies in beliefs. For instance, in a society of mind theory (Minsky, 1986), it is not impossible for two agents to have different and incompatible beliefs and desires. It is not because many theories require that beliefs or preferences be consistent that representationalist AI needs to be committed to the assumption of consistency. Even within a single module, preferences can be intransitive or inconsistent. Dennett is very familiar with work in AI. Yet he only considers a small number of possible explanations of agent level inconsistency (Dennett, 1987 Ch. 4). He provides an insufficient basis for making sweeping statements about all possible designs. For instance, he does not do justice to the broad thesis, developed in (Clark, 1989; Sloman, 1985b), that it is possible to explain mental phenomena in terms of a number of virtual machines, which use many forms of knowledge representation, some of which can be described adequately in technically defined terms of belief and desire.

---

[1]However, "moods" are notoriously very difficult to define, and it is possible that the concept is peculiar to English speaking cultures. In Québecois French, the closest term is "humeur" and it has a much narrower extension; in that language, there are only two linguistic variations of 'mood': good mood and bad mood.

This line of argumentation suggests that an important problem with Dennett's view is that it does not offer a very practicable methodology for cognitive scientists. Dennett believes that a lot of our knowledge of ourselves uses intentional constructs. Yet he does not want to allow cognitive scientists to try to tap this knowledge (except in their statement of the requirements of the system). This constraint is easy for a philosopher to obey, if he is not in the business of building models; but this is not so for a cognitive scientist. Even if the complete eradication of intentional terminology from cognitive models were ultimately needed—and that is by no means obvious—it does not follow that cognitive scientists ought not gradually to try to refine and extend intentional constructs in their models. For it is possible that this gradual refinement can lead more rapidly to good models than the alternative which Dennett proposes. In other words, part of the difficulty with Dennett is that he criticises "folk psychology" writ large on the basis of its purported inability to give accurate accounts of mental processes. He unjustifiably assumes that the choice is between a complete rejection of folk psychological categories at the design level and a complete acceptance of folk psychology at that level. But why make such a dichotomy? Is it not possible to improve some of the categories? After all, scientific physics has progressed by using and improving folk categories such as space and time. One of the most important difficulties with using folk psychological terms is that people use them in different ways. However, this does not prevent a theoretician from analysing these concepts and then defining the terms technically. In this thesis an illustration of this point is made: progress is made by providing a technical definition of the concept "goal". This definition is not a lexical one (Copi, 1986 p. 173); i.e., it is not meant accurately to reflect the meaning of the term "goal" as used by laymen.

3.4. Conclusion

In this chapter the concept of goal was expounded. A provisional hierarchy of control states was described. Goals are a subclass of motivators, and motivators. This hierarchy needs to be improved, and ways of doing this were suggested. An elaborate notion of goals was presented. The analysis suggests a richer concept of goal than has been previously supposed. Related work on purposive explanations was reviewed.

In the following chapter, processes that operate on goals are expounded.

Chapter 4. Process specification

In the present chapter, the processes that operate on goals are described. A process specification determines which state transitions are possible. This specification builds upon the concept of goal given in the previous chapters, since many processes are concerned with taking decisions or recording information about goals in terms of the dimensions and components that were given in the conceptual analysis. The present discussion is in terms of partial state-transitions rather than total state transitions. State-transitions of goals can be seen as "decisions" concerning goals, in the large sense of decision, i.e., the result of an effective decision procedure. The decisions can be of various types, including decisions that set the fields of goals, that assess the goals, or that manage the decision-making process itself. Each postulated process serves a function for the agent. This does not preclude the possibility, however, of emergent processes or locally dysfunctional processing.

Rather than bluntly presenting the specification, this chapter incrementally introduces processes. This is reflected in a succession of state-transition diagrams. This didactic subterfuge is useful for explaining the justification for the theoretical postulates. Section 4.1 distinguishes between goal generation and "management" processes, and analyses them. Section 4.2 presents an outstanding problem regarding the control of management state-transitions. Section 4.3 raises and attempts to answer the question "What limitations should there be on management processing?" Section 4.4 presents Sloman's notion of insistence filtering, which is predicated on there being limitations to management processing, and expands upon this notion. Section 4.5 summarises the states in which goals can find themselves. Ch. 4 can be read as providing requirements for an architecture. Discussion of an architecture is deferred to Ch. 5.

4.1 Goal generation and goal management

In order to expound the difficulty of the requirements of goal processes, the following process specification is given in a few stages of increasing sophistication. However, for the sake of conciseness, many of the possible specifications of intermediate complexity are not mentioned.

A simple autonomous agent might process goals according to the specification depicted in Figure 4.1. Such an agent responds to epistemic events where it notices problematic situations or opportunities by producing appropriate goals or reflex-like behaviour that bypasses normal purposive processing.[1] For example, if it perceived that a baby was dangerously close to a ditch, it might produce a goal to move the baby away from the ditch. This goal would then trigger a "goal

---

[1]Reflex-like behaviours can be purely cognitive or overtly behavioural, innate or acquired. Acquired reflexes are generally called "automatic". Since this thesis is mainly concerned with goal processing, the important conceptual and design issues concerning automaticity are not investigated. See Norman & Shallice (1986) and Uleman & Bargh (1989).

expansion" (i.e., "planning") process which determines how the system is to execute the goal. This planning could take the form of retrieving an existing solution (say if the system should happen to have a store of plans) (Georgeff & Lansky, 1986), or it might involve constructing a new plan in a combinational fashion (Cohen & Feigenbaum, 1982 part IV). Combinational planning involves considering a succession of combinations of operators until one is found that will satisfy the goal in question. Once a plan has been retrieved or constructed, the agent would execute it.

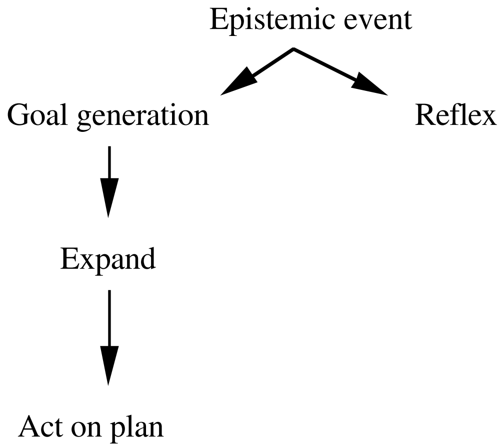

Figure 4.1. State-transitions for goals (1).

Such an agent, however, is too simplistic to meet the requirements of autonomous agency set out above. This is because, among other shortcomings, (1) it is not capable of postponing consideration of new goals; (2) it necessarily and immediately adopts goals that it produces; (3) it is not capable of postponing the execution of new goals—hence new goals might interfere with more important plans currently being executed; (4) it executes its plans ballistically, without monitoring or adjusting its execution (except to redirect attention to a new goal). Thus, a more sophisticated specification is required.

A state-transition diagram along these lines is depicted in Figure 4.2. When this agent produces goals, it does not automatically process them, but performs a "deliberation scheduling" operation which aims to decide when to process the goal further. (A more general notion of deliberation scheduling is presented below in terms of "meta-management".) If a more pressing processing task is underway, or if there does not exist enough information to deal with the goal at the time, the new goal will not continue to interfere with current tasks; instead, its consideration will be postponed. (Notice that this assumes that goal processing is resource limited. Compare section 3.2.) If the goal is to be considered now, the agent starts by determining whether the goal is to be adopted or not. Thus, if the goal is rejected the agent will have spared itself the trouble of further processing an undesirable goal. If the goal is adopted, the agent will find a way to satisfy it (as the simpler agent did). But this solution will only be executed at a convenient juncture—for the agent schedules its goals.

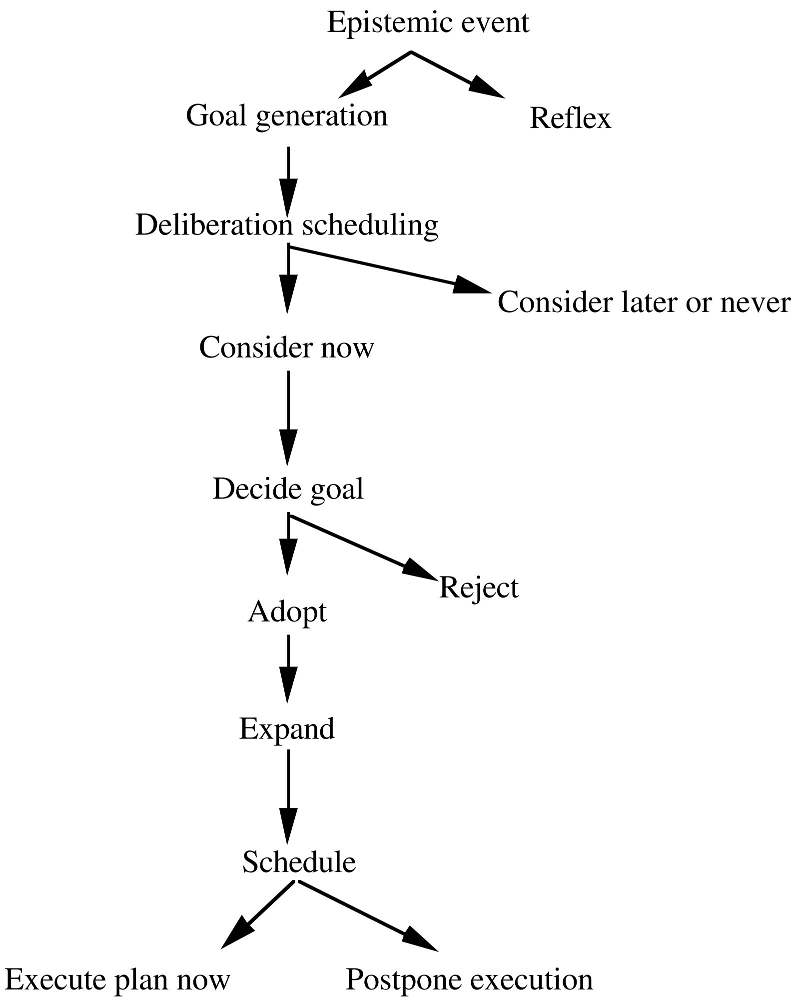

Figure 4.2. State-transitions for goals (2).

Before evaluating and improving this process specification, it is useful to propose a taxonomy of goal processes, including some new terminology.

- Goal generation refers to the production of new goal control states. There is no requirement (yet) that goals be represented as data-structures. All that is required is that the system have states that can support the goal attributes given in the previous chapter.

- Goal activation is a process that makes the goal control state a candidate for directing management processes (see below). It is assumed that whenever a goal is generated it is necessarily activated.

- Goal generactivation refers to the generation of a goal, if it does not already exist, or the activation of that goal if it does exist.

- Goal management refers to those processes involved in taking decisions about goals or management processes. The main kinds of decisions were described in the previous chapter: i.e., decisions proper, expansion, and scheduling. Taking these decisions is referred to here as the "main function" of management processing. In order to take these decisions, the system needs to be able to perform various other processes (this is referred to here as the "auxiliary functions" of management processes), including gathering information about the attributes of particular goals (importance, urgency, etc. as per Ch. 3), and assessing situations. Two other functions are part of management: the control of action and management of management processes.

Assessment of goals was discussed in Section 3.2.2.1; however, the assessment of situations, in the environment or in the agent, has not yet been discussed. An autonomous agent should be able to assess situations in order to select management strategies that are suited to the occasion and control their execution. B. Hayes-Roth (1992; 1993) presents some of the relevant situational dimensions of autonomous action: i.e., the degree of uncertainty of the environment, constraints on effective actions, availability of run-time data, and availability of a model. It is of course important to distinguish between the objective fact of the matter (e.g., what the constraints on effective action really are) and the agent's perception of these facts.

Another important dimension is the busyness of the situation. Objectively, busyness is the extent of the adverse consequences of spending a certain amount of time idling. In principle, one could characterise busyness as a temporal (possibly qualitative) function which describes the effects of idling for various periods of time. For example one could figure that if one spent 1 minute idling one might risk missing the chance to pass a message to one's friend (who is about to leave); 20 minutes idling and one would not be able to finish a letter before a meeting; and with 30 minutes idling one would be late for a meeting. Since one can be idling either in processing and/or in physical action, there may be two or three conceptions of busyness: management busyness, action busyness, or unqualified busyness. However, since management involves physical action the distinction between mental action and physical action is less relevant. Busyness can be high even if no physical action is required for one of the alternatives. For instance, one might need to decide quickly whether or not go to a banquet. If one decides not to go, no action is required of one; otherwise, immediate action might be necessary.

Specifying how information about busyness can be generated is no trivial matter. An agent will have some heuristic measures that are roughly related to objective busyness but which do not match it exactly. A subject might treat busyness as a measure of the extent to which there are important, urgent, and adopted unsatisfied (but potentially satisfiable) goals that require management—and/or action—relative to the amount of time which is required to manage the goals. No definitive function is provided in this thesis for busyness, but the concept is illustrated. One of the dimensions of busyness, in this sense, is the number of goals that are contributing to the busyness of the situation. A situation can be perceived as busy because there is one very important and urgent goal requiring attention, or because a number of urgent and more or less important goals require attention. A more dynamic measure of busyness is the rate at which goals appear in relation to the rate at which they can be processed. For instance, the problems or opportunities might appear all at once, or in a rapid succession.

Information indicating high busyness can have multifarious effects. Here are three examples. (1) It can lead to an increase in "filter thresholds", in order to decrease the likelihood of further distraction and increase the likelihood of satisfaction of current goals. See Section 4.4.1.1. (2) It can lead to an increased sensitivity to problematic management conditions, and thereby an increase in the likelihood of meta-management processes being spawned. See Section 4.2. (3) It can lead the agent to relax its criteria for successful completion of tasks and select strategies that render faster but possibly less reliable results. The third state may be called one of "hastiness". Which of these consequences follow might depend on the nature of the busyness information.

(Beaudoin and Sloman (1993) used the term "hastiness" to denote a similar concept to what is now called "busyness". A. Sloman (1994a) later remarked that the term "hastiness" is more appropriate as a definiendum of the resulting psychological state (in which an agent does things quickly without being very careful). The term "busy" has both a psychological state interpretation and an "objective" one, and is therefore more suitable than "hastiness". Moreover, like hastiness, it is neutral as to whether the goals involved are desirable or undesirable. Of course, the definition of busyness is technical and does not completely capture the tacit understanding of anglophones.)

It was said above that the control of action is a management function. That is, management processes are involved in the initiation, modulation, and termination of physical actions. The specification does allow for non-management processes to be involved in controlling actions (e.g., situation-action reflexes), though the details of this distinction are left for future research.

The specification of Figure 4.2 denoted goal management processes. One of the functions was particularly narrow. The agent was assumed to ask the question "When should this goal be processed?" This is a form of deliberation scheduling. Now that the notion of "goal management" has been introduced, this question can be rephrased as "When should this goal be managed?" Answering this question and implementing the answer is a form of "meta-management." However, meta-management has a broader function than deliberation scheduling alone; for meta-management is concerned with the control of management processing. Meta-management is defined as managing management processes (some of which might be meta-management processes). That is, a meta-management process is a process whose goal refers to a management process. The following are meta-management objectives: to decide whether to decide whether to adopt a goal; to decide when to execute a process; to decide when to decide when to execute a goal; to decide which management process to run; to decide which management process to apply to a particular goal; to decide whether to decide whether to adopt a goal; etc. The notion of meta-management processes leads to the discussion of management control in the following sub-section. (Having introduced this notion, the "deliberation-scheduling" node in Figure 4.2 should be replaced by the term "meta-management.)

(It is useful (but difficult) to a draw a distinction between (1) meta-management, which involves making "deliberate" decisions about how management should proceed, and (2) "decisions" that are implicit in control structures used by management processes. The second type of control "decisions" are decisions in the very general computer science sense of effective decision procedure. It is easier to make such a distinction when faced with a particular architecture that embodies these processes.)

4.2 The control of management processing

The process specifications depicted in the previous figures have important flaws, most of which pertain to how processing is controlled. Seven such flaws are discussed here. (1) One problem is that in human agents the order in which management decisions are taken is flexible and not necessarily the same as that given in Figure 4.2. For example, goal generactivation does not necessarily lead to meta-management—it might lead to any of the management processes, e.g., scheduling, expansion, assessment, etc. Moreover, an agent might be in midst of scheduling a goal when it decides to postpone considering it and to work on another goal instead. All this, of course, raises the question "What determines the kind of management process that follows goal activation?" More generally, "What determines the kind of management process that is dedicated to a goal at any time?" There does not appear to be a straightforward answer to these questions. The issues involved here do not seem to be addressed in the psychological literature on goal processing, which implicitly assumes a fixed order of processing of goals (e.g., Bandura, 1989; Heckhausen & Kuhl, 1985; Hollenbeck & Klein, 1987; Lee, et al., 1989). The questions are considered in more detail below.

(2) A closely related and equally important problem is that given a management process, it is not clear what determines the conclusion to which it comes. Some principles for deciding, scheduling, and expanding goals were proposed in the previous chapter, where it was said that information about importance, urgency, and instrumentality of goals (respectively) should be gathered to make decisions. However, these principles are quite abstract. The question arises whether more specific principles can be proposed.

(3) Another problem with Figure 4.2 is that it does not allow for one management function to implicate another. Whereas the various functions of management processes were described separately, they are in fact often inextricably linked. For instance, how a goal is expanded might depend on when it can be acted upon, as well as on how important it is; and when a goal is pursued might affect the chances of the endeavour succeeding. Often the process of deciding whether to adopt

a goal requires planning—at least in order to assess the cost of the goal. Therefore, executing any particular management function might involve pursuing the others. Furthermore, there is no requirement that a process be dedicated to one type of decision only.

(4) Similarly, the specification seems to imply a degree of seriality in decision-making that is not necessary. The trade-offs involved in serial vs. parallel management processing ought to be investigated. Compare section 3.2 below.

(5) The specification does not illustrate interruptability of management processes nor their termination conditions. Since management processes are to be designed as anytime algorithms (cf. Ch. 1), there need to be provisions for determining when to interrupt them and to force them to come to a conclusion.

(6) The figures do not accommodate many other types of management process that were posited as required: such as assessing situations and goals.

(7) Finally, there is an assumption that all management processes are goal directed. This assumption is subtle because goals are doubly involved. Most management processes are goal directed in the sense that they are meant to manage goals. Nevertheless, the specification allows for some processes to process other things besides goals. The process specification is goal directed in another sense: every process was described as being directed toward a type of conclusion (e.g., a scheduling decision or an assessment), as opposed to being data directed and non-purposive. This restriction is too narrow. It is sometimes useful to take high level decisions in a data-driven fashion. Indeed, people seem to use both methods, and it is convenient for the engineer to combine them (Lesser, et al., 1989). In the general case, if every process were goal directed, there would be an infinite regress and nothing could ever get done.

An improved state transition diagram is presented in Figure 4.3, which states that goal activation should lead to management processes but does not specify the order of processes, and is to be interpreted according to the requirements mentioned in this section. Whereas this view of goal processing is much more general than the previous one, it implies that quite a few control issues need to be addressed. Indeed, the difficulty of the control problems that are to be solved should be underscored. There is both an empirical problem, in knowing what determines the course of processing in humans, and an engineering problem, in knowing what are the most promising methods for directing management processing.

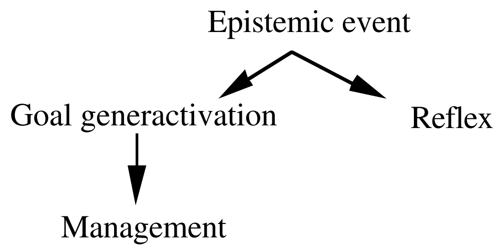

Figure 4.3. State-transitions for goals (3). This indicates that goal generactivation leads to a management process without specifying the type of m-process. The more abstract expression "goal generactivation" is used rather than "goal generation".

4.2.1 Heuristic meta-management

B. Hayes-Roth (1985) speaks of the "control problem" which is for a system to decide which of the currently possible computational actions to perform next. Solving the control problem is especially important for autonomous agents, because they must reach their decisions in good time, given the urgency and multiplicity of their goals. Now an agent cannot at every moment proceed in a decision theoretic,[1] deliberate manner, surveying the space of possible management actions to take, predicting their consequences, computing their expected "utility", and selecting the one with the highest utility. Even if the infinite regress implied by this manner were halted, this manner is too time consuming and knowledge intensive. (An exposition of decision theory is given in Ch. 6.) Instead, an agent that is capable of meta-management should only engage its meta capabilities at timely junctures where a shift in processing is required—or at least should be considered. (Compare the discussion of demon systems in (Genesereth, 1983).) Roughly speaking, these junctures can be divided into two classes: management opportunities and management problems.

Thus, there is a set of internal and external situations that can arise which require that the current management processing be redirected in some way, because otherwise time and effort will be wasted, an opportunity will be missed, or stagnation will ensue, etc. In order to make the task of managing management tractable, it is useful for an agent to be able to recognise and respond directly to such states. Some of the types of problems in management processing which an agent should be able to detect and correct are expounded below in this section. An autonomous agent that lacks the ability to respond to the following situations will perform "unintelligently" under the said conditions.

---

[1]Decision theory was originally developed to control external behaviour, but it has recently been applied to guide internal processing (Boddy & Kanazawa, 1990; Dean & Boddy, 1988; Doyle, 1989; Good, 1971b; Haddawy & Hanks, 1990; Russell & Wefald, 1991). An attempt is made to design agents which make optimal choices in a population of cases. Decision theory states that an agent aims to take control decisions which have the highest utility in that situation. Compare Simon (1959).

Sensitivity to related situations is being examined by psychologists under the subject headings "self-regulation" and "meta-cognition" (Brown, 1987; Kanfer & Stevenson, 1985; Miller, 1985).

By being sensitive to certain key problems in processing (or opportunities) an autonomous agent need not intensively monitor and analyse its management processing. I.e., its meta-management facilities need not be controlling every lower level action but need only respond to a limited set of conditions. When the problems or opportunities are detected, meta-management processing should be invoked to determine whether there really is a problem, in which case remedial responses might be elicited. The idea is progressively to identify possible problems, and for intensive verification and computation to be performed only if initial screening suggests it is needed.

Here follow five problematic conditions that should lead to meta-management. Opportunities are not covered here.

- **Oscillation between decisions**. This is when over a period of time management processes take decisions that are incompatible and that cancel or contradict previous decisions. For instance, faced with the choice between wearing a green tie and a red tie, a person might select a green tie, then change his mind and select a blue tie, and change his mind again repeatedly. Such a situation needs to be detected and resolved by some arbitration, which a meta-management process can command. In order for the decisions to be implemented some control over mechanisms that dispatch management processes needs to be exercised. This category of process should subsume cases in which physical action commences and is interrupted for some goal only to have action for the latter goal interrupted for some other goal which is possibly the same as the one driving the initial action.

- **Ongoing disruption by an insistent goal that has been postponed or rejected but nevertheless keeps "reappearing".** This situation corresponds to a manifest state of perturbance. (See Ch. 7). Such disruption might interfere with the management of important goals, and if it is detected then various means might be taken to deal with this, such as analysing the situation leading to the perturbance, satisfying the perturbing goal, or trying to prevent it from being reactivated. Neither of these solutions is necessarily straightforward. For instance, an agent who is disturbed by a motive to harm another, but who decides to reject this, might need to devise some strategies to stop considering the spiteful goals. This is a meta-management objective because the objective is produced in order to exercise control over the management. So called "volitional strategies" are expounded in (Kuhl & Kraska, 1989; Mischel, 1974; Mischel, Ebbesen, & Zeiss, 1972).

Detecting both of these kinds of problematic management conditions requires storing records of the goals that appear, and the decisions taken about them. Like (Oatley & Johnson-Laird, to appear), this theory implies that a perturbance can be detected while remaining non-analysed (i.e., the agent does not necessarily know the cause of the perturbance that is detected).

- **High busyness**. When the busyness of a situation is high, it is particularly important for prioritisation of goals to take place, and for the management to schedule its deliberation appropriately, deferring consideration of those goals that can wait, and considering the more pressing ones. This might require detecting conflicts amongst goals, and arbitrating amongst them. Thus the system should become more likely to respond to the appearance of a goal by engaging a meta-management process whose objective is to decide whether it would be best to manage this goal now, or at a future time. If the busyness is very high, it may be necessary to accelerate the process of meta-management and increase the bias toward postponing goals.

Below, a notion of goal filtering is expounded and it is suggested that filter thresholds should be high when the busyness is high. The effect of this is to keep the disruptability of management low.

- **Digressions**. A digression occurs when a goal is scheduled for deliberation, deliberation commences, but the agent loses sight of the fact that the deliberation was pursued as a means to an end, rather than for itself, or the deliberation aims to achieve a higher level of detail than is necessary. Whether a train of management is to be considered as a digression, of course, requires an evaluation of the extent to which it contributes to relevant decision-making. How is this to be detected?

- **Maundering**. Maundering is similar to digressing, the difference being that when one is maundering one is managing a goal for some length of time without ever having properly decided, at a meta-management level, to manage it. If an agent discovers that it is managing goals that are not urgent or important, but other goals are pressing, then it ought to tend to postpone consideration of the former goals.

For computational economy, heuristic ways of detecting the aforementioned problems need to be used. I.e., one cannot usually expect a system to be able to detect every occurrence of a problem; and there will sometimes be "false positives". Nonetheless, often the critical part of the work of meta-management comes not in answering the question "When should I think about this?" but in actually realising that "perhaps I should not be thinking about this". For example, it might take a person a few minutes unconsciously to realise that he might be digressing, but once he comes to ask himself "Am I digressing?" the question usually can be quickly answered. This might be because in a human being the demon for detecting digressions is not always active.

In order to be able to execute meta-management processes, a system requires a language in which to express management objectives that has in its lexicon terms referring to management processes. Organisms which do not possess such languages, which cannot produce such terms, or which do not have the mechanisms to combine them, are not capable of meta-management. Interesting empirical questions could be formulated along the lines of "What species are capable of managing their own management processes?", "What are the mechanisms that a given class of organisms has for meta-management?", "What formalisms best match their language?", "How do the meta-management mechanisms they use develop?", "What kind of variability is there within the human species?", "What pathologies of meta-management can develop?" etc. These questions might improve upon the less precise questions concerning whether other organisms have a language at all, or whether they are capable of self-reflection. The questions are particularly relevant to researchers interested in studying the space of possible designs (Sloman, 1984; Sloman, 1994c) and the relations between requirement space and design space (Sloman, 1993a).

Of course, the author has not solved the control problem. Some control conditions have been identified, but there are many other control conditions to study—e.g., opportunities. Moreover, more needs to be said about how to make the control decisions themselves.

4.3 Resource-boundedness of management processing.

It is usually assumed in AI that real agents have important "limits" on the amount[1] of "high level" processing in which they can engage (e.g., Simon, 1959). The expression "autonomous resource-bounded agents" is gaining currency, as is the expression "resource-bounded" reasoning. A large variety of implications is said to follow from the requirements of autonomous agency. Typically, they involve assuming the use of "heuristic" algorithms, as opposed to algorithms that are proven to be correct. Limits in processing play a crucial role in many theories of affect, e.g., (Frijda, 1986; Oatley & Johnson-Laird, 1987; Simon, 1967; Sloman, 1987; Sloman & Croucher, 1981). They are also said to imply that an agent should to a large extent be committed to its plans (Bratman, 1987); for by committing itself to its plans an agent thereby reduces the amount of processing it needs to do—those possible behaviours which are incompatible with its intentions can be ignored.

The issue of limits in mental resources is addressed in this thesis for two reasons. One is that resource limits have implications for designing autonomous agents—including the need for an "insistence" based goal filtering process (Sloman, 1987). See section 4.4. The author is not committed, however, to expounding the precise nature of the constraints: a general characterisation suffices for this thesis. The other is to stimulate discussion on an issue that has not been systematically explored from a design-based approach.

---

[1] The expression "amount" is just a short-hand way of referring to constraints on processing. In fact there are qualitative constraints on parallelism that can't be captured quantitatively.

Two important questions need to be asked. The first one is "What mental processes can go on simultaneously in humans?" In Ch. 2, where some literature concerning attention was reviewed, this coarse factual psychological question was broken down. It was noted that psychologists tend to assume that there are processing and memory constraints, and that empirical research must ascertain what those constraints are. A prominent empirical psychologist of attention, (Allport, 1989), upon reviewing the literature on attention, which he claims is making precious little theoretical progress, concludes that more research is needed on the function of attention as opposed to on where this or that particular bottle-neck lies. This leads to our second question, which is posed from a design stance: What limits ought or must there be on the amount of mental processing that can go on simultaneously in an autonomous agent.[1] In this section, an attempt to refine and answer this vague question is made; however, the speculative and tentative nature of the discussion needs to be underscored. The problems involved here are some of the most difficult ones in this thesis.

In order to make the question more tractable, we will focus on a particular kind of process, namely management processing. (Requirements of management processes are presented above. A design for management processes is given in Ch. 5.). So, if one were designing an agent that embodied the processes described so far in this chapter, to what extent should management processes be allowed to go on in parallel? We are not concerned with micro-parallelism here but with coarser parallelism, where different tasks are involved. Neither are we concerned with the distinction between real and simulated parallelism. We are concerned with at least virtual parallelism of management processes. This merely requires that one management process can commence before another finishes, and therefore that two management processes have overlapping intervals of execution.

If there were no constraint, then whenever a goal was generated a management process could simultaneously attempt to decide whether to adopt it and if so, to what extent it should satisfy it, how it should proceed, and when to execute it. With no constraint, no matter how many goals were generated by the system, it could trigger one or more processes to manage them, and these processes could execute in parallel without interfering with each other (e.g., by slowing each other down or corrupting one another's results). In the case of our nursemaid, whenever it discovered a problem it would activate processes to deal with them. For instance if it discovered within a short period of time that one baby was hungry, one was sick, and two others were fighting, the nursemaid could, say, prioritise these problems and then simultaneously plan courses of actions for each one of them. If there were constraints on these processes, the nursemaid might have to ignore one of the problems, and sequentially expand goals for them.

There is a general way of expressing these issues. It uses the notion of utility of computation expounded in (Horvitz, 1987). Assume for the sake of the argument that theoretically one can

[1]A related question that is sometimes asked is: Should there be any limit at all in mental processing?

compute probabilistic estimates of the costs and benefits of management processing, which are referred to as the "utility of computation". One could then ask how the total utility of computation increases as management parallelism increases. One hypothesis is that the utility of computation increases monotonically (or at least does not decrease) as the amount of management parallelism increases. Another is that, beyond a certain threshold, as the amount increases the total utility of computation decreases. There are, of course, other possible relations. This framework provides us with a convenient theoretical simplification. And it is a simplification since in practice it is usually not possible to quantify the utility of computation. Moreover, as already mentioned, there are some constraints on management processing that cannot adequately be described in terms of a quantity of management processing.

The rest of this section reviews a number of arguments that have been proposed in favour of limiting management parallelism. The review is brief and more research is required for a definitive solution to this analytical problem.

The first constraint that is usually mentioned, of course, is that an agent necessarily will have limited physical resources (chiefly effectors and sensors). Some management processes require at some point the use of sensors or effectors. For instance, in order to ascertain the urgency of dealing with a thug a nursemaid would need to determine the population density around the thug—which requires that it direct its gaze at the thug's current room. Two management processes can simultaneously make incompatible demands on a sensor (e.g., looking at one room of the nursery vs. looking at another). This implies that one of the processes will either need to do without the information temporarily, wait for a while for the sensor to become available, or wait for the information opportunistically to become available. One can imagine that in some circumstances, the best solution is to wait for the sensor to be available (e.g., because the precision of the sensor is high, and the required information cannot be obtained by inference). This implies the need to suspend a process for a while.

Now if many waiting periods are imposed on management processes, then the utility of computation might fall, to the extent that some of the suspended processes are dedicated to important and urgent tasks, since waiting might cause deadlines to be missed. Clearly, some prioritisation mechanism is needed. And in case the prioritisation mechanism should be affected by the sheer number of demanding processes, it might even be necessary to prevent some processes from getting started in case they should make demands on precious resources. This argument does not apply to processes that do not require limited physical resources. But if for some reason some severe limits are required for internal resources (e.g., memory structures with limited access) then the number of management processes requiring them also might need to be constrained.

This argument can be extended. A. Allport (1987) argues that only processes that make direct demands on limited physical resources actually need to be constrained in number. However, his criterion excludes from consideration management processes that might make indirect demands on physical resources, through "subroutines". The extension, then, is that an important aspect of management processes is that they might make unpredictable demands on physical resources. That is, it might not be possible to know before a process starts whether it will need an effector or not. For example, a person might start evaluating the urgency of a problem and discover that he has to phone a friend in order to find some relevant information. Hence one cannot easily decide to allow two processes to run together on the assumption that they will not make conflicting resource demands. This is because management processes—being fairly high level—are flexible and indeterminate and can take a variety of "search paths", and deciding which branch to take will depend on the situation. (The design of management processes in Ch. 5 will illustrate this point.) The implication, then, is that (at least in some architectures) it might be necessary to prevent the spawning of management processes in case they should claim a limited physical resource and interfere with more pressing management processes. Thus limited physical resources (and a few other assumptions) imply the need for limiting management processing.

An obvious constraint is that whatever processing hardware supports the management processes, it will necessarily be limited in speed and memory capacity, and therefore will only be able to support a limited number of management processes simultaneously. For example, there will be a finite speed of executing creating, dispatching and executing new processes, and given external temporal constraints, this might imply a limit on management parallelism. Similarly, there might not be enough memory to generate new processes. However, one could always ask of a given finite system "If it were possible to increase the speed and memory capacity of the system, would it be profitable to allow it to have more management parallelism?"

A more general argument than the latter is that there might be properties of the mechanisms—at various virtual or physical levels—that discharge the mechanisms that limit the amount of parallelism that can be exhibited. There are many possible examples of this. One example that falls in this class is, as A. Allport (1989) has argued, that an important constraint on biological systems which use neural networks is to avoid cross-talk between concurrent processes implemented on the same neural network. One can suggest, therefore, that as the number of management processes using overlapping neural nets increases beyond some threshold, the amount of interference between these processes might increase, and this might adversely affect the total utility of computation. However, since the present section is concerned with design principles (rather than biologically contingent decisions), for Allport's point to be weighty, it would need to be shown that in order to meet the requirements of autonomous agents it is necessary (or most useful) to use neural networks or hardware with similar cross-talk properties. Otherwise one could simply assume that neural nets are not to be used. Another set of examples of such constraints is used in concurrent blackboard systems that face problems of "semantic synchronisation" or the corruption of computation (Corkill, 1989). See Corkill (1989) for examples. One solution that has been proposed is temporarily to prevent regions of the blackboard (or particular blackboard items) to be read by one process during the lifetime of another process that is using it (Corkill, 1989). This is referred to as "memory locking". In other words, it is sometimes useful for regions of a memory structure to be single-read—processes wanting to read information in the region would either have to wait or redirect their processing.

Another constraint concerns the order of management processes. One might argue that some decisions logically must precede others and hence so must the processes that make them. For instance, one might claim that before deciding how to satisfy a goal one needs to decide the goal. And one might also need to know how important the goal is (so that the means not be disproportionate to the end). However, as was noted above there does not seem to be an a priori order in which management decisions must be taken. For instance, it is often (but not always) necessary to consider plans for achieving a goal before deciding whether or not to adopt it. The lack of a universal order does not imply that it is reasonable to pursue every kind of management decision simultaneously; nor does it imply that no order is more appropriate than another in a particular context. B. Hayes-Roth and F. Hayes-Roth (1979) have argued that problem solving should proceed opportunistically. This would imply that processes that can contribute to the management of goals in a given context should be activated and those that cannot should not. This is fairly obvious too. Many reasoning systems have procedures, methods, or knowledge sources that have conditions of applicability attached to them,[1] however, most of them also have mechanisms which select amongst multiple applicable procedures. The abstract question which we are dealing with here is "Why couldn't all applicable procedures run in parallel?"

It seems to be the case that the more management parallelism is allowed, the more difficult it is to ensure the coherence of management decisions, and this in turn adversely affects the utility of computation. The notion of "coherence" would need to be spelt out. It involves taking decisions that are not incompatible with other decisions (in the sense that implementing one decision does not reduce the likelihood of being able successfully to implement another decision, or increase the cost thereof); or that if such incompatibilities are engendered, they will be noted. For instance, consider a process, **P1**, that is meant to decide when to pursue a particular goal. If **P1** is operating serially it is easier to ensure that its output will be coherent with respect to other decisions if it is not running simultaneously with another scheduling procedure. (Note that assuring "coherence" can be difficult even without asynchronous management processes—e.g., because of the frame problems—and limited knowledge).

---

[1] A general notion of "opportunity" must cope with cases of graded opportunity and costs and benefits of reasoning.

Coherence is a particularly important criterion for management processing. That parallelism poses a problem for coherence is well known (Booker, Goldberg, & Holland, 1990). It has been said in (Baars & Fehling, 1992; Hayes-Roth, 1990; Simon, 1967) to imply the need for strict seriality at some level of processing. However, one could counter that there are existence proofs of systems that effectively do embody "high level" coarse-grained parallelism (Bisiani & Forin, 1989; Georgeff & Lansky, 1987).[1] It would seem, therefore, that the coherence argument needs to be made in terms of trade-offs between deliberation scheduling policies allowing different degrees of parallelism, rather than between "strict" seriality and an indefinite amount of parallelism.

One may counter that the risk of incoherence due to parallelism is not very severe, for there are already two important cases of asynchrony that are required for autonomous agents and that at least in humans are resolved in some not completely incoherent manner. One case is between management processing, perception and action. (This is taken for granted in this thesis.) The other is between management processes. That is, the system will necessarily be able (at least part of the time) to commence managing one goal before having completely managed another. The argument is that if the system has to deal with these cases of asynchrony, then it might also be able to deal with higher degrees of management parallelism. This is an implication of the kind of interruptability assumed in the requirements. Therefore, the counter to the coherence argument goes, a proper design of an autonomous agent will need to be based on a theory, **T**, of how to prevent or cope with problems of "incoherence due to management parallelism". For instance, the fact that an agent perceives changes in the world as it reasons implies that the basis for its decisions might suddenly be invalidated. This is obvious. The counter to the coherence argument then is that it is not yet clear that **T** will imply a need for severe constraints on management parallelism. It might be that quite minor constraints are sufficient for dealing with the various kinds of asynchrony (e.g., synchronising reads and writes, and establishing demons that detect inconsistency). In any case, one needs to develop such theories as **T**, and analyse their implications for management parallelism which may or may not be severe.

A final noteworthy argument has been proposed by Dana Ballard (Sloman, 1992a). In a nutshell, the argument is that in order for an agent to make its task of learning the consequences of its actions computationally tractable, it should limit the number of mental or physical actions that it performs within a period of time. The requirement of learning the consequences of one's actions is assumed to be essential for autonomous agents. The complexity of learning the consequences of one's actions can be described as follows:

---

[1]D. Dennett and M. Kinsbourne (1992) deal with philosophical issues arising from viewing the mind as a coarse-grained parallel processing system.

(1) An agent is designed to learn which of its actions are responsible for some events—i.e., to learn the consequences of its actions. Let **A** be the set of the agent's actions performed in the last **T** minutes and let **C** be the set of events which are possible consequences of elements of **A**.

(2) In principle an event in **C** might be an effect not only of one action in A, but of any subset of the elements of **A**.

(3) Therefore, the complexity of the learning task is equal to the power set of **A**, i.e., 2 raised to the power **A**.

Since the learning function is exponential, **A** must be kept reasonably small. Sloman proposed a few methods for doing this: one may abstract the features of **A**, group elements of **A** together, or remove elements of **A** (e.g., by reducing **T**, or eliminating actions which for an a priori reason one believes could not be implicated in the consequences whose cause one wishes to discover). Ballard's method is to reduce the number of actions that are performed in parallel—a rather direct way of reducing **A**.

Although Ballard's argument is not without appeal, for indeed complexity problems need to be taken quite seriously, it is not clear that his solution is the best one, or even that the problem is as severe as he suggests. Firstly, one could argue that reducing management processing is too high a price to pay for the benefit of learning the effects of management. Such an argument would need to expound the importance of learning, and the effectiveness of the other methods of making it tractable. It might suggest that abstracting the properties of the actions is more useful than reducing their number. And it would also suggest that there are some management actions which can be ruled out as possible causes (i.e., as members of **A**); compare (Gelman, 1990).

Secondly, one could argue that in most cases, causal inference is (or ought to be) "theory-driven" (or "schema driven") rather than based on statistical co-variation, as Ballard's argument supposes. This involves an old debate between David Hume and Immanuel Kant on the nature of causation and the nature of causal attribution (Hume, 1977/1777; Kant, 1787/1987). Hume believed that, metaphysically, there is no such thing as causal relations—there are only statistical relations between events. Kant, on the other hand, believed in generative transmission of causal potency. Psychologically, Hume believed that "causal inference" is illusory, and based mainly on perceptions of covariation. Kant believed that human beings can intuit causal relations. These two views have been at odds in philosophy as well as psychology, and have generated a large fascinating literature . It appears, however, that causal inference is often based on other factors besides covariation. In particular, it does not seem reasonable to assume that a causal attribution need consider (even in principle) the power set of the actions preceding an event, as Ballard's argument (axiom **2**) states. Instead, the agent can use "causal rules" or interpretation mechanisms to postulate likely causes

(Bullock, Gelman, & Baillargeon, 1982; Doyle, 1990; Koslowski, Okagaki, Lorenz, & Umbach, 1989; Shultz, 1982; Shultz, Fischer, Pratt, & Rulf, 1986; Shultz & Kestenbaum, 1985; Weir, 1978; White, 1989), and eliminate possible combinations thereof. However, the literature is too voluminous and complex to be discussed here. It suffices to say that Ballard's argument relies on a debatable assumption (axiom **2**).

Occam's criterion of parsimony is directly relevant to the discussion of this section. One may argue that if a system can meet the requirements with less concurrency than another then, other things being equal, its design is preferable. Occam's razor cuts both ways, however, and one might want to try to demonstrate that increased parallelism is necessary or that it can give an edge to its bearers. But that is not an objective of this thesis.

The preceding discussion expounded analytical or engineering (as opposed to empirical) arguments for limiting the amount of management processing in autonomous agents. This exposition does suggest that there are reasons for limiting management parallelism, but the counter-arguments raised do not permit one to be quite confident about this conclusion. The discussion did not specify or determine a particular degree of parallelism that forms a threshold beyond which utility of reasoning decreases. Such thresholds will undoubtedly depend on the class of architectures and environments that one is discussing. Despite the cautious conclusions, this section has been useful in collecting a set of arguments and considerations that bear on an important issue.

If we accept that there are limits in management processing in humans, and if we believe that they are not necessary for meeting autonomous agent requirements, they might be explained as contingent upon early "design decisions" taken through phylogeny. (Cf. (Clark, 1989 Ch. 4) on the importance of an evolutionary perspective for accounts of human capabilities. R. Dawkins (1991) argues that evolution can be seen as a designer.) The auxiliary functions of management processes (particularly those involved in predicting the consequences of possible decisions and actions) might be heavily dependent upon analogical reasoning mechanisms (cf. Funt, 1980; Gardin & Meltzer, 1989; Sloman, 1985b) that cannot be dedicated to many independent tasks at once. Analogical reasoning might itself use an evolutionary extension of perceptual processes which although powerful are restricted in the number of concurrent tasks to which they can be dedicated because of physical limits and needs for co-ordination with effectors. Therefore management processes might have inherited the limitations of vision and analogical reasoning. However, these constraints might be beneficial, if more verbal ("Fregean") ways of predicting would have been less effective. This evolutionary argument is merely suggestive and needs to be refined.

None of the above provides specific guidelines for constraining management processes. More research is required to meet that objective. In particular, it has not been shown that at most one management process should be active at a time. Nevertheless, there does seem to be a need for some limits on management processes; hence, the design to be proposed in the next chapter will assume that there must be some restrictions, but not necessarily strict seriality.

4.4 Goal filtering

It is assumed that not all goals that are generated or activated will necessarily be immediately considered by management processes, but might be suppressed (filtered out). An important rationale for goal filtering has been proposed by Sloman. In this section, Sloman's notion of filtering is described, while particular care is taken to dispel some common misconceptions about it. In the next section, some other roles which the filtering mechanism can play are proposed.

Sloman assumes that when a goal is generated (or activated) and is considered by a management process this may interrupt and at least temporarily interfere with current management process(es) and physical actions that they may be more or less directly controlling. This causal relation is supposed to follow from (a) the need for immediate attention, and (b) limits in management processing (the rationale of which was discussed in the previous section). This interference can have drastic consequences. For instance, if a person is making a right turn in heavy traffic on his bicycle and he happens to "see" a friend on the side of the road, this might generate a goal to acknowledge the friend. If this goal distracted his attention, however, it might lead him to lose his balance and have an accident.[1] For such reasons, Sloman supposes a variable-threshold goal filtering mechanism that suppresses goals that are not sufficiently important and urgent, according to some rough measure of importance and urgency. Insistence is defined as a goal's ability to penetrate a filter. The filter threshold is supposed to increase when the cost of interruption increases. Suppressing a goal does not mean that the goal is rejected. It only means that the goal is temporarily denied access to "higher-order" resource-limited processes.

When is goal filtering required? A. Sloman (1992b) says:

> This mechanism is important only when interruption or diversion of attention would undermine important activities, which is not necessarily the case for all important tasks, for instance those that are automatic or non-urgent. Keeping the car on the road while driving at speed on a motorway is very important, but a skilled driver can do it while thinking about what a passenger is saying, whereas sudden arm movements could cause a crash. However, in situations where speed and direction of travel must be closely related to what others are doing, even diverting a driver's attention could be dangerous. So our theory's focus on diverting or interrupting cognitive processing is different from the focus in Simon and the global signal theory on disturbing or interrupting current actions. (Section 10)

---

[1]An entire paper could be dedicated to elucidating this example and considering alternative explanations. The notion of suppression of motivational tendencies has a historical precedent in psychoanalysis (Erdelyi & Goldberg, 1979; Erdleyi, 1990) and is accepted by some theorists of pain (Melzack & Wall, 1988 Ch. 8 and 9). Colby (1963) describes a computer model of defence mechanisms. (See also Boden 1987, Ch. 2-3).

A subset of the cases in which preventing distraction might be important is when a rare and important opportunity requires attention (such as when a thief suddenly gets to see someone typing in a password to an expense account).

The notion of filtering calls for a new term referring to a goal attracting attention from a management process. This is called "goal surfacing". That is, a goal is said to "surface" when it successfully penetrates a filtering process. If the goal is unsuccessful, it is said to be "suppressed". Goal suppression is different from goal postponement. Goal postponement is a type of meta-management decision.

The requirement of filtering critically rests on limitations in "resources", where initiating one mental process might interfere with some other mental process. A detailed specification of how to know whether and when one process will interfere with another is needed. This would require proposing a computational architecture of goal processing. It is probably not the case that every design that meets the requirements of autonomous agents will be equally vulnerable to adverse side-effects of goal surfacing. One can imagine designs in which a system can perform complex speech analysis while driving a car in acutely dangerous circumstances. If an architecture allows some management processes to be triggered in a mode that guarantees that they will not interfere with others, then under circumstances where diverting a management process might be dangerous, non-pressing goals that appear could trigger non-interfering management processes or processing by dedicated modules (e.g., the cerebellum in humans?). Such goals would not be suppressed in a simple sense.

The situations in which Sloman says filtering would be useful all have the characteristic that even brief interruption of management processes could have important adverse consequences. Since the goal filters have the purpose of protecting management processes, it is crucial that they cannot invoke the management processes to help decide whether a goal should be allowed to be managed (that would defeat the filters' purpose). Filters must make their decisions very rapidly. This is because if the goals that are attempting to penetrate are very urgent, they might require attention immediately.

Sloman (personal communication) points out that none of this implies that computing insistence should not use highly complex processing and powerful resources. The only requirement is that the insistence-assignment and filtering mechanisms (which may be the same) act quickly without interfering with the management. Consider vision in this respect, it uses very sophisticated and powerful machinery, but it can also produce responses in a relatively short period of time (compared to what might be required, say, for deciding which of two goals to adopt or how to solve a peculiar problem). Sloman therefore emphasises that insistence and filtering mechanisms can be "computationally expensive".

It is easy to misunderstand the relation between insistence and filtering. A reason for this is that a system which is said to have goals that are more or less insistent, and that performs filtering, might or might not actually produce insistence measures. Consider two models involving filtering. In the first, a two stage model, one process assigns an interrupt priority level to a goal (this is the insistence assignment process) and another process compares the priority level to the current threshold, and as a result of the comparison either discards the goal or else puts it into a management input queue and interrupts the management process scheduler so that it receives some management processing. For instance, suppose that when our nursemaid hears a baby wailing, it creates a goal to attend to the wailing baby. Suppose that the nursemaid has a simple rule that assigns an insistence level to such goals: "the insistence of the goal to attend to a wailing child is proportional to the intensity of the wail". Suppose that the rule contains an explicit function that returns a number representing an insistence priority level. So, in this model insistence assignment and filtering are different processes. In the second model, filtering (i.e., the decision of whether or not a particular goal should surface) is based on rules that may be particular to every "type" of goal (if there are types of goal), and no explicit priority level representing the importance and urgency of a goal is computed. For instance, one such rule might be embodied in our nursemaid who responds to the intensity of wailing of babies. The system might filter out any goal to respond to a wailing baby if the wailing is below a certain intensity. In such a system, it might still be possible to talk about the goal's insistence; the insistence, however, is not computed by the system, nor is it explicitly represented.

Sloman also writes "Attention filters need not be separate mechanisms: all that is required is that the overall architecture ensures that the potential for new information to interrupt or disturb ongoing perceptual or thinking processes is highly context sensitive" (Sloman, 1992b p. 244). Therefore not only insistence but also filtering can in a sense be "implicit".

There is a subtle difference between the intentional aspect of "insistence measures", and the propensity concept of insistence as such. The intentional aspect of insistence that is typically mentioned is one which heuristically represents importance and urgency. This applies also to qualitative "measures" of importance and urgency. Such measures can in principle play other roles in a system besides determining insistence as a propensity; and they might be evaluated as more or less correct (in this respect they are at least implicitly factual). It is not correct to define insistence as a heuristic measure of importance and urgency. As was said above, some systems can have goals that can be said to be more or less insistent even if they do not produce insistence measures. Information is given the attribute "insistence" because of the role that it plays.

Sloman's actual definition of insistence is "the propensity to get through attention filtering processes and thereby divert and hold attention" (Sloman, 1992b). With this dispositional notion of insistence one can make counter-factual conditional statements regarding a goal, by saying for instance that "the goal was very insistent and it would have surfaced had it not been for the fact that the threshold was high". The dispositional notion of insistence can be very subtle in complex systems, and might require (for an adequate characterisation) that one move beyond speaking in terms of a goal being "more or less insistent" to describing the factors that would have contributed to its management in slightly different conditions, and the reason why it did not surface. One might also refer to the likelihood that the filter be in a state in which a goal with the given "insistence profile" can surface. For instance, consider a system that evaluates all goals on dimensions **A**, **B**, **C**, and **D** which might be said to comprise "insistence measures". The goal might have high measures on all dimensions but **D**; suppose it was suppressed because the filter has a rule **R** that "the goal must have high measures on dimension **D**". The system might also have a number of other rules which express requirements along the other dimensions. One might say that "this goal was very insistent". Since insistence is a dispositional notion, this statement is valid, for one understands that if only **R** had been relaxed (and perhaps only slightly), the goal would have surfaced (other things being equal). However, if it so happens that in the system in question **R** is always operative, then one might say that the goal was not insistent, because it could not have surfaced unless its measure on **D** was much higher. (Or **R** might be mutable in principle, but provably immutable in practice.) A theorist who desires in depth knowledge of the behaviour of such a system will require a language to describe insistence that reflects the components that are involved.

Figure 4.4 contains a state-transition diagram which indicates that goal filtering precedes goal management.

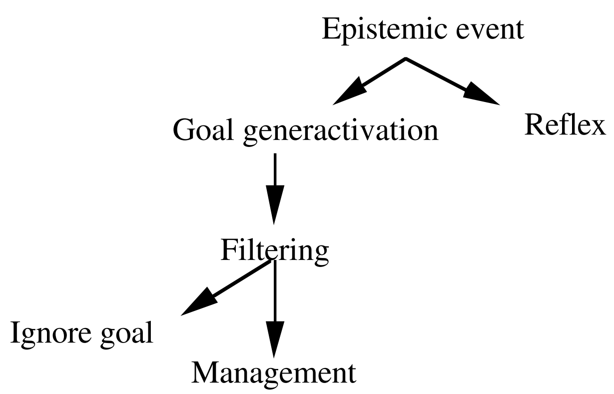

Figure 4.4. State-transitions for goals (4). Same as Figure 4.3, except that goal filtering follows goal generactivation.

In order to distinguish the role of filtering described in this section from other roles, the former will be referred to as "acute management protection", because the idea is that filtering should prevent drastic side-effects that can happen if a goal surfaces if only briefly. The processes involved in generactivating goals asynchronously to management processes, assigning insistence, and performing insistence filtering are called "vigilational processes", in contrast with management processes. The term "vigilation" is used because in effect these processes imply a readiness to redirect attention in agents that have them.

It is expected that as different designs that support insistence and filtering are developed, these concepts will be modified and improved.

4.4.1 Other functions of filtering

Related requirements can be served by filtering. All of them have in common the idea that when the cost of interruption of management by a goal is high, the filter threshold should be high. Designs that satisfy the following requirements must still satisfy all of the requirements mentioned above, especially that filtering should be done quickly and without disrupting management processing. It should be said about the following requirements that like other requirements, they are hypothetical. As such they are subject to refutation and qualification. Moreover, the following requirements are not completely independent and might overlap.

4.4.1.1 Busyness filter modulation

One requirement is that when the busyness of a situation is high, the system should become more inclined to suppress consideration of goals that are "trying" to surface unless it has reason to believe that some overriding problem is likely to surface. (Busyness was explained in section 4.1.) Let us call this "busyness filter modulation". Recall that a situation is busy to the extent that there are urgent and important goals that are being processed that require more time than is available. These conditions are different from the "acute" ones, in which a split second distraction could have drastic consequences. This is because when the busyness is high, the system might not be likely to suffer major consequences from engaging management processes for currently irrelevant goals; the management simply can itself decide to postpone consideration of the goal. Nevertheless, the system might suffer from repeated distraction from many irrelevant goals. By increasing its resistance to distraction, the system is taking the gamble that other goals that might be generated during this period of high busyness are not as likely to be relevant, and that if they are relevant that they will be sufficiently insistent to surface.

Recall that apart from the importance and urgency of the current goals, there is another dimension of variation of busyness, namely the number of current or pending goals. For example, a situation can be busy because there is one very important and urgent goal or because there are many moderately important and moderately urgent goals, etc. For the same level of busyness (in terms of importance and urgency of the contributing goals), the fewer the goals that are contributing to the busy situation, the less likely it is that a more important goal than one currently being considered will surface (other things being equal). This is because the goals being considered will be relatively important and urgent; whereas, for the same level of busyness if many goals are being considered then it is more likely that a goal that surfaces will be more pressing than one of the goals contributing to the busyness of the situation. Therefore, a potentially useful rule is that for the same level of busyness, busyness should have a greater effect on thresholds in situations where the number of urgent goals is smaller.

A simpler rule to use, which was suggested by A. Sloman (1994a), is that as the rate at which new goals arrive in relation to the rate at which they can be processed increases, the filter threshold should increase. This has the advantage that "detecting that the frequency of interrupts by new goals has exceeded some threshold may be easier than detecting other dimensions of [busyness]". In particular, this does not require computing the importance of the current goals. Analysis and simulations are required to determine how best to allow busyness to modulate filter thresholds.

The author does not mean to imply that the main effect of beliefs about current or expected busyness should be to modulate filter thresholds. Indeed, this is a relatively minor function of knowledge about busyness. There are difficult issues to address concerning how busyness should affect the system's management, such as the time windows that it gives itself for managing goals (how it controls its anytime algorithms), how it controls its perceptual processes to scan for possible problems which its beliefs about "expected busyness" imply could arise, whether it should favour quick plans for action over slower ones which might otherwise be preferred, etc.

4.4.1.2 Filter refractory period

A principle that is implicit in the previous section is that it might be problematic for the management processes to be interrupted too frequently. This might cause erratic processing and "instability". In order to decrease the likelihood of this, it might be useful briefly to increase the resistance of the filter after a goal surfaces. This is analogous to the relative refractory period of neurones, during which stimulation of a higher intensity than the normal threshold is required for triggering an action potential. The intention is not for the refractory period to involve complete intransigence to potential distractions (which is referred to as an "absolute refractory period"), although implementation issues might well imply the need for an absolute refractory period.

Applying the concept of refractory periods to psychological processes is not without precedent. M. M. Smyth et al. (1987) review literature concerning psychological refractory periods in "attentional" processes. Smyth and associates mention a variety of types of refractory periods (and reasons for them) that have been proposed. In a generalisation of a hypothesis presented by Smyth and associates, one assumes that there is a "decision-making process" that is serial and comprises successive non-interruptable sequences of processing (interruptions are delayed until the end of the current sequence). When decision-making starts, its first sequence is executed. The refractory period of the decision-making process varies as a function of the length of each component sequence. Such hypotheses have been investigated empirically in domains in which subjects are given tasks that they must commence upon presentation of a stimulus. Response to a stimulus is delayed by a predictable amount if the stimulus occurs soon after the commencement of another task. Existing psychological hypotheses are different from the current one in that (1) they assume an absolute refractory period rather than a relative one. (They do not even distinguish between absolute and relative refractory periods.) (2) They seem to assume that refractory periods are unintended side-effects of a design rather than functional aspects of a design.

4.4.1.3 Meta-management implementation

As was said in a previous section, the management ought to be able to take decisions to the effect that the consideration of a goal should be postponed, or that a goal is to be rejected and no longer considered. An example of this is if a nursemaid realises that it cannot recharge a baby because its battery is broken and it has no way of fixing it. (In fact, the current nursemaid scenario does not allow batteries to break.) The nursemaid might therefore decide no longer to try to find ways to satisfy the goal, or even that it should not try to manage it any further. The question arises, however, "How can such decisions be implemented in an agent?" In particular, an agent might want the goal to become less insistent, for if the goal remains insistent, then it will keep surfacing even after its consideration has been postponed—the management's decision to postpone it will have been ineffectual. In our example, this goal might keep resurfacing and thereby activate management processes to try to satisfy it. This might interfere with the processing of other goals which are equally important but which are much more pertinent since they can be solved.

Therefore it appears that there needs to be a link between management processes and vigilation mechanisms. For instance, the mechanisms that determine how a goal should be processed once it surfaces could be biased so that when this goal surfaces it triggers a meta-management process that examines information about the decisions that have been taken about the said goal and if that information indicates that the goal is "fully processed" then it should loop indefinitely, or simply terminates once it starts. So long as this management process does not interfere with other management processes, then this mechanism would work. However, not all architectures will offer this option, particularly if the user of the model feels that management processes need to be limited in number (for reasons mentioned above). An alternative response of course is to increase the filter threshold and hope that the generated goal simply is not sufficiently insistent. But this method is too indiscriminate, since it will affect all other goals across the board. Yet another method is to (somehow) ensure that this goal does not get generated or activated anymore in the first place.

A better method (call it **M**) is to allow management processes to tell the filter to suppress—or in other circumstances, be less resistant to— particular goals or classes of goals. If there is a mechanism that assigns numeric insistence measures, then an equivalent method to **M** is to get this mechanism to vary the insistence of the goal whose consideration has been postponed should it be activated. In our example, the filter could be made to suppress the goal to recharge the baby in question. Even if some process assigned high insistence measures to it, the filter might contain a special mechanism to prevent this particular goal from surfacing. A system that learns could train itself to refine the way it determines insistence of goals such that eventually meta-management input to the filter is no longer required. For example, an actor or ceremonial guard whose job does not permit sneezing or scratching at arbitrary times might somehow train the sub-systems that generate itches or desires to sneeze not to assign high insistence in situations where that would be counter-indicated. (One would have to determine how suitable feedback could be given to the vigilation mechanisms to evaluate its decisions.)

The concept of meta-management control of goal filters can be illustrated by a metaphor of a human manager with a secretary. The secretary can be seen as the filter. The manager might give various filtering instructions to her secretary. For instance, she could tell him that she does not want to take any calls unless they concern today's committee meeting; or that any advertisement letters should be put in the bin; or that if person **X** comes to see her he should be let in immediately. These instructions might turn out to allow some irrelevant distractions (e.g., **X** comes in but merely wants to chat); or filter out some relevant information (e.g., an advert for very affordable RAM chips which the manager needs to purchase). Some of this might lead to finer tuning of the filter in the future (e.g., the manager might tell the secretary next time "Only let **X** in if he has information about **Y**"). And the secretary might have some other basic rules of his own; e.g., if the caller is a reliable source saying that there's a life threatening emergency, then let them through. Notice that all of the rules given here are qualitative. Filtering need not be based on quantitative measures of insistence.

Meta-management filter control appears to suit the purpose at hand, but there are a number of possible objections and caveats that must be considered. One caveat is that since the basis for the meta-management's decision might be invalidated (e.g., because an opportunity arises) the system ought not to become totally oblivious to goals that it wants to be suppressed. This is not incompatible with the idea of selectively increasing the threshold for a particular goal (or goal type).

At first glance it might seem that meta-management filter control defies the purpose of filtering since it involves the use of management processes, and management processes are exactly the ones that need to be protected by the filter. It is true that this method involves the input of management; however, it is crucial to note that this input is not requested by the filter. That is, the filter does not call a management process—say as a subroutine—in order to decide whether a goal should surface or not. Instead, the filter merely consults information that has already been stored in it. If no information concerning this goal is available to the filter, then the decision is made on the basis of numeric insistence measures (or whatever other bases are normally used). Therefore, not only is the management not invoked, but the filter does not have the functionality that is required of the management.

The proposed filtering mechanism is not suitable for all designs. In simple designs it will be relatively "easy" to determine that a goal that is being filtered is of the type that the management has asked to suppress. In more complex designs, two difficulties arise. The first occurs in systems that can express the same goal descriptor in a variety of ways but that do not use a standard normal form for descriptors. For instance, in the design presented in the next chapter, seeing a baby close to a ditch generates goals of the standard form "**not(closeTo(Ditch,Baby))**". A different system with greater expressive flexibility might respond to the same situation by producing goals such as "**farFrom(Ditch, Baby)**", "**closeTo(SafeRegion, Baby)**", etc. Whereas these goals are syntactically different they might be considered by the management processes (given its knowledge of the domain) to be semantically the same. The problem is that the filter might not be able to recognise this identity. Notice that the problem of recognising identity of a "new" goal and one that has already been processed also applies to some management processes; the difference is that vigilational mechanisms have fewer resources to use. The second source of difficulty is that some systems might respond to the same situation by producing a number of goals. In this case, the goals are not simply syntactically different, they are semantically different but have the same functional role in the system. For instance, in the scenario in which a baby's batteries are broken this might generate a wide variety of sub-goals, e.g., goals that are different means of fixing the batteries. However, it might be beyond the capabilities of the vigilation processes to recognise the functional equivalence between goals.

At this juncture, it is important to note another rationale and requirement for insistence filtering: separating different functional components. It is important for goal generators and insistence mechanisms to be somewhat independent from management. The vigilation mechanisms need to be able to increase the insistence of certain classes of goals regardless of whether the management processes want them to be suppressed. This is often (but not always) useful for categories of important goals, where the designer (possibly evolution and/or learning) knows the circumstances under which they are likely to be relevant and urgent, but where the management processes might err in assessing them along these dimensions. Obvious examples of this are the "primary motives" of hunger, thirst, sex, etc. A person might decide that he will not eat or think about eating for a month. But he will not be able to implement this decision: the goal to eat will be activated with increasing insistence as time goes on. This might not prevent him from fasting, but the goal to eat will not be suppressed effectively. According to P. Herman and J. Polivy (1991), when people fast they engage in "obsessive thinking about food [...] their minds, as a consequence, [come] to be monopolised by thoughts of food, including fantasies of gourmet meals past and to come, and plans for their future career as chefs" (p.39). This holds whatever training people use (e.g., meditation is not effective). If people have goal filters, it seems that they cannot control them as easily, say, as they can move their arms. Evolution has discovered that it is best to make it increasingly difficult for management processes to postpone the goal to eat as a function of time since the last meal and other variables. So, not only should the goal generators and filters operate without disrupting management or performing the same kinds of processes that the management executes, they should be resistant to some forms of direct manipulation by the management. (The same can be said of pain generators, and other sources of motivation.)

The task of the designer is to discover a satisfactory (but not necessarily optimal) compromise between hard and fast rules and the ability of the management through its "higher level powers" to by-pass and possibly inhibit or modify them. The designer's decision needs to be based on the requirements that the system has to satisfy. There is no absolute rule that holds for all environments and all designs concerning the ways in which filtering mechanisms can be controlled by management processes. Nevertheless, researchers should try to refine the rules thus far presented. If their efforts fail, it could be argued that only learning mechanisms can solve the problem of finding suitable compromises for individuals in specific environments. If this were so, theoreticians would nevertheless have an interest in studying the compromises produced by learning mechanisms, in the hope that principles—of various degrees of generality, to be sure—could be extracted from what on the surface appear to be idiosyncratic solutions.

So far in this section the focus has been on engineering considerations. Sloman argues that even if it were good in some engineering sense for human beings to have greater control of insistence processes than they do, it might be that because they evolved at different junctures the vigilation processes are separate from management processes. That is, this separation might have evolved contingently, without offering an evolutionary advantage.

> Why can't my tongue reach my left ear? It's just too short. I can't say that evolutionary and survival considerations explain why my tongue isn't much longer. Similarly if an architecture happened to evolve with certain limitations, that need not be because it would have no value to overcome those limitations. I think some things have limited access to higher level information simply because they evolved much earlier, and originally needed only access to particular sub-mechanisms. E.g. detecting shortage of fluid and sending a signal to the brain may be done by a primitive mechanism that simply can't find out if the corresponding goal has previously been considered and rejected or adopted. (Personal communication, 25 Nov. 1993)

That is, not all extant (or missing) features of an architecture are there (or absent) for a good engineering reason, some are just side-effects of the way it developed phylogenetically. (Compare Clark, 1989 Ch. 4).

The empirical example of hunger was given above as an instance of a useful inability to control a module. However, there are other examples where the inability does not seem to be that useful. States that are described as emotions often have the characteristic that a goal (or a cluster of goals and "thoughts") tend to surface even if the management would prefer to not be distracted by them. (Sloman and Beaudoin refer to these states as "perturbance".) One may consciously and accurately believe that the goal is causing more damage than it can possibly cause good. Consider for example the hypothetical case in which a tribal man, **M1**, covets a woman who is married to a man who is in a much higher social stratum than he. **M1** might accurately believe that if he acts on his desires, he will run a severe risk of being executed, say. For the sake of the argument, we can suppose that the man has a choice of women in relation to whom he does not run the risk of punishment (so a simple argument in favour of selfish genes fails). Thus **M1** decides to abandon his goal and to stop thinking about the woman; in practice, however, there is no guarantee that his meta-management intention will be successful, even if his behavioural intention is. It might be that this disposition does not favour the individual but favours his genes. (Compare Dawkins, 1989).

In sum, some measure of management control of vigilation processes is useful for implementing meta-management decisions. But in autonomous agents such control is not (or should not be) unconstrained. Most meta-management decisions do not need to be implemented by modulating the goal filter. Yet most research on meta-level reasoning has not even used the concept of filtering.

4.5 Summary of goal state specification

Given the above process specification, it is now possible to provide more terminology to describe goal processes, and some constraints on goal processes. This will be particularly useful for the discussion of architectures in future chapters.

In this process theory, activation is a qualitative attribute of a goal's dynamic state that expresses a relation between the goal and processes that operate on it. A goal, **G**, might be a focal or contextual object of a management process. **G** is said to be a focal object of a management process, **P**, if **P** is trying to reach one of the management conclusions regarding it. **G** is a contextual object of **P** if **P** has some other goal(s) as its focal object(s), and if **G** figures in the deliberation of this management process. For instance **P** might be a process of deciding whether to adopt a goal. This goal would be the "focal goal" of **P**. The goals with which it is compared would be contextual goals.

Goals can dynamically change state between being focal and contextual while a process is executing (typically this would be through invocation of subprocesses).

The theory allows for a goal to be in one or more of the following states of activation at a time (these are predicates and relations, not field-accessing functions):

- **filtering-candidate(Goal)**. By definition a goal is a filtering candidate if it is about to go through a process of filtering, or is actually being filtered (as described above).
- **asynchronously-surfacing(Goal)**. A goal that is surfacing has successfully passed the filtering phase and is about to be actively managed (this subsumes the case of a "suspended" goal being reactivated e.g., because its conditions of re-activation have been met). This is also called "bottom-up" surfacing.
- **synchronously-surfacing(Goal)**. Such a goal has arisen in the context of a management process's execution (e.g., it is a subgoal to one of the management processes' goals). This is also referred to as "top-down" surfacing.
- **suppressed(Goal)**. A goal is prevented from surfacing by a filtering process.
- **actively-managed(Goal, Process)**. A goal is actively managed if it is the focal object of a currently executing (and not suspended) management process.
- **inactively-managed(Goal, Process)**. Since management processes can be suspended, it is possible for a goal to be a focal object of a suspended management process. In this case the goal is said to be inactively managed by the process.
- **managed(Goal, Processes)**. A goal is managed if it is actively or inactively managed by a process.
- **off(Goal)**. By definition a goal is "off" if the aforementioned predicates and relations do not hold in relation to it.

Goals that become an object of a management process without being filtered are said to be "recruited" by that process. This is referred to as a top-down process. It is assumed that a goal cannot jump from the state of being "off" to being managed, unless it is recruited by a management process. Goals that surface and trigger or modify a management process are said to "recruit" that management process. This is referred to as a bottom-up process.

4.6 Conclusion

The picture of goal processing provided in this chapter points towards an architecture with a collection of abilities of varying degrees of sophistication. These abilities span a range of areas in AI, such as prediction, causal reasoning, scheduling, planning, decision-making, perception, effector processing, etc. The picture is not complete, however. In particular, it is not yet clear how management processing can best be controlled. Moreover, whereas a high level explanation was given of the links between concepts such as importance and deciding, and urgency and scheduling, the management functions have not been specified in a completely algorithmic fashion: we have general guidelines but no complete solution to goal processing. This makes the task of designing an agent difficult: we may be able to specify the broad architecture and the kinds of processes that it <u>should</u> be able to support—in this sense we are providing requirements—but many of the details of the agent (particularly its decisions rules) are not yet theoretically determined. Thus, the architecture will be broad but shallow. Nevertheless, it is instructive to try to design such an agent, as it suggests new possibilities and it demonstrates limitations in our knowledge. This is the task of the following two chapters.

Chapter 5. NML1—an architecture

This chapter describes a proposed design of a nursemaid (called NML1) which is meant to operate in the nursemaid scenario described in Ch. 1, and to meet the requirements described in the previous chapters. Some of the limitations of the design are discussed in the final section of this chapter, and in Ch. 6.

5.1 NML1—Design of a nursemaid

There are many ways to build a model that attempts to meet the requirements and specification. NML1 is a particular design proposal that embodies a collection of design decisions with different types of justification. Many of the decisions were based on the grounds of effectiveness; others were based on an attempt to explore Sloman's extant theoretical framework. A few others were motivated by empirical conjectures; however, justifying such hunches is not easy, because any particular mechanism only has the implications that it does given assumptions about the rest of an architecture. Some decisions were simply arbitrary. And some are decidedly unsatisfactory (usually because they amount to postulating a black box) and were taken simply because some mechanism needed to be proposed for the model to work at all. All the decisions are provisional; mathematical and implementation analyses are required to judge their usefulness (some high level analyses are reported in the following chapter).

Early prototypes of NML were implemented in order to help design a more comprehensive system. However, most of the design as described here has not been implemented by the author, since much of it derives from analysis of shortcomings of what was implemented. Ian Wright of the University of Birmingham is currently implementing the NML1 specification. Since we are concerned with a proposed system, the current chapter is written in the simple future tense.

Although some of the alternative ways in which NML1 could have been built and their implications are discussed in the present chapter, a more systematic exposition of the surrounding design space is relegated to Ch. 6.

As discussed in Ch. 2, procedural reasoning systems (Georgeff & Ingrand, 1989) are worthy of further investigation for meeting the requirements of autonomous agents, though there is a need to improve them and explore alternatives. For this reason, it is proposed that NML1 be designed as a procedural reasoning system. Some of the similarities and differences between NML1 and PRS are discussed throughout and summarised in Ch. 6.

The overall architecture of NML1 is depicted in Figure 5.1. It will have a simple perceptual module that will record information about the babies and stores it in the World Model, which will be distinct from the program that will run the nursery. There will be a Perceptual Control module that will direct the camera to a contiguous subset of rooms, based on perceptual control strategies and current activities. The number of rooms that can be simultaneously viewed will be a parameter of the system. There will be Goal Generactivators that will respond to motivationally relevant information in the World Model (such as a baby being close to a ditch) and the Goal Database by producing or activating goals (e.g., to move the baby away from the ditch). The interrupt Filter will be able to suppress goals, temporarily preventing them from disrupting the management. The Interpreter will find management procedures that are applicable to goals and will select some for execution, and suspend or kills others. Management procedures will be able to cause physical action through the Effector Driver.

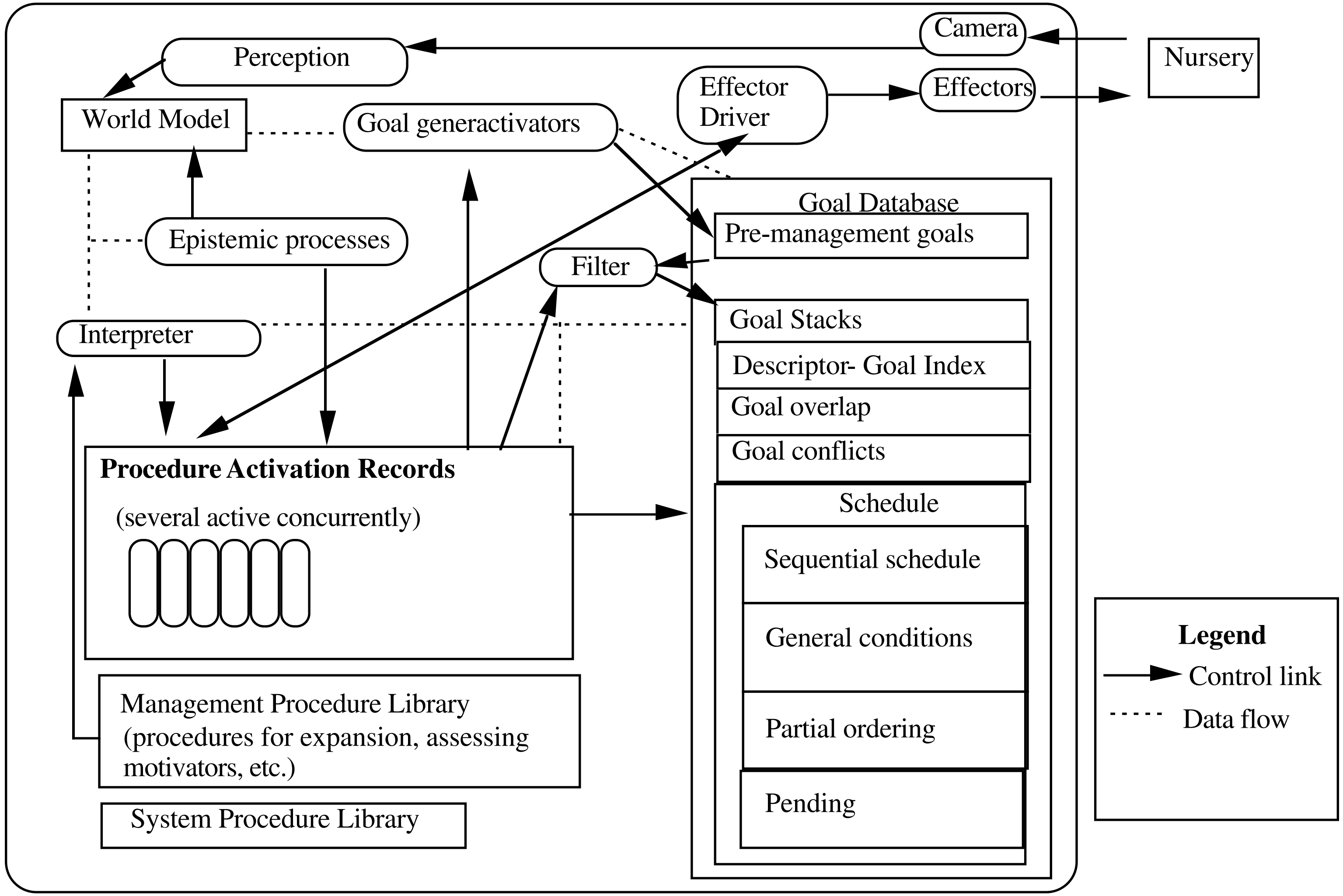

Figure 5.1 Proposed architecture of NML1. (Some of the links between modules are not displayed.)

5.2 The Perceptual Module and the World Model

Since the World Model will be distinct from the program that will run the nursery there is a possibility of information being dated and erroneous, actions having unintended consequences, etc.

Every baby actually has the following features which will be recorded in the World Model:

- A position. This will indicate the room and (**x**,**y**) co-ordinates of the baby. (According to the current design proposal, there will only be one level of position information. Approximate positions will not be represented in NML1. Still, this would be useful because information about positions quickly becomes out of date, but in principle one could have an idea of approximate location of a baby—e.g., through knowing that it would not have had enough time to move out of a part of a room.)
- Life Status. This will indicate whether the baby is dead or alive.
- Age. This will be an integer denoting the baby's age in "cycles". (Cycles are the unit of time used by the nursemaid and the simulation of the world.)
- Charge. This will be a real number between 0 and 1.
- Speed. This will represent the maximum number of steps per unit of time which a baby can take.
- IdentificationNumber. Every baby will be unambiguously identified by an integer.
- Illnesses. This will be a possibly empty list of terms denoting the baby's illnesses. There are three possible illnesses: shakes, melts, and memory-corruption.
- Injuries. This will be a list of body-parts which can be injured, possibly including the head, right or left arm, and right or left legs.
- isThug. This will be a boolean field indicating whether the baby is a thug.
- PickedUp. This will be a boolean field indicating whether the baby is picked up by the claw.

The World Model will also keep track of the co-ordinates of the claw, and its contents. The World Model will be a multiple read, multiple write data base. It will be accessed mainly by the Goal Generactivators and the management processes.

The second form of perception is a sensor attached to the claw and used locally and only by the Execution Device. No matter what the size of the room the claw sensor will only detect items that are within a 9 unit square centred on the claw. (One unit is the space taken by a baby and/or a claw.) Within this area, the sensor will be able to determine the identification number and locations of the babies and the contents of the claw. The distinction between the two forms of perception is useful because the Execution Device requires accurate information in order to determine whether actions are successful or not.

5.3 The Effector Driver

The Effector Driver (ED) will interface between the NML1 cognitive architecture and its two effectors: the claw and the camera. It will receive inputs (instructions) from the management processes. (Management processes are discussed below. In this section they will simply be referred to as "controlling processes".) The ED will also have access to sensor information of the claw in order to detect failure or success of primitive instructions. On the basis of the instructions it receives the ED will cause claw actions and camera translation movement. The controlling processes will sequentially give out instructions to the ED. Sequences of instructions can be thought of as "plans" at the level of the processes, though the ED will only know about single instructions. Thus control of the ED is useful to achieve their goals and direct behaviour.

The ED will be made of two channels. A channel will contain an input port, a processor, and an effector. One channel will be dedicated to the claw, the other to the camera. This division will allow claw and camera actions to execute in parallel.

The core information of instructions will have the following form:

instructionName(Argument1, ..., Argument N)

The arguments will be data-structures or pointers to them. There will also be a port number and an identification tag for the instruction. The port number will be used to determine whether the instruction is for the camera or the claw; the identification number will be used in records of success or failure of instructions.

Here follow the instructions that will be available and their specification. Each specification has two parts: a description of the action (if successful) and preconditions. If the pre-conditions of an instruction are violated then the action will fail, and the ED will store an error message with the identification tag in the World Model, which will be accessible to the process that initiated the instruction. This information could be used by controlling processes for error recovery.

- **pickUp(Baby)**. Pre-conditions: (1) Baby is immediately adjacent to the claw; (2) the claw is empty. Action: This will cause the claw to pick up Baby.

- **deposit(Baby)**. Pre-conditions: (1) the claw is holding Baby; (2) there is an unoccupied position that is immediately adjacent to the claw. Action: This will deposit Baby in an adjacent unoccupied position.

- **moveTo(Position)**. Pre-condition: (1) claw is immediately adjacent to Position. Action: This will cause the claw to move to Position.

- **enter().** Pre-conditions: (1) the claw is immediately adjacent to a curtain; (2) the position immediately in front of the curtain in the adjacent room is unoccupied. Action: This will cause the claw to pass through the curtain and thereby to enter the adjacent room. (A curtain connects exactly two rooms. See Figure 1.1.)

- **plug(Baby)**. Pre-conditions: (1) Baby must be adjacent or on the recharge point; (2) the claw must be adjacent or beside the recharge point. Action: This will cause the claw to plug Baby into the recharge outlet. The claw will still be left holding the baby afterward.

- **dismiss(Baby)**. Pre-conditions: (1) The claw must be holding Baby; (2) the claw must be adjacent to or on the dismissal point. Action: This will cause the baby to be removed from the nursery.

- **moveCamera(Room)**. Pre-condition: The camera is in a room that is adjacent to Room. Action: This will cause the camera to move to Room and thereby direct its gaze at it.

At any one time a channel of the ED will either be executing an instruction or not. While executing an instruction, it will be uninterruptable. (The actions are sufficiently brief that this does not imply that there will be long periods of not being interruptable.)

It will be up to the processes that control the ED to make sure that primitive actions are combined in such a way as to direct the effectors coherently and recover from whatever failures might arise. For example, the controlling process might test for whether an action, such as **pickUp(babyA)**, was successful and if it was not to decide what to do next on the basis of the error message. For example, if the error is that the claw is not adjacent to the baby then the controlling process might (re-) establish the goal to become adjacent to baby. Examples of "plans" (actually management procedures) that will be used to drive the effectors via the ED are given below.

5.4 Goals and Goal Generactivators

There will be two kinds of goal generactivators. The first kind are management procedures (abbreviated as "m-procedures"). They will be goal generators in as much as they will be able to expand a solution to a problem, and thereby produce a collection of goals. These goals will typically be means of achieving other explicit goals. (An explicit goal is a goal for which there corresponds an extant goal data-structure.) The second kind are programs running asynchronously to the management programs, which will respond to their activation conditions by producing or activating goals. (These can be thought of as reflex mechanisms based on perception of internal or external states and events.) When a goal generactivator will produce goals, it will set their descriptor fields, and their insistence. If there already exists a goal whose descriptor corresponds to the one that it would produce, then, rather than produce a new goal, the generactivators will "activate" the extant goal, i.e., they will make it a filtering candidate (hence the state of that goal will no longer be "off"). This is because, in NML1, goals will be unique and they will be identified by their descriptors (see this section, below). Table 5.1 contains the main domain top-level goals that NML1 will be able to produce, and the factors that will be used to compute their insistence. In NML1, a goal, **G1**, is considered as a top-level goal if there does not exist another goal (or set of goals) **G2**, such that **G1** is strictly a subgoal of **G2**.

Table 5.1

NML1's goals, and their insistence functions

Descriptor	Insistence
!(not(closeToDitch(Baby))	A function of the distance between the baby and the di
!(not(lowCharge(Baby)))	An inverse function of the charge
!(not(thug(Baby)))	A function of the number of babies in the room
!(not(inNursery(Baby)))[1]	A function of the population of the nursery
!(not (inNursery (Baby)))[2]	A function of the population of the room and the time during which this problem has been present.
!(inInfirmary(Baby)))[3]	A function of the number of injuries that the baby has
!(not(overpopulated(Room)))[4]	A function of the difference between the population of room and the threshold number of babies in the room

[1] This goal can occur for different reasons. In this case the rationale is that **age(Baby) >ageThreshold**.
[2] Rationale for this goal is that **dead(Baby)**.
[3] Rationale for this goal is that **injured(Baby)**.
[4] The term Room unifies with an integer representing the room that is overpopulated

The specification of NML1 goals differs from the one provided in Ch. 3—as a simplification, information about goal intensity is not computed. This is because it is not yet clear precisely how to determine intensity, nor how to use the measure in conjunction with other dimensions of goals. In other respects, the Ch. 3 requirements hold for the nursemaid.

It was said above that asynchronous goal generators whose conditions of activation are met will verify whether the goal that they would generate is present in the system, and if it is then rather than generate a new goal they will activate the existing one. This will prevent the system from generating different versions of the "same" goal. The need for such a mechanism was discovered when an early version of this architecture was implemented, and it was found that the "same" environmental contingency (e.g., seeing a baby that is close to a ditch) repeatedly triggered the construction of similar goal data structures. Comparison will be made with all goals, in parallel. Two goals will be considered as identical if they have the same descriptor[1]. Descriptors will be expressed in a rapidly obtainable canonical form to facilitate identity comparison.

Since goals will be individuated by their descriptors, the level of detail that exists in the descriptor will be quite important. For instance, if the descriptor merely states that there is "a baby close to a ditch", then this will express less information than is available if it states that "babyB is close to a ditch", and therefore more dispositions will be considered equivalent to it. The human mind allows progressive refinement of the descriptor of goals, whereas NML1 will not.

Goal generactivators must have access to parameters for determining when to generactivate what goal. These data will be contained within the generactivators. The main data will be: the dismissal age for babies, the critical charge below which NML1 should consider recharging a baby, the maximum number of babies in a room (above which babies start turning into thugs), and the maximum safe distance to a ditch. As an example of all of this, note that a certain goal generator will respond to the fact that a baby is older than the dismissal age by generating the goal to dismiss the baby.

Many other goal generators will be required. For instance, after a goal has been scheduled for execution the system might detect that it is less urgent than previously thought. This would cause a goal to be generated which has as its objective to reschedule the goal. If a dependency maintenance scheme were implemented for all types of decisions, the system could set up monitors which detect when the reason for a decision is invalidated, and that would create a goal to reassess the decision. A few other types of goal generators are mentioned below.

---

[1]It is debatable whether the goal descriptor is a sufficient basis for identity. One might argue that the rationale field ought also be included: thus, two goals with the same descriptor but different rationales would be embodied in different data structures. Philosophical aspects of the identity of motivational constructs are discussed by Trigg (1970, section V).

If a new goal does not get past the filtering phase, it will be stored in the New Pre-Management Goals database, and removed from the system when its insistence is 0. Insistence of goals in this database will decay steadily if not activated. However if the filter threshold falls faster than the insistence, then the goal may be able to surface. If a goal does get through the filtering phase, and if it is new, it will be put on the bottom of a new goal stack in the Goal Database (described below). It will be removed from there only if it is satisfied or otherwise deemed to be "inapplicable" (these are judgements that can only be made by management processes).

5.5 Insistence assignment

Insistence assignment will be performed on a cyclical basis. Insistence heuristics were abstractly described in Table 5.1. NML1 will need to be prepared for the possibility that more than one generactivator generates the same goal at any one moment. Then how should insistence be computed? There are many alternatives. For experimental purposes, it was decided that generactivators should contribute a suggestion for a goal's numeric insistence, and that more than one generactivator could contribute such a suggestion. If a goal only has one insistence suggestion, then that will determine the insistence; if a goal has more than one suggestion, its new insistence will be at least equal to the maximum suggestion, while the other suggestions will be factored into the equation; if it has no suggestion, then its insistence will be decreased by the product of its previous insistence and the insistence decay rate. Usually, there will only be one source of insistence per goal.

In the current state of the specification, the user of the model will have to tweak the insistence assignment functions so that they yield "sensible" values, based on an arbitrary set of utilities. A less arbitrary set of assignments could result from a learning process or an evolutionary mechanism, both beyond the scope of this research.

5.6 Goal Filter

NML1 will use an explicit filtering mechanism, which will take a collection of goals as input, and allow at most one of them to surface at a time. It is designed according to a winner-take-all mechanism which will allow for the possibility that no goal wins (surfaces). A variable numeric threshold will be set for filtering goals. Most filtering candidates will be subject to this threshold; but certain specific goals will have their own threshold.

The Filter will have three independently variable components: (1) a global threshold (i.e.. a threshold that will apply to most goals), (2) idiosyncratic thresholds (3) and a management efficacy parameter. The global threshold will be a real number between 0 and 1. The "idiosyncratic thresholds" will be a collection of two item collections which will contain (a) a pattern that can be unified with a goal descriptor, and (b) a real number between 0 and 1 representing a filter threshold.

The management efficacy parameter will weight the management's ability to set idiosyncratic thresholds.

Filtering will be performed according to the following three stage algorithm. Firstly for all goals that are "filtering candidates" the filter threshold will be found in parallel. If the descriptor of a candidate goal does not unify with a pattern in the idiosyncratic threshold ratios, then the global threshold will be used for it. Otherwise, the pattern's associate will be used as its threshold. Thirdly, if (and only if) there are supraliminal goals (resulting from the first and second stages), then the most insistent one will be allowed to penetrate the Filter, though a stochastic function will be used in order to prevent highly insistent goal from continuously overshadowing others (an inhibitory mechanism could also have been used that would inhibit the more insistent goals). In order to promote stability, multiple goal surfacing will not be permitted.

In NML1 only two parameters will drive the global filter threshold. This makes it different from the specification of the previous chapter. In particular, in this domain there is no need for "acute management protection". The parameters are interval busyness measures and refractory periods. Busyness measures will be computed by management processes. Interval busyness measures are rough estimates of the importance of the effects of the management process remaining idle for a certain time. The length of the period that will be used in this context is an estimate of the time it would take for a meta-management process to detect that a goal is not worth managing currently and postpone it. The user of the model will need to determine on an a priori basis the particulars of the function that takes busyness as an input parameter and returns a threshold value. This needs to be done on the basis of knowledge of the utility that corresponds to given insistence measures. For instance, if an insistence measure of 5 can be generated when the effect of non-surfacing is that a baby dies, then (ideally) the filter threshold should only be above 5 if the effect of interruption is worse than a baby dying (e.g., if it causes two babies to die). Recent work on decision theory (Haddawy & Hanks, 1990; Haddawy & Hanks, 1992; Haddawy & Hanks, 1993; Russell & Zilberstein, 1991) might be relevant for determining expedient numeric filter thresholds.

Management processes will be able to determine idiosyncratic filter thresholds indirectly. This will be a method for meta-management processes to implement decisions to postpone the consideration of goals by selectively increasing or decreasing the likelihood that a goal surfaces. This will be achieved as follows. A management process will inform the Filter that it would like to add an item (i.e., a pattern and a value) to the idiosyncratic filter thresholds. The Filter will accept any such request; however, it will weight the value by multiplying it by the management efficacy parameter. This parameter will allow the system conveniently to control the extent to which an m-procedure can control the Filter (and thereby control its own processing). If the parameter is zero, then the management process cannot directly increase or decrease its sensitivity to particular goals. The need for parameterised management filter control was discussed in Ch. 4. Idiosyncratic filter thresholds will persist for a fixed number of cycles, and then will be deleted automatically.

The state of activation of a goal that penetrates the Filter will be set to "asynchronously surfacing". If the goal does not figure in a goal stack then a new goal stack will be created; on top of this (empty) goal stack a new meta-goal[1] will be pushed. (Goal stacks are described below.) The objective of this meta-goal will be to "manage" the surfacing goal. If a goal stack does exist, and its associated m-process is suspended, then its m-process will be activated.

5.7 M-procedures and associated records

Four kinds of data that are relevant to m-processing are described in this section. (1) M-procedures (m-procedures) are structures that will discharge the management functions described in Ch. 4. As described in a following section on the Interpreter, m-procedures that are "applicable" to a surfaced goal can be selected by the Interpreter. (2) Procedure activation records are temporary records formed as a substrate for the execution of m-procedures, in response to the surfacing of goals. (They are analogous to call stack frames in procedural programming languages.) (3) Process records will contain procedure activation records (they are analogous to Process records in Pop-11). (4) S-procedures are implementation level procedures. These four types of data structures are described in turn.

M-procedures will contain information used to determine whether they ought to be executed, and to construct procedure activation records for themselves if necessary. They will have the following fields.

- Applicability detector. Normally m-procedures will be applicable to a goal if the goal's descriptor matches the procedure's goal descriptor, and some conditions that are specific to that procedure are met. However, unlike in PRS, the user of the model will have the liberty to allow a procedure to be applicable to a goal even if it is not meant to satisfy it. The applicability detector, will tell the Interpreter whether or not its m-procedure is applicable to a goal. When the applicability detector for a particular m-procedure (described below) will execute, it will "know" about the context in which it is operating (through links with the World Model). It will therefore be convenient to allow the applicability detector to be responsible for setting the input parameters of the procedure activation record as well as other individuating information (these parameters are described later in this section).

- Goal pattern (intended outcome). This is the outcome which the m-procedure aims to achieve. This can be unified with the descriptor of a goal on a goal stack. The goal will often be the

[1]This is termed a "meta" goal because the argument of the predicate of its descriptor is a goal.

achievement of a management result, such as deciding whether to adopt a goal, or when to execute it, etc.

- Body. The body will contain the instructions that will be executed when an m-procedure is run. These instructions may cause goals to surface (and thereby trigger more m-procedures), they may read and manipulate information throughout the system, and they may send commands to the ED.

- Outcome predictor. This field will be reserved for "expansion" procedures that direct physical action. It will contain a special purpose procedure that returns a collection of collections of descriptors of possible consequences of the m-procedure. Some of these consequences will actually represent failures of the m-procedure. This field will be used by other m-procedures which must decide which m-procedure to use to attain a goal. General purpose predictive m-procedures are described below, as are the difficulties of prediction. (NML1 will have to deal with variants of the frame problem.)

- Activation revision procedure. Procedure activation records will have an activation value (see below). Each m-procedure will know how to compute the activation of its activation record. This activation value will be used by the Interpreter to prioritise multiple procedure activation records that are applicable to the same goal. Activation procedures need to be designed to reflect the relative efficacy of the procedure.

- Interruption action. Procedures that use interruptable anytime algorithms will be able to store a procedure which when applied yields the currently best solution. (Note that this field will not contain intermediate results of computation. For example, it will not be a process stack.)

Here is an abstract example of an expansion (management) procedure that is meant to satisfy the goal to recharge a baby (as in the scenario described above). Its applicability detector will respond to any situation in which its goal pattern matches a surfaced goal, and where the goal is scheduled for current execution. Its goal pattern will have the following form:

!(recharged(Baby)))

where **Baby** is an identifier that will be unified with a data structure containing information about a baby, as described above. The body of the m-procedure could be defined in terms of the following procedure, which uses the PRS goal expression syntax described in Ch. 2, within a Pop-11 context (Anderson, 1989).

Procedure 5.1

```
define recharge1(baby);
   ! position(baby) = rechargePoint /*rechargePoint is a global variable*/
   ! plug(baby);
   # hold(baby) and ! recharged(baby)
enddefine;
```

As in PRS, expressions preceded by an exclamation mark (**!**) denote goals to be achieved, and the pound symbol (**#**) denotes a goal of maintenance. Either symbol will cause a goal structure to be created and pushed onto the goal stack of the process record in which the procedure activation record is embodied. This particular m-procedure body will assert the goal to move the baby to the recharge point. Then it will assert the goal to plug the baby into the battery charger. It will then assert a goal to hold the baby until it is recharged.

In order for an m-procedure to be executed, the Interpreter must create a procedure activation record for it. Procedure activation records are temporary activations of a procedure. It will be possible for there to be many concurrently active procedure activation records for the same m-procedure. The following information will be associated with procedure activation records.

- An m-procedure, with all its fields (expounded above). In particular, the body of the procedure will be used to drive execution.
- Input parameters. These are data on which the process will operate, (-baby- in the example in Procedure 5.1) and which will be provided by the applicability detection procedure.
- An activation value. This will be the strength of the procedure activation record. It will be determined by the activation revision procedure contained in the m-procedure. It will be used to prioritise m-procedures when more than one m-procedure applies to a goal.
- A program counter indicating what to execute next.
- Focal goal. This will be the goal that triggered the m-procedure. (Unlike the goal information in the procedure, this field can contain literals.)
- Contextual goals. These will be the goals in relation to which the focal goal is being examined.

Procedure activation records will either be stored within an invocation stack of a process record, or within a temporary collection of candidate records from which the Interpreter will choose one to be applied to a goal.

There is a need for process records. These structures will contain the following information.

- An invocation stack, which will be a stack of procedure activation records.
- A pointer to the goal stack on which the process record's procedure activation records will put their goals.
- Dynamic state information, indicating whether the process is shallowly suspended or not, deeply suspended or not, and live or dead. A process (**P**) is shallowly suspended if it is suspended by the Interpreter while the Interpreter is doing its book-keeping; **P** can be deeply suspended by m-processes—for instance, if an m-process (**M**) determines that two processes are interfering with each other, **M** might suspend one of them. A process is dead if it has completed its last instruction or has been killed by some other process. Dead processes will be removed from the collection of process records.

It is expected that future versions of NML will have richer process records, possibly including "strengths of activation" measures, which will be used to resolve conflicts between processes (this will be analogous to contention scheduling in (Norman & Shallice, 1986)). (Cf. next chapter.)

S-procedures are procedures that will be invoked in the same way as procedures in the implementation language (e.g., Pop-11, Smalltalk, Pascal), i.e., they will be "called", and will not use the dispatching mechanism. (Dispatching is a function of the Interpreter and is described below.) Of course, there are important differences between implementation languages in how they handle procedure application. In principle they could make use of connectionist networks. But the important thing about s-procedures is that the Interpreter's dispatching mechanism (described below) is not used. That is, they will allow NML1 to perform actions that by-pass its regular dispatching mechanism (see the section on the Interpreter, below). This will allow primitive actions to be taken which can make use of the ED. This might later prove useful for implementing cognitive reflexes.

Although s-procedures cannot be invoked by the Interpreter, it will be possible for s-procedures to be called within the body of m-procedures or actually be the body of m-procedures. One needs to know, however, if a particular s-procedure does make calls to m-procedures, in which case it can only be applied within the scope of an m-procedure (otherwise goal assertions will fail).

5.8. Databases of procedures

The architecture will contain separate databases of m-procedures, s-procedures, and process records. There will be an m-procedure database and an s-procedure database. Procedure activation records will be stored within process records.

Some of the algorithms for m-procedures used by NML1 are described in section 5.12.

5.9 The Goal Database

The Goal Database (GD) will contain instances of goals—as opposed to goal classes. (Goal classes will be implicitly within goal generators and management procedures.) Decisions and other information concerning goals will be recorded in the GD. Some of the information about goals will be stored in temporary data-structures that are not mentioned here. In particular, information about the importance and urgency of goals will be implicit in procedures that do scheduling and arbitration amongst goals. Nevertheless, it will be useful to have the following separate stores of information about goals, within the GD. Decisions recorded in the database will be taken by management processes. The information will be read by management processes, the Interpreter, and some goal generators.

- New Pre-Management Goals. When goals are first generated, before they go through the filtering phase they will be put in this database, and will be removed whenever they surface or their insistence reaches zero, whatever happens first.

- Goal Stacks. These are structures which contain dynamic information for the execution of m-processes (See section 2.2.3). A goal that surfaces asynchronously for the first time will be moved from the Pre-Management Goal Database, to the bottom of a goal stack. The goal stacks will contain stacks of goals. On any stack, if goal **B** is above goal **A**, then goal **B** is will be considered to be a means of achieving **A** (i.e., a subgoal of **A**). There is no specific limit to length or number of goal stacks. Goals stacks will also be components of process records.

- Descriptor-Goal index. This will be used for mapping descriptors to goals. (Goal descriptors are described in section 3.2). Every goal in the system will have an entry there. Before a new goal is produced, the system will check this index to make sure that there is no other goal with the same descriptor. If there is, that goal will be used or activated, rather than allowing two goals to have the same descriptor. (Notice that this will not preclude the possibility of ambivalence, which can occur if the importance field stores both positive and negative information about the goal.)

- Overlapping Goals. The system will attempt to discover which (if any) of its goals have overlapping plans. These are opportunities to "kill two birds with one stone". M. Pollack (1992) refers to the satisfaction of overlapping goals as "overloading intentions". This information is stored in the Overlapping Goals database. This information can be used in determining whether a goal should be adopted or not (overloaded goals might be favoured over other ones). Note that according to the ordinary sense of "opportunity", not all opportunities are best described as overlapping goals: the opportunity might involve a new goal, such as when a motorist sees a flower shop and decides to purchase roses for his partner.

- Goal Conflicts. The system also will record goals that it concludes to be incompatible. This may trigger an m-process to resolve the conflict (e.g., by selecting between the incompatible goals).

Representing goal relations (e.g., conflicts and opportunities) and reasoning about them raises difficult unsolved questions, such as "How can we discover and represent which particular part of one or more plans interfere with one another?", and "When are utility measures of incompatibility useful and when are they not?" (Compare Hertzberg & Horz, 1989; Lesser, et al., 1989; Peterson, 1989; Pryor & Collins, 1992b; Sussman, 1975; Wilensky, 1983).

Although there will be no separate database equal to a two way process-purpose index (Sloman, 1978 Ch. 6), which maps goals to processes, and processes to goals, this indexing information will be accessible to the system. The reasons for actions will be recorded in the goal field of procedure activations. And goals themselves will have a plan field containing information about procedures for satisfying them, along with the status of execution of the procedures (e.g., if the procedure will have been activated, a pointer to the procedure activation will be available.)

The schedule will be a multifaceted part of the Goal Database. It will contain the different types of decisions that can be taken regarding when certain goals should be executed.

- The sequential schedule. This will contain a list of goals which will be executed one after another, with no other goals executed in between. Hence, this will contain an expression of the form: (**Goal1 Goal2 ... GoalN**), where **GoalN** is to be executed immediately after **Goal(N-1)**. Unless explicit information within the goal itself indicates that the goal is suspended, it will be assumed that the first goal in this list is always executable (i.e., an execution m-procedure might be activated to get it going). If such a goal is suspended, its activation conditions must be recorded somewhere else in the schedule, otherwise its entry might be removed from this part of the schedule. The reason for this is to allow the rapid processing of goals in this part of the schedule.

- Ordered pairs. This will be a list of pairs denoting a partial ordering of goals that is used to control the sequential schedule. A partial order is a useful intermediate state when enough information for a total ordering is not yet available. These will be expressions of the form (**Expression1 Expression2**), where at least one of the expressions is a Goal. If **Expression1** is a goal, then that means that it should be executed before **Expression2** is executed or occurs. For instance, the pair **((isEmpty(infirmary)) (!heal(babyC))** means that the goal to heal babyC should be executed after the infirmary is empty. More typically, both expressions will be goals. The ordering is partial, in that goals can be executed between any of the items within a pair. Constate that the sequential schedule denotes stronger ordering relations than this (nevertheless, the sequential schedule is open to revision). There will be goal generators that can detect inconsistencies between the ordered pairs schedule and the sequential schedule and trigger a goal to resolve the inconsistency. However these goal generators will not always be active and therefore might not detect every inconsistency that arises. In general detecting inconsistency can require arbitrarily long computation times.

- General conditions. This will be a list of schedule items that have the form, **(Condition-Goal)**. This will be useful useful when the system knows that an action should be taken contingently upon a condition, but when it does not know the sequence of events that will precede the condition being true. **Condition** is an arbitrary expression that evaluates to true or false. It could be, for instance, that a particular amount of time has elapsed, or that a room is now available, etc. **Goal** is a regular goal expression. This could be an execution goal, or a meta-management objective (e.g., to schedule a goal or to resolve a conflict). Here are examples of general conditions schedule items:

[[unoccupied(recharge-point)] [!(recharge(babyA))]]

Thus, when the recharge-point is unoccupied, the system should execute the goal to recharge babyA. If this goal is already on a goal stack, it will be activated. Otherwise, it will be pushed onto a new goal stack.

[[isolatable(babyA)] [!(isolate(babyA))]

Here, the goal to isolate babyA will be executed whenever it is deemed possible (e.g., when there is an empty room, or when the population density is below a threshold). This goal might be a subgoal of a goal to handle a thug —i.e., **!(not(thug(baby2))).**

A goal activator, which is a schedule monitor, will verify whether the condition of a schedule item is met, and if it is it will activate the goal expression. These goals will have to go through a filtering process like any other asynchronously activated goal.

- Pending goals. These will be goals about which the system has not taken a specific decision, except perhaps to deal with it at some undetermined future point. They need to be reactivated when free time is available.

The reason that the system is designed to represent many types of decision is to reflect the variety of scheduling decisions that humans can take. However, having such disparate scheduling information poses consistency and dependency maintenance problems. Some of them are discussed below. There are many such unresolved problems in other parts of the design as well. It is tempting to require that any information only be added to the schedule if it does not produce an inconsistency. However, this requirement is incompatible with the constraints of autonomous agents (particularly limited knowledge and limited time in which to make decisions); moreover, humans tolerate unrecognised inconsistency while usually dealing with inconsistency when it is detected.

5.10 Epistemic procedures and processes

There will be a collection of epistemic procedures. These will be s-procedures that run independently from the management processes, and perform updates to the system's knowledge, such as setting flags to fields. Currently, these processes will only be used for determining whether a goal is satisfied or not. (See the section on the Interpreter below, particularly the text concerning its maintenance procedure.) These functions will not be discharged by the m-processes, in order not to interfere with m-processes or the Interpreter. These processes will not require "limited resources" like the Interpreter, the claw, and the camera. These processes can be seen as discharging some of the functions which A. Sloman (1978 Ch. 6) attributed to special purpose monitors and general purpose monitors. Functions that are analogous to Sloman's monitors will be distributed throughout NML1: in the asynchronous goal generators, the m-procedures, the perceptual module, and the epistemic processes. Future research might suggest the need to reorganise and expand monitoring functions in the nursemaid.

The epistemic procedures will reside in their own database. Associated with each epistemic procedure is an activation condition, a disactivation condition, and activation mechanism. Thus they can be conceived as demons.

5.11 The Interpreter

In this section, the Interpreter is described, and in the next an example of the system in action is given. The role of the Interpreter is to choose and run management processes in response to goals that surface. It should execute quickly. It should be able run management processes in parallel. It itself is not meant to engage in any of the management functions. The Interpreter will execute in a cyclical fashion. It could be defined in terms of Procedure 5.2.

```
Procedure 5.2
define interpreter(nursemaid);
   lvars   parsGoals;
   repeat forever;
      applicableProcedures(managementProcedures(nursemaid)) -> parsGoals;
      selectPars(nursemaid, parsGoals) ;
      runPars(processRecords(nursemaid));
      maintenance(nursemaid);
   endrepeat;
enddefine;
```

The Interpreter will select processes to run as follows. To begin, it will use the s-procedure **applicableProcedures** which instructs each one of the m-procedures (stored in the m-procedure library) in parallel to return a list of all of the surfaced goals to which they apply. The database of m-procedures will be given as input parameter. The output parameter **parsGoals** will be a list of goals and the procedure activation records which apply to them. (This could be a list of the form:

[[goal_1 [$\text{par}_{1,1}$ ... $\text{par}_{1,N}$]]

[goal_2 [$\text{par}_{2,\,1}$ ... $\text{par}_{2,N}$]]].

where $\text{par}_{1,1}$ to $\text{par}_{1,N}$ are procedure activation records that apply to goal_1). Normally, an m-procedure will apply to a goal only if it is designed to satisfy it. However, it will be possible for an m-procedure to be triggered by a goal which it is not even designed to satisfy. For instance, the combination of a certain goal plus some belief might trigger a procedure to verify whether the goal should be adopted. For each procedure that is applicable to one or more goals, **applicableProcedures** will construct a procedure activation record. When a procedure activation record is constructed, its activation strength will automatically be computed.

For every collection of procedure activation records that apply to a goal **selectPars** will choose one to be executed. (In order for more than one procedure simultaneously to work for the same goal—i.e., to achieve "threaded processing"—one of the procedures must create a new goal stack which will trigger the other procedure. Threaded processing has not yet been carefully studied by the author.) The selection will be made on the basis of the strength of activation of the candidate procedure activation records. By an overrideable default, there will be the constraint that previously tried procedures cannot be reapplied to identically the same goal. **selectPars** therefore will record which procedure has been selected for which goal. If the goal does not have a process stack, then one will be created. The procedure activation record will be pushed onto the appropriate process's procedure invocation stack.

runPars will execute in parallel all of processes for which a procedure activation record has been selected. Each m-process is assumed to execute by itself (as if a separate processor were allocated to it). Each m-process will be responsible for pushing goals onto its own goal stack. When it does, it will set its status to needing-dispatching, and shallowly suspends itself. (Dispatching denotes applicability detection and procedure selection.) M-processes can also perform actions that by-pass the Interpreter's m-procedure invocation mechanism: by making direct calls to procedures (in a similar manner to traditional procedural programming systems). These procedures will themselves, however, be able to assert goals and thereby trigger m-procedures through the Interpreter. Some of the actions of m-procedures will be to examine or modify databases (e.g., the World Model or the Goal Database), to perform inferences, and to give commands to the Effector Driver.

As m-processes are running in parallel, **runPars** will continuously test to see whether it should exit and proceed to dispatching. By default, the exiting condition is simply that **N** processes are needing-dispatching (where **N** by default equals 1); however, the exiting condition is redefinable. Those processes which have not suspended themselves by the time **runPars** exits will keep processing, and will not be terminated by the Interpreter, though they can be suspended by other processes; at a future time point procedures may be dispatched for whatever goals these processes might subsequently push onto their goal stacks.

The **maintenance** s-procedure will remove satisfied goals from the tops[1] of the goal stacks, and remove the awaiting-dispatching status from the corresponding processes. If no m-procedure is successful at satisfying a goal, the goal will be said to fail, and the goal thereafter will be ignored. (In further developments of the model goal failure could trigger a new goal to deal with the situation—PRS can do something along these lines—or failed goals could be periodically retried.) The maintenance procedure requires that there be a procedure which can easily and dichotomously tell whether a goal is satisfied. This raises an interesting problem. On the one hand, since the Interpreter will not be able to finish its cycle until the maintenance procedure terminates, it is important that this s-procedure execute quickly. On the other hand, it is not always easy to determine whether a goal has been satisfied—for instance, verifying whether the goal to isolate a thug has been satisfied might require the use of the camera to make sure that there are no babies in the room. This suggests that determining whether a goal is satisfied might sometimes (but not always) best be achieved by producing a goal to do so; this goal could then trigger an m-procedure. However, using an m-procedure for this purpose will sometimes be a bit of a sledge-hammer, if the information is readily available.

---

[1]This s-procedure does not check whether goals that are not on the very top of a goal stack are satisfied.

The solution that is used for this problem is as follows. For every procedure on the top of a goal stack an epistemic process (call it **P**) will be generated that tracks whether a goal is satisfied or not. (Future research should posit mechanisms for limiting the number of goals which **P** verifies.) This has the advantage that if the information for making this determination is already implicit in the system, the information can be gathered and will be guaranteed not to interfere with m-processes. Moreover, if the system always had to generate a goal to determine whether a goal is satisfied, as described below, it would be in for an infinite regress. **P** will associate two kinds of information with the goal that it monitors. The first indicates whether the goal is satisfied or not. The second indicates whether the belief about whether the goal is satisfied or not is up-to-date. An example of a reason why the information might be known not to be up-to-date is if some information that was required for the judgement was not available perhaps because the camera has not been directed lately to a certain room. (Recall that epistemic processes will not be allowed to use the camera, or to generate a goal to that effect.) When the Interpreter has to find out whether a goal (call it **G**) is satisfied, it will check first whether it knows whether **G** is satisfied. It will be able to do this by directly examining the up-to-date flag associated (by **P**) with the goal. If the up-to-date flag is false, then the Interpreter will create a goal, **G2**, to determine whether G is satisfied, and it will push **G2** on top of **G**. This will trigger an m-process whose objective is to find out whether **G** is satisfied.

There will be book-keeping chores that are too mundane to describe here. In the following chapter variants of the Interpreter are considered, including some in which parallelism is increased.

5.12 Algorithms for m-procedures

So far, NML1 has been described in fairly abstract terms at the architectural level. This architecture will support a wide variety of very different algorithms at the management level. The aim of this section is to show how management algorithms could be expressed within the constraints of the architecture and using information about goals discussed in Chapters 3-4 (e.g., regarding urgency). As is discussed at the end of this chapter and in the next one, more research is required to provide principles for designing management algorithms. The present section should be read in the light of this fact. No claim is made that the algorithms fully meet the requirements of autonomous management. Code for the algorithms is not presented here.[1]

Given the simplicity of mapping goals to plans, NML1's behaviour will mostly be controlled by its scheduling m-processes as opposed to its planning processes. (It is important to keep in mind that "scheduling" is not totally separate from the other management functions, especially since when trying to schedule a goal the nursemaid might knowingly postpone it beyond its expected terminal

[1]The algorithms were implemented in part in an earlier version of the nursemaid, which did not use PRS procedures (i.e., that use goal invocation) but regular procedures (that are invoked by name).

urgency, which is normally tantamount to rejecting it, while allowing the possibility of a reprieve if the circumstances change. In other words, the deciding function of management can take place within scheduling processes.) NML1 will have a collection of scheduling algorithms that mainly use the sequential schedule, but also use the other schedule databases. Recall that goals in the sequential schedule are supposed to be executed one after the other; however, NML1 will be able to alter its sequential schedule, e.g., truncating it or inserting a goal; moreover, goals in the general condition schedule can take precedence over goals in the sequential schedule if their conditions of execution are met. There is a main scheduling algorithm and a collection of other algorithms.

The main algorithm can be described as follows:

1. Suggest a collection of sequential schedules.

2. Project the consequences of each schedule.

3. Select one of these schedules, either on the basis of utility measures (and if so then compute the utility measures) or on some other basis.

Step 1 usually will involve suggesting a number of possible indices at which a focal goal can be executed. For example, if the schedule contains 5 goals the algorithm might consider executing the focal goal in the first, second, and fourth position. Usually, it would be too time consuming to consider all possible ordering of goals in a schedule; therefore, the algorithm which proposes an index for a goal usually will not consider re-ordering the other elements in the schedule (unless it detects a possibly useful re-ordering). A further constraint is that the sequential schedule cannot exceed a certain length. This is useful both in limiting the amount of search that is performed, and because in the nursery it is difficult to predict far ahead in the future. The nursemaid may also compare the prospects of executing a goal some time after the last goal in the schedule with executing it somewhere within the list of goals that will be executed one after another.

In Step 2, in order to predict the effects of a sequential schedule, NML1 will use a generic prediction procedure, **PP**. **PP** will take as input parameter a list of goals and will return a "projected world model" (**PWM**). (There may be several different PWMs in use by different m-procedures.) A **PWM** will comprise different slots that are temporally indexed from time 0 (present) onward. Each successive slot will contain a sub-model of the world that is supposed to hold after the application of a certain expansion m-procedure. Each sub-model will contain the usual information of a world model (as described above), a list of "valenced descriptors", and a time point at which the model is supposed to hold. The valenced descriptors will represent events or enduring states of importance that are expected to happen while a procedure is being applied (e.g., how likely it is that a baby will have fallen into a ditch; how many babies a thug might have hit by then, etc.). Appendix 1 specifies the syntax and semantics of valenced descriptors. Time in PWMs will be linear—PWMs will not support branching time. The system will simulate the effect of an m-procedure by applying an "expansion prediction procedure" (**EPP**) that will be associated with the expansion procedure that will be associated with the goal being considered. (**PP** requires that the goals have expansion m-procedures associated with them; in other words it will not be able to predict the effect of a goal without knowing exactly which procedure will be used to satisfy the goal.) **PP** will operate according to the following three stage algorithm:

1. Let the first item in the **PWM** be a copy of the present World Model.

2. For each goal in the list of goals use its **EPP** to predict the subsequent state of the world.

3. Put this prediction in the next slot of the **PWM**.

Each **EPP** will take as input parameter a goal specification, a **PWM**, and an index, **i**, indicating the previous slot of the **PWM**, which will represent the state of the world as it will be supposed to be before the **EPP** will be applied (i.e., after the previous goal is executed). Its effect will be to add a sub-model to **PWM** at index **(i+1)**. The EPPs' predictions will be very limited in scope. The valenced descriptors that will be returned will be based on the goal descriptors. For instance, if NML1 is currently anticipating the effect of trying to recharge a baby, and there is a chance that the baby will die before it is recharged, then NML1 might produce an estimate of the likelihood that the baby will have died as a result of its low charge—this will be one of its "valenced descriptors". Another valenced descriptor will indicate how long the baby will have "suffered" from having its charge lower than threshold (recall that having a low charge is considered as intrinsically bad). It will also give a point estimate of the baby's position and the claw's position at the end of the plan. The algorithm which makes predictions for plans for the goal to recharge a baby follows.

1. Determine the total distance that the claw will have to travel as it moves from its position at the start of the plan to the baby's position, and from there to the recharge point. Call this **distanceToTravel**.

2. Determine how much time it will take to travel this distance, assuming the claw will travel at its maximum speed. Call this **duration**.

3. Assuming that the charge decays at the normal decay rate (a global parameter) for **duration** cycles, predict the final charge of the baby as it reaches the recharge point and assign this value to the baby's **charge** field.

4. Set the position of the baby and the claw (for **PWM(i+1)**) to be that of the recharge point.

5. Produce valenced descriptors. For example, one prediction has the form

state(lowCharge(Baby, FinalCharge), Probability, Duration)

where Baby is a pointer to the baby in question, FinalCharge is a number representing Baby's final charge and Duration is a number representing the amount of time during which Baby's charge will have been under the charge threshold. (See Appendix 1 for more details).

If the baby's charge is predicted to be zero, then include a prediction of the form

event(death(Baby), Probability)

which indicates that Baby is expected to die.

6. Store the valenced descriptors in **PWM(i+1).**

A similar **EPP** will be associated with a plan that retrieves babies that are close to a ditch. (For that matter, every procedure that solves physical domain goals will have its **EPP**.) The valenced descriptor associated with this procedure will note the probability of the baby dying.

Scenario S1 (described above) can be reconstructed in terms of these algorithms. Recall that babyA had a low charge. Call the goal to recharge babyA "**Goal1**". The nursemaid did not have a previous goal scheduled for execution, so it decided to execute "**Goal1**" immediately. At this point suppose the nursemaid discovers that babyB is dangerously close to a ditch. Suppose it adopts the goal to retrieve babyB (call it "**Goal2**"). Then, let us say, it applies a sequential scheduling m-procedure which considers two possible orderings for these two goals: **Goal1-Goal2**, or **Goal2-Goal1**. Suppose this m-procedure predicts the outcome of the first ordering to be:

{
state(lowCharge(babyA, 0.25), 50, unknown);
event(death(babyB), 0.4).
}

That is, it is predicted (with an unknown probability) that babyA will have a low charge during 50 cycles (where its final charge will be 0.25) and that there is a 40 percent probability that babyB will die (if **Goal1** is executed before **Goal2**). Regarding the second order, suppose it is predicted that:

{
state(lowCharge(babyA, 0.2), 65, unknown);
event(death(babyB), 0.4).
}

In step 3 a decision will be made. In order to decide which of the two outcomes in our scenario is best, the system will select an arbitration m-procedure (using the Interpreter's dispatching algorithm). Selection amongst schedules will be made on the basis of utility measures unless more specific tests can assign an ordering. Utility judgements implicate a collection of specialised m-procedures that will compute the utility of individual descriptors in isolation. Each type of descriptor will have associated with it a procedure that computes a "utility" measure, which roughly will represent the importance of the outcome weighted by its probability of occurrence. (The notion of utility is criticised in Ch. 6.) The utilities of the descriptors for a given schedule will be summed. The schedule with the highest total utility will be chosen. For example, there will be a utility procedure, call it **Pu1**, for expressions of the form **state(lowCharge(Baby, FinalCharge), Duration, Probability)**. **Pu1** will return a numeric result that is proportional to **Duration** and inversely proportional to **FinalCharge** and **Probability**. And there will be a utility procedure, call it **Pu2**, that applies to descriptors of the form **event(death(Baby), Probability)**. **Pu2** will return a result that is proportional to (1) the importance of **Baby** (which takes into consideration its age, how healthy it is, and whether it is a thug), and (2) **Probability**.

There will be more specific management rules that apply only to certain configurations of outcomes. (As already said, if the more specific rules apply, then utility judgements are not made.) For instance, there will be an algorithm that can be used when the system has to choose between two collections of outcomes which involve a different number of deaths: the system will select the schedule whose outcomes involve fewer deaths. Another m-procedure will respond to situations in which both collections of outcomes implicate the same number of deaths, but where in one case there is a significantly greater probability of death than the other. This m-procedure would be invoked in the aforementioned scenario, and would favour the second set of outcomes, since it predicts a significantly inferior probability of a baby dying (0.1 vs. 0.4). If the probabilities of death are not significantly different, the nursemaid will favour the more valuable babies (according to the domain specification). There will be a dozen or so other rules that are designed for this domain, the specifics of which do not really matter; instead what matters is that the system should allow for decision-making that does not merely consider utility measures.

As mentioned above, the system not only will update a sequential schedule, it will also take decisions about partial orders of goals. Moreover, not all of these decisions will be based on projections (though many will be at least based on the anticipated state of the world <u>before</u> the goals are executed). There will be a scheduling m-procedure that will be applicable when it might be useful to decide to execute a goal before or after another. For instance, according to one rule, if a problem involves moving a baby to an overpopulated room, and there is a goal to depopulate the room, then the latter goal should be executed before the former. Thus a judgement to execute a goal to recharge a baby after a goal to depopulate the recharge room might be recorded in the partial order schedule. The system will prefer to deal with thugs before dealing with the babies they injure, provided there is a way to isolate the thugs. There will be a collection of rules for dealing with problems of the same type. For instance, if the system has to choose between recharging two babies, it will choose to recharge the one with the lowest charge first, unless it is already anticipated that it cannot recharge both, in which case it will prioritise the most important one. Another heuristic will be to be sensitive to opportunities. For example, if there are two goals to dismiss dead babies, start with the baby that is closest to the claw at the time when the plan will be executed. NML1 will check whether prescriptions of the partial order schedule have been violated, and if they have it will activate an m-procedure (call it **P**) to decide whether this violation is worthwhile; if it is not it will consider reversing it.[1] NML1 will record which goals **P** has processed, and it will refrain from applying **P** to the same goals. Another heuristic is to prioritise those goals for which there are plans that overlap (according to information on the subject that will be stored by m-procedures in the goal overlap database).

Finally, there will be m-procedures that fill in the general conditions schedule. They will mostly be useful for scheduling activities around "resources", i.e., the infirmary and the recharge-point. When there will be many babies requiring one of these rooms, the nursemaid might decide—say—to fix babyA when the infirmary's population is below threshold. Whereas reasoning about the sequential scheduling will be relatively straightforward, the indeterminacy of the general schedule will make it more difficult. For instance, given a sequential schedule it will be difficult (and often impossible) to predict when items of the general schedule will be executed in relation to those of the sequential schedule (e.g., it might be difficult to predict when the infirmary will be free, because babies might walk into it).

Thus there will be a wide variety of scheduling algorithms to choose from and decisions that can be taken. One of the main problems with this proposal is that processing can become somewhat baroque as different procedures are triggered to operate on the schedules. (Simulation would be useful to help evaluate the m-processing.) There will be a need to inhibit different scheduling m-procedures from operating on the same goals at the same time, or interacting negatively with each other. Requirements and design for this have yet to be proposed, although there is a growing literature on AI scheduling techniques (Beck, 1992; Decker, Garvey, Humphrey, & Lesser, 1991; Desimone & Hollidge, 1990; Donner & Jameson, 1986; Drummond, Bresina, & Kedar, 1991; Fox, Allen, & Strohm, 1981; Fox & Smith, 1984; Gomes & Beck, 1992; Haddawy & Hanks, 1990; Prosser, 1989; Slany, Stary, & Dorn, 1992).

---

[1]This is an example of NML1 being selective in its scheduling. NML1 does not consider every possible sequential scheduling order. But it may question a particular order within the sequential scheduling (say the fact that Goal1 occurs before Goal2), and possibly modify it.

The scheduling and arbitration algorithms given above will only indirectly determine what action will be produced on a given cycle. There will be an m-procedure that is more directly responsible for current action selection (i.e., selecting which goals to execute now). There might be more than one goal that is ready for execution at one time: the head of the list of goals in the sequential schedule and any number of goals in the general schedule. Each goal which is known to be ready for execution will be stored in a collection **goalsReadyForExecution**. The current action selection routine will take a decision about which of these goals, call it **G1**, to execute, and will record that all of the others are not to be executed before **G1**. In this manner, it will not continuously have to reconsider the postponed goals. Ideally, current action selection would be performed according to an anytime algorithm; in the current design, however, the system will use a fixed two stage algorithm:

1. It will collect a list of **goalsToConsider** from the goals in **goalsReadyForExecution**. **goalsToConsider** will essentially be a copy of **goalsReadyForExecution** from which will have been removed most of the goals which are not supposed to be executed before others.

2. It will apply its arbitration rules to **goalsToConsider** and if none of them yields an ordering it will choose at random. In order to prevent the system from repeatedly interrupting itself and not getting anywhere, the arbitration routine will have a bias toward the currently executing goal (if there is one) and this bias will increase with the amount of interruption.

The system needs to be able to respond to situations in which a supergoal (**S**) of a process (**P**) that is currently executing is satisfied, so that it can consider whether all of its subgoals are still necessary or whether execution can continue to the next instruction after **S** in **P**. This might prevent unnecessary processing and execution. For example, **S** might indicate that the population of a room should be set to below 5. This will usually repeatedly trigger a procedure which will select a baby and move it out of the room. While this procedure is running, the population of the room might fall below 5 (e.g., because a baby runs out of the room, and some other baby dies). When this happens, the claw (which might be in a different part of the nursery) might be in the process of heading toward a baby that it intends to move out of the previously overpopulated room. However, it would be useful at this point to abandon this goal since its reason for being is now (serendipitously) satisfied.

In NML1 this is achieved by having a goal generator that responds to such a situation by generating a goal to "respond-to-satisfied-process-supergoal". This in turn will trigger a meta-m-procedure that will suspend **P**, truncate its goal stack from **G** upward, and cause the execution pointer to be set to the instruction following **S** in **P**. This meta-m-procedure is rather unrefined, since often some recovery action would be required (such as depositing a baby the claw is carrying). However, it remains for future research to improve upon this with more general principles. There are analogous cases in more realistic domains where action is redirected and there would need to be procedures that are executed before control passes from one action to the other. For example, an administrative secretary who decides to stop working on a file before attending to his supervisor's request for assistance might need to save all of the files in his word processor.

This section gave more specific indications of how NML1 could be designed. Still, there is a need for more research to provide more elaborate guidelines on designing m-procedures. The next chapter discusses some of the problems that need to be addressed in this respect.

5.13 Conclusion

Trade-offs and possible enhancements are discussed in the following chapter. In the process of that discussion, the current design is further elucidated, since it is explained from a variety of standpoints.

Chapter 6. Critical examination of NML1

In this chapter a critical examination of the NML1 architecture and the principles behind it is undertaken. As Karl Popper has argued, scientific progress requires careful critical examination of theories, with an enthusiastic focus on their weaknesses (Popper, 1956/1983; Popper, 1959). From the design stance, it is important to assess the ways in which a design meets the requirements, and the respects in which it falls short. The requirements of NML1 are extremely challenging, and in this respect it is not surprising that it and the more general theory behind it fall short of a satisfactory explanation. Therefore, in this chapter many theoretical shortcomings and requirements are identified.

6.1 Some strengths of the contribution

Much of the justification and supposed advantages of NML1 have already been given above. In this section the boons are summarised. Subsequently, weaknesses and problems are expounded.

NML1 will benefit from many of the advantages of a procedural reasoning system (discussed in Ch. 2). The Interpreter and management processes will be interruptable and redirectable. The system will be able to change its current plans in response to new information. Planning and physical actions will occur simultaneously or in an interleaved fashion. PRS procedures can rely heavily on information gathered at run time (thus deferring a lot of decision-making). Using procedures requires less explicitly stated control knowledge than production systems. Procedural reasoning systems achieve task-level decompositions as Brooks (1986b) does; however, allowing meta-management processes permits a top level control to take place in a more reflective manner than Brooks's inhibition-excitation control links permit.

NML1 (as a design) also differs from and improves upon PRS as follows. Having asynchronous goal generators is a flexible way of making the system reactive to new problems and opportunities. In PRS new facts can trigger procedures, but they cannot directly trigger goals. A weakness of the PRS approach is that unlike other systems (e.g., STRIPS) the system must commit itself at the outset to a plan (i.e., a procedure) for responding to a situation unless more than one procedure is triggered by the situation. NML1 will have the opportunity to select from a number of procedures which apply to a goal that was triggered by a fact. Such a procedure could be one that decides whether or not even to accept the goal. Another advantage of asynchronous goal generators is that it will make the Interpreter more efficient, because it will not serially have to check for "knowledge areas" being applicable in this way. Moreover, NML1's method will allow for goal generators to process over longer periods of time (e.g., perhaps detecting complex conditions) because their processing will not compromise the reactivity of the Interpreter. One could also make a case that human motivation exhibits such coarse grained parallelism too. (Such parallelism is a feature of many psychological models.)

NML1 will have richer representations of information about goals. Only a subset of the structure corresponds to the normal concept of a goal. It will also have a structured Goal Database in which various kinds of decision can be recorded. There will be a distributed two way action-motive index (Sloman, 1978), which will map actions onto the goals that they are meant to achieve, and goals to the processes operating for them. The NML1 Goal Database will be able to represent goal relations beyond goal-subgoal relations (e.g., conflicts). In PRS, the goal database just contains stacks of goals, and a uniform database of facts is assumed to contain all information about goals and beliefs. However, principles are required to structure this database, especially since on every cycle each knowledge area must decide whether it is applicable via unification with facts in the database. The Goal Database of NML1 is an attempt at structuring knowledge about goals. Georgeff assumes that all intentional information in the whole system— goals, applicability conditions of procedures, and the database—should be represented in a uniform formalism: first order logic with temporal operators. In contrast, NML1 as a design does not restrict the form of representation that can be used for different purposes. For instance, it will be possible to add a component to NML1 which uses analogical representations to draw predictions (compare Gardin & Meltzer, 1989).

Goal filters are supposed to protect the system heuristically from untimely distractions. The utility of this has been amply discussed in Ch. 4 and publications of Aaron Sloman.

NML1's Interpreter will be capable of running procedures as it is going through its dispatching routines, whereas in PRS all knowledge areas are suspended as the Interpreter works. NML1's method is useful for m-procedures that need to adjust their behaviour in response to incoming information, and that cannot afford repeatedly to be suspended for dispatching. NML1 will still retain the ability to suspend such procedures, both synchronously and asynchronously to their operation. In order to achieve NML1's level of management asynchrony, a PRS system would need to be composed of many PRSs (which is something Georgeff explored).

Two potential advantage of PRS's interpreter over NML1's are worth mentioning. One is that it allows one to prove properties of the system, given that preconditions, applicability, and goals are expressed in the same formal language. (See Georgeff & Lansky, 1986). However, the utility of these proof rules is constrained by the fact that environments of autonomous agents preclude complete and accurate knowledge of the world—and such knowledge is required in order to prove that a behaviour will necessarily achieve its intended effect. The second is a matter of time. Georgeff and colleagues claim that their interpreter has a provable upper bound on the amount of time it takes to decide whether a procedure is applicable, given that applicability is based on direct unification of knowledge area data with database information (no inference can be performed by the interpreter.) However, this is only a boon for systems in which the interpreter waits for applicability detection to be performed. In NML1, applicability detection is assumed to take place in parallel for all procedures. Moreover, NML1's Interpreter will not wait beyond a certain amount of time to find out if processes are applicable.

The NML1 design is not an overhaul of PRS. It builds upon PRS and combines it with design principles proposed by Sloman, others, and myself. Despite the appeal of NML1 and the principles behind it, the requirements of autonomous agents have not yet been satisfied. In the following sections, possible improvements to NML1 are discussed, as are the shortcomings in the explanations of required capabilities of autonomous agents. The following discussion has important implications for how a more sophisticated procedural reasoning system could be developed in future research.

6.2 Valenced knowledge and conation

Little has been said about the World Model and its links to goal generators. It has simply been assumed that goal generators will have conditions of activation that can be based on facts in the World Model or other information in the system (e.g., information in the Goal Database). The theories behind PRS and NML1 were criticised above because they do not provide guidelines for designing databases. In particular, the global structure of the database as well as the finer structure of items within the database ought effectively to convey information about functionally relevant or valenced facts. Valenced information implies or states that some fact is contrary to, or congruent with motivators. (Examples are given below.) These facts need to be summarised appropriately, and to some extent "attract attention"; and the system needs to be able to ascertain this "extent".

NML1 will produce goals in response to information in the World Model. For instance, one goal generator will react to the position of a baby by producing a goal to move it away from a ditch; another goal generator will respond to a baby's battery charge by producing a goal to recharge it; etc. Notice, however, that the information to which these goal generators will respond is descriptive and not explicitly evaluative. The World Model will not contain motivationally "tainted" information, such as that a baby is too close to a ditch, or that its charge is too low. As argued below this is a weakness of the design. A better design would have ways of representing information in such a way as to flag significant data and thereby make them "salient". It is not yet clear how this can best be done. The simplest way is for a tag to be added to appropriate data; e.g., "important". But this will no doubt be overly simplistic in more complex systems.

The capability of perceiving and encoding states and ongoing events as good or bad is intimately related to Gibson's notion of the perception of "positive and negative affordances" (1979 Ch. 8). Gibson stresses that much of what one sees does not merely consist of actual spatio-temporal features of the world, but of affordances. "The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or for ill" (Gibson, 1979 p. 127). For instance, one can literally perceive that a chair affords support. Related empirical work demonstrates that which causal rule people select when faced with mechanistic interactions is affected by the degree to which the information required by the rule is perceptually salient (Shultz, et al., 1986). Gibson's ideas about affordances are integrated and expanded by Sloman (1989), which outlines requirements for vision. Vision can provide intricate information indicating threats of predators or vulnerabilities of prey. Humans often (but not always) do not seem to need extensive reasoning in order to detect problems and opportunities as such. Now the concept of affordance does not completely capture valenced knowledge if it merely implies a potential for good or bad; this is because one can also perceive something which is actually good or bad: compare the difference between knowing that an animal can bite you (that it "affords biting") and knowing that it will attack you or is attacking you. Rapidly perceptually detecting actual problems is important.

Detecting the motivational relevance of information might involve producing useful summaries of information about a situation. For instance, simply noting that a baby is dangerously close to a ditch might stand for a lot of information, such as that the baby is mobile, near a ditch, can potentially fall into the ditch, and that falling into a ditch is a bad thing. This is not to say that little further information will be required for dealing with the situation, but that the valenced information might be enough to trigger and direct the management.

An advantage of being able perceptually to produce valenced information distinctly from goal generation is one of modularity. It allows one to vary one process independently from the other. Hence one can change what goals are generated in response to a kind of situation while keeping constant the respects in which the situation is seen as good or bad.

One might be tempted to pursue this line of reasoning even further and say that goal generators are only useful when the best way of dealing with a type of situation is known before a particular instance of the situation is detected. Consider the case of NML1 responding to a baby being close to a ditch. It will produce the goal to move the baby away from the ditch. Here, the goal will implicitly serve as an index to procedures that it ought to trigger and that are designed to deal with the situation. One might argue that fixing a link between the situation and a goal overly restricts the system, because it effectively constrains the systems response to a situation to a class of plans (i.e., those plans that can in turn be triggered by the goal). This seems to rule out the possibility that plans that serve a different but related goal be executed. For instance, it appears that the system could not use a plan that responded to a baby being close to a ditch by blocking the path to the ditch. One might argue then that rather than producing such an inappropriate goal, the system ought simply to "note the problem" (i.e., see the situation in a valenced way as problematic or an opportunity) and trigger an m-procedure to decide how to deal with it. In contrast with goals, such valenced knowledge does not directly specify what needs to be done or achieved. It might merely indicate that it is likely that something needs to be done. (The above text was written in terms of "appearances" (e.g., "it appears that ...") because the NML1 architecture (unlike PRS) will be capable of responding to one goal by enacting a procedure that does not satisfy it. So NML1 could respond to the goal to move the baby away from the ditch by blocking the path to the ditch.) In defence of NML1, however, one response to the argument against situations directly triggering goals is that NML1's goal generator in question will just be misdesigned: perhaps it should simply produce a more abstract goal such as "prevent the baby from falling into the ditch". (The system could be made to generate an even more abstract goal: to decide what to do about the fact that the baby is too close to the ditch.) Such a goal would have indexed a wider class of procedures from which to choose. This response is valid. (Incidentally, realising that a goal is not sufficiently abstract for a "situation" is not merely important for the designer: it is also an important ontogenetic task.) None of this implies that it is useless to be able to record information about what is good or bad in a current situation before producing a goal in response to it. Indeed, noting what is good or bad about a situation is a key requirement for determining (and articulating) which goals ought to be generated.

The level of abstraction of the goal that is generated in response to a situation might need to vary according to the situation and as a function of learning. In cases where a quick response is required and where there are valid indications that the best way to respond to a baby being close to a ditch is by going to pick it up, it might be best to trigger this goal rather than its more abstract relative (to prevent the baby from falling into the ditch). Alternatively, one could stick to the current design of NML1 and just ensure that the dispatching process trigger an m-process to pick up the baby (rather than, say, to block its path).

If some of the information about the significance of events is encoded by vigilational processes (rather than management processes) then it might be useful for the information to have varying levels of insistence as well. So far in this thesis, insistence has only been applied to goals. However, Sloman's notion of insistence is more general than this: "[there is] a need for variable-threshold interrupt filters to control the ability of new motivators, thoughts, or percepts to disturb or divert attention" (Sloman, 1992b). In particular, insistence can be applied to representations of problems. Possible roles of insistence for these other categories should be explored. In the present context, if there were causal links between valenced knowledge and goal generators or between such knowledge and m-procedures (via the Interpreter), then the insistence of the information could modulate these causal links. For instance, higher degrees of insistence of information could lead to a greater likelihood that associated goals are generated. As an example, the more insistent is the belief that a baby is dangerously close to a ditch, the more likely it should be that a goal is triggered (and this might even affect the insistence of the goals). The degree of insistence of information could also have effects on high level monitoring of situations (and possibly cause perceptual reflexes to focus on the insistent information). For instance, the system might become more interested in examining the baby (to see if it really is dangerously close to the ditch). More insistent problem information would be more likely to be processed at a management level. Insistence of problems would be similar to what is often called "salience"; except that salience is usually taken to be an objective external property whereas insistence is subjective. Many cognitive psychologists propose degrees of activation of mental information (e.g., Meyer 1971)—this too resembles insistence.

One may ask whether allowing perceptual processes to encode the valence of events would not imply an unnecessary duplication of functionality, since an important function of management processes is to evaluate situations, goals, and actions. An answer to this charge is that assessments can require more or less lengthy deliberation. Assessments that require "limited resources" of the management would be performed by the management. Those evaluations that can be performed at very little cost at the perceptual level might be worth performing there. A system might also benefit from heuristic extremely rapid mechanisms for determining whether to bother performing "perceptual evaluation". Moreover, the point of this section is not only that assessment can occur at a "perceptual level", but (as said above) that the generation of goals in response to a situation needs to be separated from the process of evaluating the situation.

There are many relevant issues related to perception, conation, and purposive behaviour that have been examined in the literature. C. Pfaffmann (1960) examines behavioural and neural correlates of "pleasurable sensations". M. Boden (1972 pp. 274-281) expounds links between a purposive agent's perceptual capabilities and its abilities to recognise the need for action and to verify whether it has achieved its goals. She notes that in order properly to index and apply its procedures to problems an agent needs to produce appropriate perceptual information. If our nursemaid could not distinguish between the recharge point and the infirmary it would encounter some difficulties. Furthermore, fine perceptual feedback is often required for the guidance of behaviour. Recently in AI it has been argued that very sophisticated patterns of apparently intentional behaviour can be produced by mechanisms that are responsive to complex perceptual features and a minimum of internal state (Agre, 1988; Agre & Chapman, 1987; Agre & Chapman, 1990; Maes, 1990a; Schoppers, 1987). Such a stance is, of course, familiar from psychology and ethology. W. T. Powers (1973) develops a control theoretic view of behaviour in which perception serves in feedback loops. L. Pryor and G. Collins (1992a) expand the thesis that perception can trigger information about interactions between a perceived object and one's plans. This information can be used to alert and direct a reasoning module to possible opportunities and dangers. There are theories of "incentive motivation" which emphasise the importance of perceptual activation of motivational systems (Bindra, 1959 Ch. 7; Bindra, 1978; Toates, 1986 Ch. 3). A central tenet of incentive theories is that perceiving something "pleasurable" can create a desire for that thing.[1] Advertising rests on this principle. Motivational priming is a related phenomenon whereby consummation of a desirable stimulus can elicit a desire for that stimulus. This is sometimes called the "peanut-effect", by analogy with the fact that tasting one peanut is often enough to trigger a bout of peanut eating. There are studies that investigate how desire for intra-cranial electrical stimulation of so called "reward centres" of the brain (Stellar & Stellar, 1986) can be elicited by the stimulation itself (Gallistel, 1966; Reid, Hunsicker, Lindsay, & Gallistel, 1973). R. Gallistel (1983 pp. 280-283) proposes that the neural systems involved in this priming effect might underpin some natural motivational priming.

6.3 Goal generators

A few improvements of the asynchronous goal generators that the system will use might be in order. One issue pertains to the identification of goals. In NML1, asynchronous goal generators whose conditions of activation are met will verify whether the goal that they would generate already exists in the system, and if it does then rather than generate a new goal they will activate the existing one. In NML1 it is assumed that goals are expressed in a canonical form, and that comparison between the potential goal's descriptor and all existing goals is made in parallel. In more complex systems, however, determining goal identity might be more difficult; this could be the case if the management uses a more complex syntax for goals than vigilational mechanisms can cope with, or if vigilational mechanisms do not have access to the Goal Database (for reasons discussed in Ch. 4). In this case, goal identification might need to be delayed until after surfacing.

In NML1 it is assumed that goal generators are always active, unless they are transient. In NML1 a transient goal generator is one that exists until its conditions of activation are met and the goal that it produces surfaces; after this the goal generator will simply be destroyed. An example of a use for transient goal generators is for discharging deliberation scheduling decisions: when an m-procedure wishes to postpone the consideration of a goal, **G**, until the infirmary is free. Here a transient goal generator will be set up and will activate **G** in due course (when the infirmary is free). There will be no use for this goal generator after it has performed its task. However, in more complex systems, it might prove useful to be able to activate or disactivate goal generators as a function of the context (e.g., if there is a need to limit the amount of monitoring of internal information). For instance, a system that can perform multiple tasks including playing chess might have a collection of goal generators that are suited for chess (e.g., some generators might detect and respond to threats to one's pieces, potential check-mates, etc.) If these generators are very specialised such that they do not apply to other contexts besides chess, then one could turn them off altogether (or decrease their

[1]At first glance this seems like an analytical proposition. But Trigg (1970) has shown that pain and aversion can be separated conceptually as well as empirically; in principle the same may apply to pleasant experience and desire. (Ultimately, whether the proposition is analytical depends on one's definition of the terms. Compare Dennett, 1978).

activation) when one is not playing chess. Later the chess goal generators could be activated by different sources in the system: e.g., when being faced with a chess display; or when high level m-procedures try an analogical reasoning strategy of "seeing the current situation as a chess playing situation". There would then be a learning task of discovering when to activate and when to disactivate goal generators. One would need to distinguish between "triggering" a goal generator (i.e., making it generate or activate a goal) and "activating" it (i.e., turning it on so that it can potentially generate goals). An interesting form of pathology would exist in systems which fail to disactivate goal generactivators even when they are no longer required. (This might be a good way of describing some aspects of obsessive-compulsive disorder. See Ch. 7.)

Another limitation of NML1 is that whereas it will produce new instances of goals, it will not produce new classes of goals. The classes of possible goals are specified by the kinds of descriptors that can be produced. What descriptors are possible is a function of the built-in synchronous and asynchronous goal generators. This limitation was anticipated and accepted because issues concerned with learning were excluded from this thesis on account of space and time limitations. Some nativist theorists might argue that it is not possible for humans to produce new classes of goals, (e.g., Piattelli-Palmarini, 1989); however, this seems to resolve to an issue of terminological preference—i.e. what is to be called a "new" cognitive entity.

6.4 The Interpreter and management processes

There are a number of respects in which the Interpreter and the management processes of NML1 could be improved.

In the literature on blackboard systems (compare Ch. 2) useful facilities for controlling KSAR execution have been introduced. Some of them could be adapted to NML1. A common feature of blackboard systems is for their schedulers to rate KSARs along the dimensions of importance, efficiency, and credibility, and to schedule them according to these ratings (e.g., in AIS). Since NML1 will support parallel m-processes, these ratings would not be used in the same way as AIS. (That is, NML1 will not have a scheduler that sequentially executes m-processes). However, there will be two sequential aspects of NML1's Interpreter that could use rating and selection mechanisms. The first one is in the **selectPars** s-procedure that was described in the previous chapter. This s-procedure will be invoked by the Interpreter when there are m-procedures that apply to a goal. It will select the m-procedure with the highest "activation value". Now this activation value could be based on a combination of ratings as in AIS. The second use is in managing conflicts between processes. Conflicts between processes have not been investigated in NML1. An example of a conflict is when two processes simultaneously request the use of a single access effector. Resolving conflicts could partly be based on dynamic ratings of m-processes.

It is important that a system with multiple objectives and multiple processes use principles and mechanisms for dealing with possible conflicts between processes that go beyond the standard techniques of computer operating systems (such as described by Lister & Eager, 1988). Procedural reasoning systems have very flexible and powerful means of controlling m-processes. In NML1, m-processes will be subjected to "bottom-up" control from vigilation mechanisms and "top-down" control from meta-m-processes that are capable of suspending and even removing other m-processes. However procedural reasoning systems were not designed with explicitly stated principles for detecting and resolving conflicts. So, although the systems have powerful control mechanisms, it is not clear how best to use them. Therefore, implementations of PRS use ad hoc tricks for dealing with conflicts (Georgeff, et al., 1987 pp. 27-28).

In contrast, issues of conflict management arising from multiple processes are being studied extensively in the blackboard literature (Bisiani & Forin, 1989; Corkill, 1989; Decker, et al., 1991; Lesser, et al., 1989). For instance, Corkill discusses the problem of "semantic" synchronisation of changes to a global database. He notes that problems can arise when multiple KSARs through time make use of information that is scattered throughout a blackboard. Simple system level synchronisation of slot access is not usually sufficient to prevent corruption of processing. He discusses a number of alternatives, including having single-write data structures (where one can write but not alter a datum), or mechanisms that allow one temporarily to lock either whole objects or "regions of the blackboard" (i.e., collections of objects). This poses problems if overlapping regions or objects can be locked, for which techniques have been developed. In a different line of research, Lesser et al. (1989) propose a taxonomy of goal relationships which are used to control problem solving. They envisage a system that can detect whether knowledge sources are working on goals that overlap (or "assist" one another), and that can inhibit some of these goals in order to reduce search. The principles involved here and others might be useful for procedural reasoning systems.

There might be reason to modify the Interpreter's dispatching. If the Interpreter's selection amongst applicable m-procedures were time consuming, then one would be well advised to try to constrain the applicability detection procedure of the Interpreter, **applicableProcedures**, such that fewer m-procedures would be applicable to any goal requiring dispatching. One could do this by designing a mechanism which affects the likelihood that a procedure will be made applicable to a goal. If there were a notion of degree of applicability of an m-procedure (as opposed to all-or-none applicability) then a filtering mechanism could be used to disregard m-procedures that are not sufficiently applicable. One can think about this in terms of m-procedures having a "field of applicability". These fields could be constricted or dilated on the basis of busyness. The tighter the field, the smaller the number of applicable procedures, and the more rapid would be the process of selection amongst applicable procedures. Developments of this idea could be investigated empirically in humans, and related to hypotheses that "breadth of thinking" narrows during periods of "stress"

(Pennebaker, 1989). (This area of psychological research could use input from a design-based perspective.)

6.4.1 Reasoning about procedures

While procedure-based reasoning has many advantages there are problems with reasoning about procedures that need to be overcome. One reason for this is that procedures typically have many conditional branches that depend on dynamic states of the world, and they can use arbitrarily sophisticated control constructs. This makes it difficult to predict the course of a procedure, and therefore it is difficult to select amongst applicable procedures on the basis of expectations of their effects. (The problem of prediction is of course not peculiar to procedural reasoning systems, but their sophisticated control constructs do imply certain complexities that are not present in system's whose plans are described by ADD and DELETE lists. (Compare Allen, 1984).) There is a need for a meta-theory that allows the description of m-procedures to facilitate scheduling and procedure selection. The method that PRS uses is simply to associate success states and failure states with procedures (i.e., the effects of the procedure if they achieve their goal or not). This is rather coarse, as it does not allow one to reason about the many intermediate states that are brought about during the course of the procedure's execution. The method that will be used in NML1 is to simulate the application of the procedure and create a temporally indexed simulated world model. This will yield a time-line, rather than branching time. According to the current design, however, the intermediate information will be used only to predict the final state; the system will need to be extended to assess the intermediate effects themselves, e.g., in order to detect whether some adverse condition is brought about during the plan but is not obvious after it has finished. (Many combinational planning systems can do this, (e.g., Sussman, 1975), though most of them can be described by ADD and DELETE lists.)

There is a further issue concerning planning. One of the boons of procedures is that they are potentially reactive, while also supporting the ability of scheduling future actions. (These abilities are of course not peculiar to procedural reasoning systems.) That some foresight is required in intelligent agents is obvious, and is not the point the author wants to make.[1] Rather, the point is that researchers have not yet devised mechanisms for dynamically tailoring the course of PRS type procedures before they execute. Not withstanding ad hoc measures, currently such systems can only plan to execute entire procedures or strings of procedures: M-procedures will not be able to decide to execute procedures in a particular way. As an example of this point, consider the case described in Ch. 5 where the goal to recharge a baby surfaces. Assume that the expansion procedure (**P1**) applied here is the one described in Procedure 5.1 (Section 5.7). Now suppose that the nursemaid decides to defer

[1] Agre (1988) makes an important argument (grosso modo) to the effect that human activity requires a large amount of run-time decision-making as opposed to the application of pre-determined projective plans.

executing **P1** until it has physically solved some other problem. According to the current design, NML1 will not be able to refine **P1**. Even if it has some spare cogitation time on its hands (e.g., if it has to hold some other baby for a few cycles), it will not be able to produce a more specific solution to the problem, unless there is some more precise expansion procedure in the m-library that it can select. Explicitly embedded in **P1** is an instruction, **G**, to move the baby to the recharge point—i.e., **G= ! position(baby) = rechargePoint**. There are many ways in which **G** could be achieved (and at **P1**'s run-time there might be many candidate procedures that apply to **G**). Nevertheless, NML1 will not be able to make more specific commitments to decide how it will execute **P1** until **P1**'s run-time. (Contrast hierarchical planners, which are specifically designed to allow expansion before run-time.) Yet, one would like NML1 to be able to realise before running **P1**, say, that in order to achieve **G** it ought to move its claw through Room **X** instead of Room **Y** because there are many babies in Room **Y**. It will perhaps prove useful to adapt solutions to this problem from other planning systems with a view to achieving flexible pre-run-time determination (or biasing) of the course of m-procedures. The resultant framework should contain a taxonomy of planning situations and solutions that are appropriate to them.

6.5 The need for a theory of decision-making—Problems with decision-theory

In Ch. 4 the main functions of management (scheduling, deciding goals, and expansion) as well as the auxiliary functions that were required for them (mainly projecting and assessing goals, plans, and situations) were discussed. In Ch. 3 a number of dimensions of assessment of goals, including importance and urgency were proposed. It was not said precisely how the assessments are to be computed, nor how they are to be represented.

R. Wilensky (1990 p. 269) writes "[the SIMULATE-AND-SELECT meta-plan] makes a number of presumptions about evaluating the cost and worth of goals and of comparing them to one another. [...] we shall not dwell on exactly how the evaluation is done. Partly this is because the details of how to do this are not completely clear; moreover, they are not crucial for the upcoming discussion." In this passage, Wilensky is pointing at the difficulty of assessing goals. Although he does not make a big fuss about it, it could be argued that he is talking about one of the biggest unsolved problems in psychology, AI and the normative and practical sciences of moral philosophy, and economics. That is, how could/does/ought one assess goals? How could/does/ought one choose amongst them? These questions need to be refined and perhaps even replaced, because (for instance) some theorists argue that decisions are not merely about goals but about how to adjust value functions (compare Ch. 3). In this section, the prevalent theory of decision-making is discussed. It is argued that there is a need for a design-based theory of decision-making. D. W. Taylor (1960 pp. 66-72) argues for a similar conclusion.

As researchers in AI have taken a greater interest in autonomous agency, there has been a need for a theory of decision-making. Many have turned to "decision-theory" to fulfil this purpose (e.g., Dean & Boddy, 1988; Dean & Wellman, 1991; Doyle, 1989; Feldman & Sproull, 1977; Good, 1971b; Haddawy & Hanks, 1990; Haddawy & Hanks, 1993; Hanks & McDermott, 1993; Horvitz, Breese, & Henrion, 1988; Toomey, 1992). Psychological decision-theoretic models, referred to as "expectancy-value" models, are also popular, although interest has waned recently. In this section decision-theory is described and criticised. A distinction is made between "standard" decision-theory and "weak" decision theory. This distinction is made in order to make as strong a case as possible for decision theory. In its standard form decision-theory is both empirically implausible and practically of little usefulness; the weak forms, however, are more viable—and the weaker the better. According to standard decision theory (French, 1986), when agents are faced with a situation, they should (1) envisage a collection of actions, and (2) select the action with the highest utility, where the utility of each envisaged action is measured according to the following equation from Haddawy & Hanks (1990).

(Equation 6.1). $$\mathrm{EU}(A) \equiv \sum_{s} \mathrm{P}(s|A,S_0)\mathrm{U}(s)$$

S_0 is the initial world state (the exact initial state, however, is not necessarily known to the agent), A is a type of action, $\mathrm{P}(s|A,S_0)$ is the probability that A when executed in S_0 will actually lead to state s, and $\mathrm{U}(s)$ is the utility of state s. Notice that the utility equation considers a collection of actions, and a collection of situations in which the actions will be performed. A relaxation of the assumptions of standard decision theory is required for it to apply to autonomous agents: i.e., these collections are not necessarily complete. That is, an agent can overlook actions that are applicable to the current situation, and he may overlook situations in which the actions may be performed.

It is useful to make explicit the abilities that decision theory requires of agents: (1) Agents can determine in a situation which actions are possible; (2) they can predict the outcomes of potential behaviours; (3) they can numerically evaluate the desirability of the outcomes; (4) they can ascertain numeric probabilities of these outcomes; (5) they can perform multiplication; (6) they can sum products; (7) they can determine which number is the highest; (8) they can determine their behaviour on the basis of utility. Strong versions of decision theory require that agents must make their actions according to the utility functions given above. In this section, it is assumed that weaker versions of decision theory (1) allow agents to use other mechanisms besides the utility function for decision-making; (2) allow for different utility functions—e.g., which allow interactions between utility judgements; (3) allow judgements of utility of actions to be influenced and biased by various mechanisms. Decision theory does not specify how the various capabilities are realised (e.g., how agents select potential actions); and weak decision theory can allow the judgements (e.g., of probability) to be wrong. Thus whereas strong decision theorists require the ability to make optimal choices (e.g., Zilberstein & Russell, 1992) in populations of cases, this is not a requirement of weak decision theory.

The central aspect of weak decision theory is that numeric measures of valence and probability are associated with actions and outcomes. These measures can be combined in different ways (e.g., with or without multiplicative weights); and other numeric measures can be used in action selection. We are not concerned here with criticising particular models, but the whole class of models, which are divided into strong and weak subsets of decision theory.

Decision theory is used because it appears to offer many benefits. A principal benefit is that it allows the comparison of very disparate alternatives on the basis of a common scale or "currency". The argument is analogous to an economic one: without a common currency, one needs a potentially large collection of equivalence functions: one for every set of thing amongst which one has to choose. (For example, a function might indicate that X pears are equal to (or are worth) 2 times X apples. From this an agent if given the choice, say, between 1 pear and 1 apple will choose the pear.) Some researchers literally claim that the human brain has reward and pain centres that evaluate the positive and negative value of things in terms of a common measure that guides decision-making (Toates, 1988 pp. 21-22). Decision theory provides an explicit mechanism for factoring uncertainty and risk. Deciding always resolves to a straightforward arithmetic comparison. Moreover, it allows interval judgements (i.e., one can specify the extent to which one alternative is better than another, instead of just knowing that one is better than another). It can thereby be used to provide fine control of behaviour (e.g., for determining how much of a good thing one should work for, and for how long). There are plenty of other advantages of working within a decision theoretic framework, if only it were practicable.

However, decision theory has been criticised both as an empirical model of how humans make decisions and as an engineering or normative model to guide decision-making for individuals, machines, and social organisations. Let us first summarise the empirical problems. The conclusion that will emerge from the empirical criticism is that many of them apply to particular models but they do not all apply to the general and flexible thrust of decision theory as outlined above. Thus, many empirical criticisms will be answered here.

It is sometimes argued that numeric models of valence are contradicted by the intransitivity of preference, e.g., (McCulloch, 1945). (Intransitivity of preference is empirically demonstrated by Tversky, 1969). However, decision theory can cope with intransitivity by postulating that assessments of importance are multi-dimensional (cf. Guttenplan, 1991 (June 4); Winterfeldt & Fischer, 1975). Another supposed problem is vacillation of preference: one knows from personal experience that one sometimes assesses an outcome as better than another, only to change one's mind. This can partly be explained decision theoretically by providing a probabilistic weight for decisions (Simon, 1959). This is similar to psychophysical theories of perception that account for one's ability to discriminate between two objects along a dimension, such as length. If **A** is only slightly longer than **B**, then one might only be slightly more likely to say that **A** is longer than **B** than to say that **B** is longer than **A**. However, this account is unsatisfactory if one believes that there are non probabilistic factors at work in such cases: namely where at one time one weighs one dimension more heavily than at a later time. Nevertheless, vacillations of preference can be accommodated by decision theory, because it does not say how assessments along the valenced dimension are performed; it only says what do with them once one has them.

Another problem has to do with the indeterminacy of preference. Although the present author does not know of empirical research on this subject, decision theory could not satisfactorily cope with the case in which an individual (1) is incapable of choosing between **A** and **B**; (2) can choose between **A** and **C**, and between **B** and **C**; (3) and claims that **A** and **B** are incommensurable. The main decision theoretic explanation of this situation is that the utilities of **A** and **B** are so similar that one cannot choose between them. However, this does not account for the supposed incommensurability as well as a model that indicates that there are no rules which apply to the given choice, or that there is not enough about the respective merits of **A** and **B**.

There are a host of other empirical criticisms. One difficulty has to do with the non-independence of the dimensions of assessment. For instance, it appears that subjects let the attractiveness of the outcomes colour their judgements of how difficult it will be to attain them (Bandura, 1989). Rather than being an indictment of decision theory in general, however, this merely implies that there are causal links between the mechanisms that compute the various numeric measures. Then there is sound evidence that in practice human probability judgements often do not abide by normative statistical theorems (Tversky & Kahneman, 1983).[1] This fact does argue against pure decision theory as an empirical model; however, in a broader framework decision theorists need not be committed to assuming that the probability estimating processes are accurate. And decision theory writ large need not assume that subjects make optimal choices. H. Simon (1959) claims that even when subjects are given clear situations in which there is a well defined optimal choice, subjects often are incapable of finding it. He supposes instead that people search for satisfactory ("satisficing") solutions. In an empirical investigation of whether people satisfice or maximise, F. Ölander (1975) provisionally concludes that when people are given sufficient time and information about alternatives, they will tend to maximise utility; however, the satisficing principle is still operative in determining whether the subjects will consider further behaviour alternatives. His point is that if an individual has to choose between two outcomes of differing subjective utility, he will

[1]The interpretation of Tversky and Kahneman's results is mute. See the target article of L. J. Cohen (1981) and his commentators.

choose the one which he thinks has the highest utility; but some mechanism is required to determine whether he keeps looking for another behaviour (with a better outcome). Subjects can apply a satisficing criterion on a mechanism that sequentially considers alternatives in a decision theoretic framework.

Another often cited problem with decision theory is that subjects seem to prefer good outcomes sooner rather than later, and that objects seem to be more attractive the closer they are (Bandura, 1989). (This is related to Dollard and Miller's notion of "goal strength"). However, G. F. Loewenstein and D. Prelec (1993) find in their paradigm that when subjects are asked to choose between entire sequences of outcomes, (as opposed to gaining experience with them gradually through successive actions) they prefer to spread valenced outcomes through time. Whether or not the methodology of Loewenstein and Prelec is sound, one might expect that some individuals are more impatient than others. All that matters from the present perspective is that the models presented in the literature that predict impatient behaviour or that predict spreading out of payoffs are expressed numerically in terms of utility. Some variants of decision theory can cope with either set of data.

There is no knock-down empirical argument against the decision-theoretic framework. Particular decision theoretic models might be falsifiable (compare Kuhl, 1982); however, the general framework or thrust of decision theory probably is not, since one can always suppose that some combination of valence and probability led to the given choice. (This is not intended as a Popperian criticism of decision theory.) Decision theory has been evolving to cope with empirical problems. In that sense it is a moving target. Opponents of decision theory are better off documenting problems with it from an engineering perspective and constructing <u>systematic</u> alternatives to decision theory than trying to show that it is false.

The main analytical considerations about decision theory are as follows. Decision theory rests heavily on the assumption of being able to make probabilistic predictions of the consequences of actions. Yet explicit and detailed prediction is a hard task. In order to make a probability estimate, one needs experience with a population of cases, and to be able to recognise that one is faced with an instance of this population. These estimates are hard to obtain with limited knowledge in a complex world. Moreover, since finding an <u>optimal</u> solution in a planning or scheduling situation will often require considering more alternatives than is feasible, usually a small number of the possible actions needs to be considered from which one would choose. (Compare Goodwin & Simmons, 1992; Ölander, 1975). The number of alternatives that is considered could be subject to an anytime algorithm.

One of the main drawbacks of decision theory lies in the assumption that information about uncertainty and value can adequately be represented in an exclusively numerical fashion. The case against numeric representation of uncertainty has been argued convincingly by Cohen (1985). P.R. Cohen notes that a lot of probabilistic measures of uncertainty do not allow one to distinguish uncertainty from ignorance. In order to reason about uncertainty, one needs to record the justifications for and against one's beliefs. A measure cannot do that. Moreover, computing a degree of belief is often not necessary if one can directly compare the justification for one's beliefs. (For example, if one knows that Bob is a liar and John is not, one might be more favourable to believing Bob's claims than Johns.) Conversely, in some situations where one has to choose between two hypotheses, one might not be able to evaluate them numerically: but one might still be able to make an ordinal judgement; thus the requirement for interval measures is sometimes too strict. Cohen designed a model of uncertainty that is based on the notion of constructing positive and negative endorsements of beliefs. Although Cohen's arguments were designed to address uncertainty, a similar argument can be made in favour of the valence component of deciding. A good topic of research would be to explore an endorsement analogue to uncertainty: endorsements of value. One could thereby make a case for not always basing one's decisions on numeric estimates of value, but sometimes on qualitative bases. To some extent such a project has begun with the work of (Agre, 1988), who provides a qualitative theory of decision-making in terms of "running arguments". Here, an agent is viewed as producing arguments in favour of or against actions. The agent forms an intention on the basis of the current state of the argument.

Using qualitative information sometimes can save time and produce better responses than using decision theoretic methods. Consider NML1's decision-making. In order to select between two possible schedules, or two courses of action it will use at least three different methods. The first method will predict their effects (which it represents by descriptors, some of which denote valenced facts). On the basis of these predictive descriptors it will compute a utility measure for each schedule or action. It will choose one solely on the basis of the utility descriptors. (This is similar to Goodwin & Simmons, 1992.) By using the second method, it will be able to recognise that one action is better than the other on the basis of its descriptors, without having to make utility judgements for the descriptors. With the third method, it will not even have to make projections: it can directly decide on the basis of the prospective schedules which one is preferable. This last method, although riskier, will be more economical, especially if projection is difficult and time consuming.

NML1 could make better choices if it used its intermediate predictive descriptors to adjust its potential actions, rather than simply making a choice between actions based on the collection of predictions of their consequences (or on utility measures based on them). Consider the following hypothetical scenario. There are two goals which the nursemaid has to schedule. **Goal1** is to dismiss an old baby (say, babyA). **Goal2** is to heal a sick baby (say babyB). The nursemaid will retrieve a plan for each problem. Then it will attempt to schedule them. To do this, it will apply an m-procedure that considers the order **(Goal1, Goal2)** and the order **(Goal2, Goal1)**, predicts the outcomes of each, and selects amongst them. NML1 will be able to predict the problems with each of these schedules; it will only uses this knowledge to decide amongst them. It would be better if instead of simply rejecting one ordering in favour of another, it could fix the problem with the first ordering. Suppose that the nursemaid anticipated that a problem with the second ordering is that in order to bring babyB to the infirmary it would go through the room in which babyA and other babies are, and that since this room is almost overpopulated there would be overpopulation and a risk of thuggery developing. Instead of simply using this fact as a disadvantage of a schedule, it could be used to modify slightly the plan for goalB by making the nursemaid pass through a different room. In this scenario this might lead to a better plan/schedule than the first ordering that was considered. This involves the violation of a soft constraint. Pandora (Wilensky, 1990) has a similar capability to the one described here. That system is capable of generating ("detecting") goals while projecting the effects of its plans, and modifying its plans on the basis of these anticipated interactions. However, Pandora uses ad hoc mechanisms for re-planning (and ultimately makes its behavioural decisions based on quantitative information). There is still a need for a theory of decision-making based on qualitative as well as quantitative information.

But decision-theorists can respond to the idea of this adjustment mechanism by saying: "To say that qualitative information plays a large role does not touch decision theory so long as the actual decision is still made on the basis of utility measures associated with actions." This is true. Still, it has not been shown that utility based decision-making is superior to other forms of reasoning; and it still seems that non decision theoretic reasoning about actions is important.

Perhaps the main problem with decision theory is its assumption that an agent is capable of evaluating outcomes in terms of their value. For a very simple agent, determining the value of a state is straightforward, if it has access to a built in evaluation procedure that was given to it by its designer. Ironically, however, for autonomous agents with a highly sophisticated cognitive architecture (in particular humans) the situation is not that simple at all. An autonomous agent has a collection of motives. But how is it possible to determine their relative importance if that has not been done for him? The decision theorist might reply that the burden is on the human designer to give the machine a value function. However, the designer himself might not know. This brings us back into the realm of moral philosophy. The indeterminacy of future outcomes and of value has been argued by existentialists such as Jean-Paul Sartre (1947). Whereas these philosophers are partly correct in claiming that there often is no objective basis for making a decision (or objectively valid decision-making principles), it is important to note that <u>given</u> some assumptions, some situations can objectively be decided. For instance, if one knows that state **A** is more valuable than state **B**, one can make objectively valid decisions about which other states (subgoals) to select, provided one knows about the causal relations amongst the sub-states. However, this line of argument will not be followed here since that would lead to a protracted discussion.

J. A. Feldman and R. F. Sproull (1977) claim that decision-theory provides a "principled" method for making decisions, and they contrast it with "heuristic" methods. (A. Tversky (1983) makes a similar contrast between normative and heuristic mechanisms). It is ironic that in that very paper, they provide a number of heuristics for applying decision theory! Moreover, in the case of autonomous decision-making in general, there is no way to achieve "principled" decision-making, if that requires algorithms that are assured to achieve the optimal solution over a population of situations. "Pure" decision theory can be enhanced with various "heuristic" techniques, as well as techniques that take qualitative information into consideration. For instance, one can design a machine that has a component for making decision-theoretic judgements at run time but where these judgements can be overridden by a mechanism that detects that a particular response is likely to be useful without numerically evaluating utility or otherwise using utility measures. NML1 uses this principle. Thus one need not completely reject or completely adhere to a decision-theoretic framework.

Still, there are many unsolved problems for decision theorists and others. There are few detailed general theories about how to make assessments of importance. Decision theorists such as P. Haddawy and S. Hanks (1990) recognise this: "The problem of assessing utility functions, especially the goals' utility of satisfaction functions and the residual utility function, still remains. The difficult task is to generate, for each new planning problem, utility functions that accurately reflect the agent's current and expected future objectives and resource needs" (p. 57). Moreover, where there is no well defined entailed preference, a system needs to be able "sensibly" to determine its preferences. At least humans seem to develop preferences that are subjective and not clearly related to top level objectives. There is a need for theories to account for apparently arbitrary preferences.

6.6 Conclusion

Once the improvements of the design are made and NML1 meets its present requirements, a number of extensions of the requirements would be worth investigating. There could be more varied forms of sensation. Information gathering could vary in complexity and resource requirements. A version of the scenario was explored that had more than one claw. This requires more complex control and co-ordination capacity, and action thereby consumes more management resources. There could be a greater variety of positive and avoidance goals differing in their urgency functions and the actions required for them. One would need to study the issue of mental resources in more detail, both to find what kinds of management parallelism can in principle be investigated, and to model human attention. There is plenty of scope for learning, many of the possibilities have already been mentioned; but there is also room for developing new goal generators, new goal comparators, new heuristics for assigning insistence, scheduling, deciding and expanding goals, new plans, new or improved feedback control loops during actions, new concepts, etc. This thesis has focused on goals, but other kinds of motivators and control states discussed in Ch. 3 should also be investigated (e.g., "personality", and "moods"). When these improvements have been made it will be possible to perform analytical studies and computer simulations to determine whether the resultant models produce attentional errors of the types that occur in humans, such as forgetting an intention while continuing with an action procedure (Norman, 1981; Reason, 1984), and whether marring NML1 can lead to management deficiencies seen in brain damaged humans (Shallice, 1988 Ch. 14 and 16; Shallice & Burgess, 1991). Perturbance and pathologies of attention might also be studied, as suggested in the following chapter.

This chapter assumed a system that builds directly upon an existing clearly specified architecture, PRS. The space of possible designs is large, and there is no point in committing ourselves to a single basic design from which to build. Different designs could have been explored. For instance, it is worth asking how the AIS architecture would need to be modified in order to support the goal processes assumed here. Moreover, it might be worth trying to elaborate the architecture supposed by the Communicative theory of emotions so that it could run these processes. These experiments would lead to improvements in both the process specification and the proposed architectures. But we should not limit ourselves to using extant designs as bases. New architectures should be explored (possibly combining features of existing autonomous agent architectures). The objective of such research would be to provide a "map" of the space of possible designs.

The comments in this section pertain to the NML1 architecture in particular. The following chapter outlines proposals for future research on the general view of affect and attention proposed by Sloman and expanded in this thesis.

Chapter 7. Conclusion—summary of progress and directions for future research

This research has clarified and extended Sloman's theory of motive processing, and Georgeff's Procedural Reasoning System. Some of this progress is summarised as follows. A number of different areas of research in psychology and AI that are relevant to goal processing in autonomous agency were reviewed. A systematic notion of goals was proposed which included not merely quantitative dimensions of goals, but many qualitative features as well. Principles for distinguishing between control states were proposed. Processes involving goals that had been investigated by Sloman were organised by the author in two categories: vigilation processes and management processes. The functions of these classes of processes were further subdivided. For instance, a distinction was drawn between the main function of management processes and their auxiliary functions. It was shown that the state-transitions of management processing in an autonomous agent are more complex than previously thought—in particular surfacing does not necessarily immediately lead to meta-management. The notion of insistence filtering was originally proposed by Sloman. In this thesis, a distinction between intentional and propensity interpretations of insistence was drawn. A number of other functions of filtering besides "acute" management protection were proposed. Design-based reasons for the assumption of limited management resources were discussed, though more research is required to determine precisely what and when limitations are required. All of these contributions represent progress towards understanding goal processing in autonomous agents.

PRS, an architecture from the AI literature, was adapted to discharge some of the goal processes described in Ch. 4. This illustrated some of the strengths of PRS and extant theories of goal processing, but it also uncovered a number of new problems with them, which were discussed in Ch. 6. In particular, it was argued that a separation should be made between the description of problems and opportunities and the conative processes responding to them (<u>e.g.</u>, goal generation, insistence assignment, and management processing); reasoning about procedures is difficult; and a theory of decision-making is needed for the design of management processes. Moreover, it is unclear how best to control management processing. Resolution of these theoretical problems will permit future researchers to propose improved architectures and mechanisms for designing autonomous agents. Expounding these problems constitutes a first step towards solving them.

Useful new conceptual generalisations and terminology were proposed. For example, the concept of <u>urgency</u>, which was previously conceived as the amount of time before it is too late to satisfy a goal, was generalised to the notion of a function which describes the importance and cost of acting at different time points. <u>Busyness</u> generalises the generalised notion of urgency of goals by applying it to whole situations. The notion of <u>criticality</u> of a goal (or plan) to a goal was developed. This generalises from the notion of absolute necessity of a goal to the notion that a goal may be more or less necessary (critical) to another goal. Thus necessity becomes a special case of criticality. There was a need for a concept of surfacing of a goal, which means that a goal has successfully passed the filtering stage and is about to be managed. The term dispatching was defined as the process of selecting m-procedures to apply to goals. The transitive verb to decide was technically defined as determining the adoption status of a goal. The author also played a role in the development of other terminology and conceptual refinements used in this thesis.

The research for this thesis was conducted in a design-based fashion, as briefly described in Ch. 1. Programming was used to comprehend, test, and develop various ideas in the literature. The implementation of the NML1 design is not complete, and hence is not reported in this thesis. However, Ian Wright of the University of Birmingham is currently working on a simulation based on my specification, suggesting that—although some details need to be revised and extended—the specification proposed here is implementable.

The importance for cognitive scientists of working at the design level with knowledge of programming techniques and experience in implementation needs to be underscored. Even without fully implementing a model one can learn from the process of design. This process helps one to uncover theoretical gaps and conceptual flaws; it also helps one to suggest new and possibly more general principles, mechanisms, and stores of information. Of course, if the designer does not have sufficient knowledge of computer implementation he can easily fall into the trap of producing a theory which is too vague, inconsistent, or clearly not implementable. This is not to say that implementation is useless, only that sometimes it is worth delaying implementation while still making progress.

7.1 Future research

Many avenues for future research were discussed in previous chapters. There is a need for a theory on how best to control management processes, to determine which of the types of management objectives should be pursued (e.g., deciding, evaluating, or expanding goals). Existing work on opportunistic planning might give some clues (Hammond, 1989; Hayes-Roth, 1992; Hayes-Roth & Hayes-Roth, 1979). Many issues which have been addressed for non-autonomous agents need to be re-addressed for autonomous agents: e.g., dependency maintenance, planning, scheduling (Haddawy & Hanks, 1990; Prosser, 1989). There is a need for a theory which prescribes how information about the urgency, importance, intensity and criticality of goals, as well as assessments of situations and plans, should be generated and utilised to determine the course of management processing. In particular, given a critique of utility-based decision-making, a theory of qualitative decision-making is required that shows how agents can choose amongst actions on the basis of predictions of their possible consequences.

It was shown that mechanisms for determining insistence require further research. This will benefit from development in theories of importance and urgency, for the intentional interpretation of insistence is given in terms of (heuristic measures of) importance and urgency.

Empirical research could shed light on autonomous agency. Two avenues are discussed here. The first is an in depth study of human experts at playing a computer game version of the nursemaid scenario (i.e. where a person plays the role of the nursemaid). This could include a "knowledge engineering" component. The computer investigation should start with an open-ended pilot study in which subjects think aloud while performing. The speed of events in the nursery could be decreased to compensate for the extra demand of talking while acting. This study would improve all aspects of this research by: clarifying the requirements of autonomous agents, improving the architecture (e.g., there might be a need to better integrate visual perception and action, or a need for "cognitive reflexes", etc.), and suggesting new capabilities that had previously been overlooked such as new goal generators, new ways of designing m-procedures, criteria for controlling their state transitions, etc. This might help to determine the control conditions for management processes: e.g., given a surfacing goal what type of management process should operate over it, and how should it come to its conclusion.

The second avenue is to study humans in the field in settings where requirements of autonomy are particularly salient: e.g., hospital surgeons and anaesthetists, air traffic controllers, military commanders, people managing hospitals, or factories, or even real nursemaids. This too would yield results in all aspects of the study of autonomous agents. If one studied real nursemaids, one would investigate not only the strengths of their abilities, but also limitations on these abilities. For instance, one could characterise how the quality of care which a nursemaid provides varies with the number of babies that she nurses. The author might have made his own life too difficult when designing NML1 because he tried to design management algorithms that could cope with arbitrary numbers of babies. In practice, it appears that nursemaids only look after small numbers of babies. For instance, legislation in the province of Ontario (Canada) concerning day-care centres prohibits the ratio of infants to nursemaids to be greater than four to one.

7.2 Attention and affect

This thesis is part of the "Attention and Affect Project" of the Cognitive Science Research Centre of the University of Birmingham. The objectives of the Attention and Affect Project are to determine and address the requirements of autonomous agents. This thesis can be seen as a "breadth first" approach to the project objectives—rather than to focus on one of the requirements, the author examined many—but not all—of them.

The main hypothesis that drives the project is that autonomous agents, defined as agents that face the requirements listed in the Section 1.2, are likely to find themselves in states of "perturbance", at least occasionally (Sloman, 1987; Sloman, 1992b; Sloman & Croucher, 1981). An implication of the perturbance hypothesis is that it is not possible to design an autonomous agent which is not subject to perturbance. As mentioned earlier, perturbance is a state in which insistent goals tend to disrupt attention. Thus perturbance involves both attention and affect. The author has not given many details about perturbance in the core of the thesis, nor has he mentioned the perturbance hypothesis, because it is part of the logic of the hypothesis that before studying perturbance one needs to have an understanding of goal processing in autonomous agents.

In the rest of this chapter, perturbance is discussed. In the following section the relation between perturbance and "emotion" are noted. In the subsequent section, ways of studying perturbance are suggested. In the last section, prospects for explaining an anxiety disorder are outlined. Given the preliminary and speculative nature of this concluding discussion, the author allows himself to use colloquial terminology (e.g., "thoughts", "emotions") along with the technically defined terms. Future research will involve more precise concepts. Many important features of perturbance already have been explained and will not be discussed here. (Compare Sloman, 1992b).

7.2.1 Perturbance and "emotion"

Let us extend the hypothetical scenario described in the introduction. Recall that Tommy fell off his chair. Tragically, he knocked his head on the hard tile floor, suffered intracranial bleeding, fell into a coma, and died from the injury. Subsequently, the nursemaid went through a severe period of grieving including feelings of remorse. For years after the event she would find herself remembering the tragic moment, thinking about what she could or should have done to prevent the accident, wishing she could turn back the hands of time, feeling for Tommy's parents, etc.. These thoughts and feelings came to her despite her best efforts to rebuild her life and forget about the calamity.

Although fictional in detail, this scenario is realistic and analogous to actual human experience. (See Kuhl, 1992; Tait & Silver, 1989; Uleman & Bargh, 1989) for empirical perspectives on this. It illustrates the main characteristic of the state of perturbance: a loss of control of one's own mental processes, as they are repeatedly drawn back to thoughts and desires related to the object of the perturbance. Numerous other colloquial examples could be adduced. For example, a similar phenomenon is involved when a person is "romantically in love" and keeps thinking about his darling, wanting to be with her, planning what to do with her next, etc.. The greater the infatuation, the less control he has over his thought processes and the less able he is to not think about her.

The relationship between emotion-like states and loss of control of attention has an historical precedent.[1] It implies a distinction between mechanisms that can divert attention (e.g., "vigilation" mechanisms), and mechanisms that are attentional (e.g., "management" mechanisms). More generally, it implies a notion of mind as comprising multiple modules or agencies that can have incompatible tendencies and actions. (Compare Minsky, 1986). For instance, in the course of arguing that enjoyment is not a passion, G. Ryle (1954) draws an analogy between the political state and the mind. Having described break-downs in law and order of a state, he likens "passions" to these break-downs:

> We need not trouble ourselves here to look for unpicturesque paraphrases for the representations of control and loss of control of fury and terror in terms of the maintenance and breakdown of law and order. [...] to revive a now rather old-fashioned word, we give the title of 'passions' to the potentially subversive agencies in a man, namely terror, fury, mirth, hatred, disgust, despair, and exultation [...] Terror, fury and mirth can be paroxysms or frenzies. A person in such a state has, for the time being, lost his head or been swept off his feet. If a person is perfectly collected in his deliberations and movements, he cannot, logically cannot, be described as furious, revolted, or in panic. Some degree of temporary craziness is, by implicit definition, an internal feature of passion, in this sense of 'passion'. (p. 65)

This quotation is highly instructive because (1) it illustrates the importance, and historical precedent, of "loss of control" as a feature of emotion or passions, and (2) it contains a subtle equivocation[2] that is at the heart of much confusion in discussions about emotions. The equivocation is that Ryle is referring to passions both as the agents which (can) cause a subversion—i.e. which get out of control (as Ryle implies)—and as the subversion or loss of control itself. When he says "temporary craziness is, by implicit definition, an internal feature of passion" he implies that one cannot have a passion while being in control (e.g., having a dormant passion). However, he also says that passions are "potentially subversive", which implies that in principle they can be controlled; in this sense passions are more like motives which may or may not be controlled. One needs a theory of mind in order to make the distinction clear between motives and loss of control of thought processes.[3] Such a theory (as Sloman's) might claim that there are "passions" (in the sense of motives) which can be more or less insistent, and there are management processes which operate on the passions (motives).[4] The passions (now as loss of control) happen when the passions (as motives) are very insistent. With such a theory, motives that vary in insistence can be characterised. Now prima facie, one would not

---

[1] This is not the place for a review of the literature on emotion and attention. See Mandler (1980) for a history of "interruption theories of emotion", Oatley (1992) for a theory that emphasises loss of control in emotion, Mahoney (1991 Ch. 8) for a claim that "direction of attention" is the primary function of emotion, and Frijda (1986) for a scholarly review of the literature on emotion. Simon (1967) proposed some of the key ideas of information processing theories of emotion, including Sloman's.

[2] Despite the following criticism, Ryle is well worth reading. Furthermore, it is not clear from the text whether Ryle is unaware of the distinction, or whether he has merely failed to make the distinction explicitly within the text.

[3] Ryle never explicitly says that it is the thought processes that are out of control, he talks about the "passions" being out of control.

[4] Ryle seems to be referring not only to insistent motives but also intense ones, in the sense defined in chapter 3.

say that a state of perturbance can get out of control, since perturbance is the loss of control. However, on closer examination, states of perturbance can be more or less severe—some will be easier to harness than others via meta-management strategies. The (meta-) management processes for controlling perturbance once it has been detected would be a fascinating further study. The capabilities for detecting perturbance could figure in a control system which enhances or reduces perturbance (e.g., some people can make themselves angry in order to deal with others whom they fear). Furthermore, recall that a distinction was made between latent and manifest perturbance (as done in Ch. 4), where manifest perturbance involves motives that actually divert and possibly maintain attention. (R. M. Gordon, 1987, Ch. 6) expounds the thesis that emotions do not act upon us, but involve something else which acts upon us.)

It should be noted, furthermore, that although Ryle characterises these passions as subversive, one should not impute to him the view that they do not serve a useful function. Protests, revolts, and even revolutions can improve a political state! Most theories of emotion assume that emotions have a function (e.g., according to Frijda, 1986; McDougall , 1936; Oatley, 1992; Oatley & Johnson-Laird, to appear; Swagerman, 1987) they help to meet requirements of autonomous agency). According to Sloman's theory, however, perturbance is a by-product of mechanisms which have a function, but is not intrinsically functional or dysfunctional—it is afunctional (sometimes emotional states are good, sometimes they are bad). However, although the issue of functionalism is the subject of heated debate, it appears to resolve to a mixture of terminological preference and different design decisions. The terminological preference relates to the criteria one selects for applying the label "emotion". Some theorists reserve the word "emotion" for certain kinds of temporary control states (e.g., Sloman), whereas others use the word "emotional" as an epithet for a collection of mechanisms which perform control functions. The design issue is whether emotional states are only due to special mechanisms which operate only in cases where emotions are present (e.g., Oatley and Frijda), or whether the mechanisms which produce emotions are also active in a wide variety of circumstances many of which do not involve emotional episodes. Still, the question of whether perturbance is functional, afunctional, or dysfunctional is pernickety and will be dealt with in more detail in future publications (Beaudoin, 1993).

It is necessary at this stage to emphasise that definitions of "emotion" are extraordinarily controversial. To be sure, there are many other features besides loss of control of attention that are more or less reliably associated with what is commonly referred to as "emotion" (e.g., various permutations of: cognitive evaluation, interruption of thought, facial expression, activation of prototypical behaviour patterns, autonomic arousal, etc.) Although it is tempting and common to argue about what the real features of "emotion" are, members of the Attention and Affect Project have concluded that it is overly optimistic to think that agreement will come about on this matter and also that it is foolish to think there's any right answer. Furthermore, too much debate in the literature on emotion actually resolves to differences in opinion about the "right" definition of the term emotion. This has been futile and even counter-productive. Given such a controversy it really does not matter what particular sign one uses to refer to the phenomena in which one is interested: i.e., the phenomena matter more than the terms that are used to described them.[1] Therefore, rather than argue in vain about what "emotions" really are, we merely study a certain kind of state which we have defined and to which we have given a new technical label, namely "perturbance".

7.2.2 Towards a study of perturbance

There is no elaborate theory of perturbance yet. To achieve this one must first propose a way of studying perturbance. This can be done from the design stance and through the use of phenomena-based methods. Once a nursemaid has been designed and implemented which meets the requirements, it will be possible to examine perturbance in this context. This work will have two interacting stages. One is to characterise perturbance theoretically, before making observations with the simulation. This requires proposing criteria for attributing perturbance, dimensions of variation of perturbance, ways of detecting it and possibly measuring it. On the basis of this, one can analytically determine when perturbance will occur in an architecture. The second stage is to run computer simulations and observe human beings.

Many of the relevant dimensions and features of perturbance have been mentioned above. They include whether the perturbance is latent or manifest, and how insistent the goals are that are causing the perturbance. Notice that NML1 will be a "goal directed" architecture, and perturbance has been defined in terms of interactions between goals and management processes. However, if some of the improvements discussed in Ch. 6 are incorporated in the design, it will be possible for other factors to be involved in perturbance. For instance, if there is a notion of "problems" that is separate from goals, and if they have insistence levels, then they might contribute to perturbance not only through filtering mechanisms but through the other mechanisms proposed in Ch. 6. Analytically characterising perturbance is a matter for future research.

In order to study perturbance in a simulation of a nursemaid (or any other autonomous, resource limited, agent acting under stressful conditions) one would have to accomplish the difficult task of separating those features that are attributable to the implementation from those that are truly consequences of the design. (See Cooper, Farringdon, Fox, & Shallice, 1992). Then, one might address questions such as "How does the frequency of interruptions affect the quality of performance?", "How long does it take to deal with interruptions?", and "How does the cost of interruption of management processes by the surfacing of a goal affect filtering?" This last question is

---

[1]Nevertheless it is fruitful to point out equivocations, and to perform conceptual analyses of emotion (Gordon, 1987; Ortony, et al., 1988; Ortony, et al., 1987; Sloman, 1987), provided one does not then go on to say that one of the meanings of the equivocal term is the right one.

difficult to answer. In principle, one would only want surfacing to occur when its benefits outweigh its cost. But it is non-trivial precisely to ascertain costs and benefits; this is true not only for a resource-limited agent, but also for an observer of the agent. (Compare the discussion on decision theory in Ch. 6 where it was argued that there may not be a common measure of utility.) When characterising an episode of perturbance one would also want to know whether the goal that surfaces is one which the management would tend to reject in the current circumstances or not: i.e., is the system dealing with an adverse or a welcomed interruption. M. E. Bratman (1988) develops a similar notion for describing attention filters.

In order to develop a notion of loss of control of management, one needs to have a theory of what it means for the management to be in control of itself. This is left for future research.

One way to study perturbance is to develop a mechanism for tracking and describing perturbances, e.g., to give the nursemaid reflective abilities. The process specification has already supposed an ability to detect perturbance, which should lead to the activation of a meta-management process which would decide either to satisfy the perturbing goal or to suppress it. Design space is partitioned as follows. Architectures vary according to whether they can detect perturbance and respond to it. They also vary in their criteria for detecting perturbance and their response to it. Systems that can explicitly postpone consideration of goals may or may not be able to detect whether their attempt to implement these meta-management decisions have been successful. Those that can may (or may not) be able to respond to failure—e.g., by increasing their effort or adopting different strategies, and possibly learning from this. Such meta-management capabilities appear to be very sophisticated and it will be interesting to see how difficult it is to design systems that use them.

Of course, human beings are not always aware of their own perturbances. And when they are they are not necessarily very good at appeasing them. Some have even argued that the very attempt to prevent certain thoughts from coming to mind leads to an opposite effect (Wegner & Schneider, 1989; Wegner, Schneider, Knutson, & McMahon, 1991). D. M. Wegner and his colleagues asked subjects to verbalise their thinking. They were then asked not to think of white bears for a five minute period, but to ring a bell when they did think of a white bear. Subjects rang the bell a mean of 6.1 times. In another experiment, one group (Group A) was asked not to think of white bears, whereas another group (Group B) was told to think of them. Subsequently, subjects were told they could think freely for five minutes. Group A had a significantly greater level of thinking of white bears than Group B (15.7 bells vs. 11.8). Variations on this experiment were performed; nevertheless, no convincing explanation has been proposed and tested. Assuming that these data indicate that attempts at thought suppression (and perhaps the control of perturbance) are often counter-productive, one could ask (from the design stance) "What are the designs that have this feature as a consequence?"

That is, such a feature might not be functional in itself, but is it a consequence of design features that are functional?

An empirical investigation of perturbance could be performed in the two settings mentioned previously—i.e., looking at people playing the role of a nursemaid in a computer simulation of the domain; and real nursemaids in day-cares or hospitals or many other activities besides looking after children. One could try to discover whether there is evidence that subjects filter out goals that they produce, or whether filtering is more "implicit" (e.g., they do not even produce or activate goals that would not be sufficiently insistent). Methods of ascertaining this would be developed. If filtering could be identified, then it would be useful to determine whether the conditions that are supposed to affect filter thresholds (cf. Ch. 4) are actually used by humans. Would examples of perturbance be seen in these settings? Would subjects occasionally find it difficult to postpone consideration of goals, and indulge in considering goals that they know are not currently as worthy of consideration as others? If so, then under what conditions does this happen, and how could the effect be enhanced? Similar questions could be asked of real nursemaids (or other autonomous agents).

7.2.3 Perturbance and obsession

A cogent theory of normal psychological phenomena should shed light on how mental mechanisms can break-down into pathology. In this respect, and to conclude this thesis, it is hoped that the study of perturbance could be used as a basis for understanding some of the core features of obsessive-compulsive disorder (OCD).[1] OCD is a debilitating anxiety disorder marked by "obsessions", i.e., "persistent ideas, thoughts, impulses, or images that are experienced at least initially, as intrusive and senseless (for example, a parent having repeated impulses to kill a loved child, or a religious person having recurrent blasphemous thoughts)" (American Psychiatric Association, 1987 p. 245). Compulsions are stereotyped purposive responses that aim to attenuate obsessions. OCD has been investigated empirically for over a hundred years (Barlow, 1988; Emmelkamp, 1987; Kozak, Foa, & McCarthy, 1988; Teasdale, 1974; Toates, 1990), and empirically the phenomenon is well understood; but there has yet to be design-based explanation of the phenomena as following from requirements of autonomous agency. The models that have been proposed are quite coarse grained, and often behavioural or biological.

It might be possible to characterise OCD as a great susceptibility to perturbance. Obsessions themselves can be viewed as states of perturbance. There is definitely a loss of control of attention in OCD. D. H. Barlow (1988) notes that "patients with OCD are most often continually buffeted with aversive, out-of-control, unacceptable cognitive processes" ([emphasis mine] p. 614). Thus the

---

[1]Other disorders involving affect and attention should also be examined, such as a condition that has received much attention in the child psychopathology literature: attention deficit disorder with hyperactivity (Barkley, 1988).

obsessive mother cannot control the surfacing of her desire to kill her child. But the obsessive (perturbing) desires do not necessarily trigger behaviour. It's as if the insistence of some of the desires were too high, but the intensity was under control—e.g., the mother does not have a behavioural inclination to kill her child (at least the intensity of the desire is negative, not positive). In contrast, the compulsive aspect of OCD involves desires which may be unimportant but are positively intense—e.g., the mother may compulsively pray in order to rid herself from her insistent desire even if she sees this prayer as having no operational benefit. Thus, in the first instance, the vocabulary of goal processes might provide a way of characterising obsessions.

One would need to give more architectural details of OCD. Here are a few very sketchy ways of doing this that require a more sophisticated design than NML1. Obsession as perturbance might result from impaired descending control of goal generactivators—e.g., perhaps normally goal generactivators are disactivated when goals are satisfied or rejected, but in OCD the disactivating mechanism fails. (As noted in Ch. 4, even normal people do not have complete control over their generactivators.). There might be impaired insistence assignment, or impaired filtering. Goals that should not be insistent (under both the intentional and propensity interpretations) are insistent. Or perhaps obsessions follow from impaired functioning of the interpreter, which in its dispatching phase favours non-insistent goals over goals which should be triggering management processes. Or perhaps obsessions could result from management processes that go awry and although the interpreter or other management processes try to suspend or destroy them, they keep on executing. These are just concluding suggestions, and more research is needed to develop a convincing explanation.

As with emotions, there are many features of obsessions besides perturbance, but unlike emotions there seems to be a consensus that loss of control of attention is the defining feature of obsessions. This makes it an alluring topic for future research for the Attention and Affect Project.

Appendix 1

Specification of valenced descriptors

NML1 will be able to direct its behaviour on the basis of predictions of world states following possible choices. The "predictions" have two components: factual world model information, and a list of "valenced descriptors" which denote information that is relevant to the decision making process. The valenced information could already be implicit in the factual descriptors, but the valenced descriptor is nonetheless useful in underscoring a certain fact.

NML1's conception of the world (its ontology) will be broadly divided into two kinds of relations: enduring states (which can contain change, and hence implicate processes), and events. States have a duration, but events do not. NML1 can attach probabilities to events and states. Probabilities range from 0 to 1. Probabilities and durations can be unknown. A Prolog syntax is used to express relations.

States are represented as ternary relations:

state(Relation, Duration, Probability)

And events are represented as binary relations:

event(Relation, Probability)

Here follows a list of relations. Keep in mind that the relations can either figure in event or state descriptors. For example, the relation **dead(Baby)** can figure in a state as **state(dead(Baby), Duration, Probability)** or in an event as **event(dead(Baby), Probability)**—the former indicates that a **Baby** is dead whereas the former indicates that a **baby** dies. Moreover, the predicate's arguments can be single objects or lists of objects. For instance, the relation, **dead(Baby)** denotes that **Baby** is dead, whereas the relation **dead(Babies)** denotes that a number of babies will die, namely those denoted by **Babies**. For economy of space, the following list is given in the plural form only, except for those relations which NML1 will only conceive in the singular form.

dead(Babies)
This indicates the belief that **Babies** are dead or will die.

ill(Babies, Illnesses)
This indicates the belief that **Babies** are ill, or will become ill. The argument "**Illnesses**" will usually be a list containing the names of the specific illnesses involved.

injured(Babies, Injuries)
This indicates the belief that **Babies** are injured, or will be injured. The argument "**Injuries**" will usually either be a list of injuries (if the descriptor is a state) or a number representing the number of injuries which will be inflicted.

lowCharge(Baby, FinalCharge)
This indicates the belief that the charge of **Baby** is or will be below a critical threshold at the beginning of the predicted interval. **FinalCharge** represents the projected charge of the baby at the end of the predicted interval (denoted by **Duration**).

old(Baby, FinalAge)
This indicates the belief that **Baby**'s age is or will be greater than the dismissal age at the beginning of the interval. **FinalAge** represents the age of the **Baby** at the end of the interval of prediction.

thug(Baby)
This indicates the belief that **Baby** is or will become a thug.

Ch. 5 contains instantiated valenced descriptors.

NML1 will use a look-up table which maps valenced descriptor patterns onto utility functions. Thus, for instance, there will be a utility function which is used for descriptors matching the pattern **event(ill(Babies, Illnesses), 1).**

List of abbreviations

AI: Artificial Intelligence

ED: Effector Driver of NML1.

AIS: Adaptive Intelligence System proposed by B. Hayes-Roth (1993).

BBS: Blackboard system.

DCA: Dynamic Control Architecture blackboard system developed by B. Hayes-Roth (1985).

EPP: Expansion prediction procedure used by NML1

GD: Goal database of NML1.

KSAR: Knowledge source activation record.

NML1: Nursemaid design proposed by Luc Beaudoin.

OCD: Obsessive-compulsive disorder.

PP: Prediction procedure.

PRS: Acronym for the Procedural Reasoning System described in (Georgeff & Lansky, 1986).

PWM: Predictive world model

EPP: Expansion prediction procedure.

PP: Prediction procedure.

SAS: Supervisory Attentional System (Norman & Shallice, 1986).

References

Agre, P. E. (1988). The dynamics structure of everyday life (AI No. 1085). Massachusetts Institute of Technology Department of Computer Science.

Agre, P. E., & Chapman, D. (1987). Pengi: An implementation of a theory of activity. In Proceedings of the Sixth National Conference on Artificial Intelligence, (pp. 268-272). Seattle: AAAI.

Agre, P. E., & Chapman, D. (1990). What are plans for? Robotics and Autonomous Systems, 6, 17-34.

Allen, J. F. (1984). Towards a general theory of action and time. Artificial Intelligence, 23, 123-154.

Allen, J. F. (1991). Temporal reasoning and planning. In J. F. Allen, H. A. Kautz, R. N. Pelavin, & J. D. Tenenberg (Eds.), Reasoning about plans (pp. 1-68). San Mateo, CA: Morgan Kaufmann Publishers.

Allport, A. (1987). Selection for action: some behaviorial [sic] and neurophysiological considerations of attention and action. In H. Heuer & A. F. Sanders (Eds.), Perspectives on selection for action (pp. 395-415). Hillsdale, NJ: Lawrence Erlbaum Associates.

Allport, A. (1989). Visual attention. In M. I. Posner (Eds.), Foundations of cognitive science (pp. 631-682). Cambridge, MA: MIT Press.

Allport, A., Styles, E. A., & Hsieh, S. (In press). Shifting intentional set: exploring the dynamic control of tasks. In C. Umiltà & M. Moscovitch (Eds.), Attention and performance XV: 'Conscious and nonconscious information processing' MIT Press.

Allport, G. (1961). Pattern and growth in personality. New York: Holt, Rinehart and Winston.

American Psychiatric Association (1987). Diagnostic and statistical manual of mental disorders (Revised third ed., Revised). Washington, D.C.: APA.

Anderson, J. (Ed.). (1989). Pop-11 Comes of Age. Chichester: Ellis Horwood.

Aristotle (1958). Nicomachean Ethics. In J. D. Kaplan (Eds.), The pocket Aristotle (pp. 160-274). New York: Washington Square Press.

Ash, D., Gold, G., Seiver, A., & Hayes-Roth, B. (1992). Guaranteeing real-time response with limited resources (KSL Report No. 92-04). Knowledge Systems Laboratory, Department of Computer Science, Standford University.

Austin, J. L. (1968). A plea for excuses. In A. R. White (Eds.), Philosophy of action (pp. 18-42). Oxford: Oxford University Press.

Baars, B. J., & Fehling, M. (1992). Consciousness is associated with central as well as distributed processes. Behavioral and Brain Sciences, 15, 203-204.

Bandura, A. (1989). Self-regulation of motivation and action systems through internal standards and goal systems. In L. A. Pervin (Eds.), Goal concepts in personality and social psychology (pp. 19-85). Hillsdale, NJ: Lawrence Erlbaum Associates.

Barkley, R. A. (1988). Attention deficit disorder with hyperactivity. In E. J. Mash & L. G. Terdal (Eds.), Behavioral assessment of childhood disorders (pp. 69-104). New York: Guilford Press.

Barlow, D. H. (1988). Anxiety and its disorders: The nature and treatment of anxiety and panick. New York: Guilford Press.

Bates, J., Loyall, A. B., & Reilly, W. S. (1991). Broad agents. Sigart Bulletin, 2(4), 38-40.

Beaudoin, L. (1993). La motivation est construite, mais les émotions sont fortuites [Emotion as an emergent property]. In Annales de l'ACFAS, 61 (pp. 314). Rimouski, Québec: Université du Québec à Rimouski and Association canadienne-française pour l'avancement des sciences.

Beaudoin, L., & Sloman, A. (1991). A proposal for a study of motive processing (Cognitive Science Research Paper No. 91-6). Cognitive Science Research Centre, School of Computer Science, University of Birmingham, Birmingham, UK, B15 2TT.

Beaudoin, L., & Sloman, A. (1993). A study of motive processing and attention. In A. Sloman, D. Hogg, G. Humphreys, A. Ramsay, & D. Partridge (Ed.), Prospects for Artificial Intelligence (Proceedings AISB-93), (pp. 229-238). Birmingham: IOS Press.

Beaudoin, L. P. (1991). Requirements, domain, and design for a program exhibiting features of intelligent motivation. In R. Dallaway, T. DelSoldato, & L. Moser (Eds.), The Fourth White House Papers (pp. 18-23). (CSRP No. 200) School of Cognitive and Computing Sciences, University of Sussex, Brighton, UK, BN1 9QH.

Beck, H. (1992). An overview of AI scheduling in the UK (AIAI-TR No. 117). Artificial Intelligence Applications Institute, University of Edinburgh.

Bindra, D. (1959). Motivation, a systematic reinterpretation. New York: Ronald Press.

Bindra, D. (1978). How adaptive behavior is produced: a perceptual-motivational alternative to response-reinforcement. The Behavioral and Brain Sciences, 1, 41-91.

Bisiani, R., & Forin, A. (1989). Parallelization of blackboard architectures and the Agora system. In V. Jagannathan, R. Dodhiawal, & L. S. Baum (Eds.), Blackboard architectures and applications (pp. 137-152). New-York: Academic Press.

Bobrow, D. G., & Winograd, T. (1985). An overview of KRL, a knowledge representation language. In R. J. Brachman & H. J. Levesque (Eds.), Readings in knowledge representation (pp. 263-286). Los Altos, CA: Morgan Kaufmann Publishers.

Boddy, M., & Kanazawa, K. (1990). Controlling decision-theoretic inference. In Proceedings of the AAAI Spring Symposium on Planning in Uncertain, Unpredictable, or Changing Environments, .

Boden, M. (1972). Purposive explanation in psychology. Cambridge, MA: Harvard University Press.

Boden, M. (1987). Artificial intelligence and natural man (2 ed.). Cambridge, MA: MIT Press.

Boden, M. A. (1988). Computer models of mind—Computational approaches in theoretical psychology. Cambridge: Cambridge University Press.

Booker, L. B., Goldberg, D. E., & Holland, J. H. (1990). Classifer systems and genetic algorithms. In J. G. Carbonell (Eds.), Machine learning: Paradigm and methods (pp. 235-282). Cambridge, Massachusetts: MIT Press.

Brachman, R. J., & Levesque, H. J. (Ed.). (1985). Readings in knowledge representation. Los Altos, CA: Morgan Kaufmann Publishers.

Bratman, M. E. (1987). Intentions, plans, and practical reason. Cambridge, MA: Harvard University Press.

Bratman, M. E. (1990). What is intention? In P. R. Cohen, J. Morgan, & M. E. Pollack (Eds.), Intentions in communication (pp. 15-31). Cambridge, MA: MIT Press.

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

Brooks, R. (1986a). A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, 2(1).

Brooks, R. A. (1986b). Achieving Artificial Intelligence through building robots (AI Memo No. 899). MIT AI lab.

Brooks, R. A. (1990). Elephants don't play chess. Robotics and Autonomous Systems, 6, 3-15.

Brooks, R. A. (1991a). Intelligence without reason. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, 1 (pp. 569-595). Sydney, Australia:

Brooks, R. A. (1991b). Intelligence without representation. Artificial Intelligence, 47, 139-160.

Brown, A. (1987). Metacognition, executive control, self-regulation, and other more mysterious mechanisms. In F. E. Weinert & R. H. Kluwe (Eds.), Metacognition, motivation, and understanding (pp. 65-116). Hillsdale, NJ: Lawrence Erlbaum Associates.

Bullock, M., Gelman, R., & Baillargeon, R. (1982). The development of causal reasoning. In W. Friedman (Ed.), The Developmental Psychology of Time (pp. 209-254). New-York: Academic Press.

Cawsey, A., Galliers, J., Logan, B., Reece, S., & Jones, K. S. (1993). Revising beliefs and intentions: a unified framework for agent interaction. In A. Sloman, D. Hogg, G. Humphreys, A. Ramsay, & D. Partridge (Eds.), Prospects for Artificial Intelligence (Proceedings AISB-93), (pp. 130-139). Birmingham: IOS Press.

Chapman, D. (1987). Planning for conjunctive goals. Artificial Intelligence, 32, 333-337.

Christ, G. (1991). Toward a model of attention and cognition using a parallel distributed approach part 1: Backround. The Journal of Mind and Behavior, 12(2), 247-262.

Clark, A. (1989). Microcognition: philosophy, cognitive science, and parallel distributed processing. Cambridge, MA: MIT Press.

Cohen, J. D., Dunbar, K., & McClelland, J. L. (1990). On the control of automatic processes: a parallel distributed-processing account of the stroop effect. Psychological Review, 97(3), 332-361.

Cohen, L. J. (1981). Can human irrationality be experimentally demonstrated. The Behavioral and Brain Sciences, 4, 317-370.

Cohen, P. R. (1985). Heuristic reasoning about uncertainty: An Artificial Intelligence approach. Boston: Pitman Publishing Ltd.

Cohen, P. R., & Feigenbaum, E. A. (Ed.). (1982). The handbook of Artificial Intelligence. Los Altos, CA: William Kaufmann, Inc.

Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. Artificial Intelligence, 42(3), 213-261.

Cooper, R., Farringdon, J., Fox, J., & Shallice, T. (1992, Autumn). New techniques for computational modelling. AISB Quarterly, pp. 21-25.

Copi, I. M. (1986). Introduction to logic (7 ed.). New York: Macmillan Publishing Company.

Corkill, D. (9 Jun 1993). Re: Blackboard systems in C. In comp.ai (Internet news group). No: 4053.

Corkill, D. D. (1989). Design alternatives for parallel and distributed blackboard systems. In V. Jagannathan, R. Dodhiawal, & L. S. Baum (Eds.), Blackboard architectures and applications (pp. 99-136). New-York: Academic Press.

Dawkins, R. (1989). The selfish gene (2 ed.). Oxford: OUP.

Dawkins, R. (1991). The blind watchmaker (2 ed.). New York: Penguin Books.

Dean, T., & Boddy, M. (1988). An analysis of time-dependent planning. In Proceedings of the Seventh National Conference on Artificial Intelligence, (pp. 49-54). St. Paul, MN: AAAI.

Dean, T. L., & Wellman, M. P. (1991). Planning and control. San Mateo, CA: Morgan Kaufmann Publishers Inc.

Decker, K., Garvey, A., Humphrey, M., & Lesser, V. (1991). Effects of parallelism on blackboard system scheduling. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, 1 (pp. 15-21). Sydney, Australia.

Dennett, D. (1978). Brainstorms: Philosophical essays on mind and psychology. Montgomery, VT: Bradford Books.

Dennett, D. (1987). The intentional stance. Cambridge, MA: MIT Press.

Dennett, D. (1988). Science, philosophy, and interpretation (Author's Response). Behavioral and Brain Sciences, 11(3), 535-545.

Dennett, D. C. (1978). Why you can't make a computer that feels pain. Synthese, 38, 415-456.

Dennett, D. C., & Kinsbourne, M. (1992). Time and the observer: The where and when of consciousness. Behavioral and Brain Sciences, 15, 183-247.

Desimone, R., & Hollidge, T. (1990). Case studies in fleet operation modelling: An application of AI scheduling techniques (AIAI-TR No. 78). Artificial Intelligence Application Institute, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, UK.

Donner, M. D., & Jameson, D. H. (1986). A real time juggling robot (Computer Science No. RC122111(54549)). IBM.

Doyle, J. (1979). A truth maintenance system. Artificial Intelligence, 12, 231-272.

Doyle, J. (1989). Reasoning, representation, and rational self-government. In Z. W. Ras (Ed.), Methodologies for intelligent systems (pp. 367-380). New York: Elsevier Science Publishing.

Doyle, R. J. (1990). Creating and using causal models. In W. Horn (Ed.), Causal AI models: Steps towards applications (pp. 59-82). London: Hemisphere Publishing Corporation.

Emmelkamp, P. M. (1987). Obsessive-compulsive disorders. In L. Michelson & L. M. Ascher (Eds.), Anxiety and stress disorders—Cognitive behavioral assessment and treatment (pp. 311-329). New York: Guildford Press.

Emmons, R. A. (1989). The personal striving approach to personality. In L. A. Pervin (Ed.), Goal concepts in personality and social psychology (pp. 87-136). Hillsdale, NJ: Lawrence Erlbaum Associates.

Erdelyi, M. H., & Goldberg, B. (1979). Let's not sweep repression under the rug: toward a cognitive psychology of repression. In J. F. Kihlstrom & F. J. Evans (Eds.), Functional disorders of memory (pp. 355-402). Hillsdale, NJ: Lawrence Erlbaum Associates.

Erdleyi, M. H. (1990). Repression, reconstruction, and defence: History and integration of the psychoanalytic and experimental frameworks. In J. L. Singer (Ed.), Repression and dissociation: Implications for personality theory, psychopathology and health (pp. 1-31). Chicago: University of Chicago Press.

Feldman, J. A., & Sproull, R. F. (1977). Decision theory and Artificial Intelligence: The hungry monkey. Cognitive Science, 1, 158-192.

Fikes, R. E., & Nilsson., N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence, 2, 189-208.

Fox, M. S., & Smith, S. F. (1984). ISIS—a knowledge-based system for factory scheduling. Expert Systems, 1(1), 25-49.

Fox, M. S., Allen, B., & Strohm, G. (1981). Job shop scheduling: an investigation in constraint-based reasoning. In Proceedings of the Seventh International Joint Conference on Artificial Intelligence. Vancouver, British Columbia: IJCAI.

French, S. (1986). Decision theory: an introduction to the mathematics of rationality. Chichester: Ellis Horwood.

Frijda, N. H. (1986). The emotions. Cambridge: Cambridge University Press.

Frijda, N. H., & Swagerman, J. (1987). Can computers feel? Theory and design of an emotional system. Cognition and Emotion, 1, 235-257.

Funt, B. V. (1980). Problem-solving with diagrammatic representations. Artificial Intelligence, 13, 201-230.

Gallistel, C. R. (1966). Motivating effects in self-stimulation. Journal of Comparative & Physiological Psychology, 62, 95-101.

Gallistel, C. R. (1983). Self-stimulation. In J. A. Deutsch (Ed.), The physiological basis of memory (pp. 269-343). New-York: Academic Press.

Gardin, F., & Meltzer, B. (1989). Analogical representations of naive physics. Artificial Intelligence, 38, 139-159.

Gelman, R. (1990). First principles organise attention to and learning about relevant data: Number and the animate-inanimate distinction. Cognitive Science, 14, 79-106.

Genesereth, M. R. (1983). An overview of meta-level architecture. In Proceedings of the Third National Conference on Artificial Intelligence, (pp. 119-124). Washington: AAAI.

Georgeff, M. P. (1987). Planning. Annual Review of Computer Science, 2, 359-400.

Georgeff, M. P., & Ingrand, F. F. (1989). Decision-making in an embedded reasoning system. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 2 (pp. 972-978). Detroit, MI: IJCAI.

Georgeff, M. P., & Lansky, A. L. (1986). Procedural Knowledge. Proceedings of the IEEE: Special issue on knowledge representation, 74(10), 1383-1398.

Georgeff, M. P., & Lansky, A. L. (1987). Reactive reasoning and planning. In Proceedings of the Sixth National Conference on Artificial Intelligence, 2 (pp. 677-682). Seattle, WA: AAAI.

Georgeff, M. P., Lansky, A. L., & Schoppers, M. J. (1987). Reasoning and planning in dynamic domains: An experiment with a mobile robot (Technical Note No. 380). Artificial Intelligence Center, SRI Internatinal, Menlo Park, CA.

Gibson, J. J. (1979). The ecological approach to visual perception. Boston: Houghton Mifflin Company.

Gomes, C. P., & Beck, H. (1992). Synchronous and asynchronous factory scheduling (AIAI-TR No. 119). Artificial Intelligence Applications Insitute, University of Edinburgh.

Good, I. J. (1971a). Comments on Herman Rubin's "Occam's razor needs new blades". In V. P. Godambe & D. A. Sprott (Eds.), Foundations of statistical inference (pp. 375). Toronto: Holt, Rinehart, and Winston.

Good, I. J. (1971b). The probabilistic explication of information, evidence, and utility (and twenty-seven principles of rationality). In V. P. Godambe & D. A. Sprott (Eds.), Foundations of statistical inference (pp. 108-141). Toronto: Holt, Rinehart, and Winston.

Goodwin, R., & Simmons, R. (1992). Rational handling of multiple goals for mobile robots. In J. Hendler (Ed.), Artificial intelligence planning systems, (pp. 70-77). College Park, Maryland: Morgan Kaufmann Publishers.

Gordon, R. M. (1987). The structure of emotions: Investigations in cognitive philosophy. Cambridge: Cambridge University Press.

Green, C. D. (1994). Cognitivism: whose party is it anyway? Canadian Psychology, 35(1), 112-123.

Guttenplan, S. (June 4, 1991). Rationality and preference. Seminar given within the context of the Cognitive Science Seminar Series, University of Sussex, Brighton, UK, BN1 9QH.

Haddawy, P., & Hanks, S. (1990). Issues in decision-theoretic planning: Symbolic goals and numeric utilities. In Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control. San Diego, CA: DARPA.

Haddawy, P., & Hanks, S. (1992). Representations for decision-theoretic planning: Utility functions for deadline goals. In Proceedings of the Third International Conference on Knowledge Representation and Reasoning, KR92. Boston.

Haddawy, P., & Hanks, S. (1993). Utility models for goal-directed decision-theoretic planners (Technical Report No. 93-06-04). Department of Computer Science, University of Washington, Seattle, WA 98195.

Hammond, K. (1989). Opportunistic memory. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1 (pp. 504-510). Detroit, MI: IJCAI.

Hanks, S., & McDermott, D. (1993). Modeling a dynamic and uncertain world I: Symbolic and probabilistic reasoning about change (Technical Report No. 93-06-10). Department of Computer Science, University of Washington, Seattle, WA 98195.

Hayes-Roth, B. (1985). A blackboard architecture for control. Artificial Intelligence, 26(3), 251-321.

Hayes-Roth, B. (1990). Architectural foundations for real-time performance in intelligent agents. Journal of Real-Time Systems, 2, 99-125.

Hayes-Roth, B. (1992). Opportunistic control of action in intelligent agents (KSL Report No. 92-32). Knowledge Systems Laboratory, Department of Computer Science, Standford University.

Hayes-Roth, B. (1993). An architecture for adaptive intelligent systems (KSL Report No. 93-19). Knowledge Systems Laboratory, Department of Computer Science, Standford University.

Hayes-Roth, B., & Hayes-Roth, F. (1979). A cognitive model of planning. Cognitive Science, 3(4), 275-310.

Hayes-Roth, B., Lalanda, P., Morignot, P., Pfleger, K., & Balabanovic, M. (1993). Plans and behavior in intelligent agents (KSL Report No. 93-43). Knowledge Systems Laboratory, Department of Computer Science, Standford University.

Hayes-Roth, B., Washington, R., Ash, D., Hewett, R., Collinot, A., Vina, A., & Seiver, A. (1992). Guardian: a prototype intelligent agent for intensive-care monitoring. Artificial Intelligence in Medicine, 4, 165-185.

Hayes-Roth, F. (1987). Rule-based systems. In S. C. Shapiro (Ed.), The encyclopedia of Artificial Intelligence (pp. 963-973). New York: Wiley-Interscience Publications.

Heckhausen, H., & Kuhl, J. (1985). From wishes to action: The dead ends and short cuts on the long way to action. In M. Frese & J. Sabini (Eds.), Goal directed behavior: The concept of action in psychology (pp. 134-159). London: Lawrence Erlbaum Associates.

Herman, P., & Polivy, J. (1991). Fat is a psychological issue. New scientist, 16, 41-45.

Hertzberg, J., & Horz, A. (1989). Towards a theory of conflict detection and resolution in nonlinear plans. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 2 (pp. 937-942). Detroit, MI: IJCAI.

Heuer, H., & Sanders, A. F. (Ed.). (1987). Perspectives on selection and action. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hollenbeck, J. R., & Klein, H. J. (1987). Goal commitment and the goal-setting process: Problems, prospects and proposals for future research. Applied Psychology, 72(2), 212-220.

Holyoak, K. J., Koh, K., & Nisbett, R. E. (1989). A theory of conditioning: Inductive learning within rule based default hierarchies. Psychological Review, 96(2), 315-340.

Horvitz, E. J. (1987). Reasoning about beliefs and actions under computational resource constraints. In L. N. Kanal, T. S. Levitt, & J. F. Lemmer (Ed.), Proceedings of the Third AAAI Workshop on Uncertainty in Artificial Intelligence, 8 . Seattle, WA: Elsevier Science Publishers.

Horvitz, E. J., Breese, J. S., & Henrion, M. (1988). Decision Theory in Expert Systems and Artificial Intelligence. International Journal of Approximate Reasoning, 2, 247-302.

Hume, D. (1977/1777). An inquiry concerning human understanding. Indianapolis: Hackett Publishing Company.

Jagannathan, V., Dodhiawala, R., & Baum, L. S. (Ed.). (1989). Blackboard architectures and applications. New-York: Academic Press.

Johnson-Laird, P. N. (1988). The computer and the mind. Cambridge, MA: Harvard University Press.

Kagan, J. (1972). Motives and development. Journal of Personality and Social Psychology, 22(1), 51-66.

Kanfer, F. H., & Stevenson, M. K. (1985). The effects of self-regulation on concurrent cognitive processing. Cognitive therapy and research, 9(6), 667-684.

Kant, E. (1787/1987). Critique de la raison pure [Critique of pure reason] (J. Barni, Trans.). (2 ed.). Paris: Flammarion.

Keppel, G. (1982). Design and analysis: A researcher's handbook (2 ed.). Englewood Cliffs, NJ: Prentice-Hall.

Koslowski, B., Okagaki, L., Lorenz, C., & Umbach, D. (1989). When covariation is not enough: The role of causal mechanism, sampling method, and sample size in causal reasoning. Child Development, 60, 1316-1327.

Kozak, M. J., Foa, E. B., & McCarthy, P. R. (1988). Obsessive-compulsive disorder. In C. G. Last & M. Hersen (Eds.), Handbook of anxiety disorders (pp. 87-108). New York: Pergamon Press.

Kuhl, J. (1982). The expectancy-value approach within the theory of social motivation: Elaborations, extensions, critiques. In N. Feather (Ed.), Expectations and actions: Expectancy value models in psychology Hillsdale, NJ: Lawrence Erlbaum Associates.

Kuhl, J. (1986). Motivation and information processing: A new look at decision making, dynamic change, and action control. In R. M. Sorrentino & E. T. Higgins (Eds.), Handbook of motivation and cognition: Foundations of social behavior New York: Guilford Press.

Kuhl, J. (1992). A theory of self-regulation: Action versus state orientation, self-discrimination, and some applications. Applied Psychology: An International Review, 41(2), 97-129.

Kuhl, J., & Kazen-Saad, M. (1988). A motivational approach to volition: Activation and de-activation of memory representations related to uncompleted intentions. In V. Hamilton, J. Bower, & N. H. Frijda (Eds.), Cognitive perspectives on emotion and motivation (pp. 63-85). Dordrecht: Kluwer.

Kuhl, J., & Kraska, K. (1989). Self-regulation and metamotivation: Computational mechanisms, development, and assessment. In R. Kanfer, P. L. Ackerman, & R. Cudek (Eds.), Abilities, motivation, and methodology: The Minnesota Symposium on Individual Differences. (pp. 343-374). Hillsdale, NJ: Lawrence Erlbaum Associates Inc.

Laffey, T. J., Cox, P. A., Schmidt, J. L., Kao, S. M., & Read, J. Y. (1988). Real-time knowledge-based systems. AI Magazine, 9(1), 27-45.

Lamontagne, C. (1987). Sensorymotor emergence: Proposing a computational "syntax". In W. Callebaut & R. Pinxten (Eds.), Evolutionary epistemology: A multiparidigm programme (pp. 283-310). Boston, MA: Reidel Publishing.

Lee, D. N., & Lishman, J. R. (1975). Visual proprioceptive control of stance. Journal of Human Movement Studies, 1, 87-95.

Lee, T. W., Locke, E. A., & Latham, G. P. (1989). Goal setting theory and job performance. In L. A. Pervin (Ed.), Goal concepts in personality and social psychology (pp. 291-326). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lesser, V. R., Corkill, D. D., Whitehair, R. C., & Hernandez, J. A. (1989). Focus of control through goal relationships. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 2 (pp. 497-503). Detroit, MI: IJCAI.

Lister, A. M., & Eager, R. D. (1988). Fundamentals of operating systems (4 ed.). London: Macmillan.

Loewenstein, G. F., & Prelec, D. (1993). Preferences for sequences of outcomes. Psychological Review, 100(1), 91-108.

Logan, G. D. (1989). Automaticity and cognitive control. In J. S. Uleman & J. A. Bargh (Eds.), Unintended thought (pp. 52-74). New York: Guilford Press.

Maes, P. (Ed.). (1990a). Designing autonomous agents: Theory and practice from biology to engineering and back. Amsterdam: Elsevier Science Publishers.

Maes, P. (1990b). Situated agents can have goals. In P. Maes (Ed.), Designing autonomous agents: Theory and practice from biology to engineering and back (pp. 49-70). Amsterdam: Elsevier Science Publishers.

Mahoney, M. J. (1991). Human change processes: The scientific foundations of psychotherapy. New York: Basic Books.

Mandler, G. (1980). Emotions: A cognitive-phenomenological analysis. In R. Plutchik & H. Kellerman (Eds.), Theories of emotion (pp. 219-243). San Diego, CA: Academic Press.

Marr, D. (1982). Vision. New York: W. H. Freeman & Company.

McCulloch, W. S. (1945). A Heterarchy of values determined by the topology of nervous nets. Bulletin of mathematical biophysics, 9, 89-93.

McDougall, W. (1936). An introduction to social psychology (23rd ed.). London: Methuen.

Melzack, R., & Wall, P. D. (1988). The challenge of pain. London: Penguin Books.

Meyer, D. E., & Schvaneveldt, R. W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. Journal of Experimental Psychology, 90, 227-234.

Miller, P. H. (1985). Metacognition and attention. In D. L. Forrest-Pressley, G. E. MacKinnon, & T. G. Waller (Eds.), Metacognition, cognition, and human performance: Instructional practices (pp. 181-221). Toronto: Academic Press.

Minsky, M. (1986). The society of mind.

Mischel, W. (1974). Processes in delay of gratification. In L. Berkowitz (Ed.), Advances in experimental social psychology (pp. 249-292). New York: Academic Press.

Mischel, W., Ebbesen, E. B., & Zeiss, A. R. (1972). Cognitive and attentional mechanisms in delay of gratification. Journal of Personality and Social Psychology, 21(2), 204-218.

Nii, H. P. (1986a, Summer). Blackboard systems: (Part one) the blackboard model of problem solving and the evolution of blackboard architectures. AI Magazine, pp. 38-53.

Nii, H. P. (1986b, August). Blackboard systems: (Part two) blackboard application systems, blackboard systems from a knowledge engineering perspective. AI Magazine, pp. 82-106.

Norman, D. A. (1981). Categorization of action slips. Psychological Review, 88(1), 1-15.

Norman, D. A., & Shallice, T. (1986). Attention to action. In R. J. Davidson, G. E. Schwartz, & D. Shapiro (Eds.), Consciousness and self-regulation: Advances in theory and research. (pp. 1-18). New-York: Plenum Press.

Oatley, K. (1992). Best laid schemes: The psychology of emotions. Cambridge: Cambridge University Press.

Oatley, K., & Johnson-Laird, P. N. (1987). Towards a cognitive theory of emotions. Cognition and Emotion, 1, 29-50.

Oatley, K., & Johnson-Laird, P. N. (to appear). The communicative theory of emotions: Empirical tests, mental models, and implications for social interaction. In L. Martin & A. Tesser (Eds.), Goals and affect Hillsdale, NJ: Lawrence Erlbaum Associates.

Ölander, F. (1975). Search behavior in non-simultaneous choice situations: Satisficing or maximising? In D. Wendt & C. Vlek (Eds.), Utility, probability, and human decision making (pp. 297-320). Dordrecht: D. Reidel.

Ortony, A. (1988). Subjective importance and computational models of emotions. In V. Hamilton, G. H. Bower, & N. H. Frijda (Eds.), Cognitive perspectives on emotion and motivation (pp. 321-343). Dordrecht: Kluwer.

Ortony, A., Clore, G. L., & Collins, A. (1988). The cognitive structure of emotions. Cambridge: Cambridge University Press.

Ortony, A., Clore, G. L., & Foss, M. A. (1987). The referential structure of the affective lexicon. Cognitive Science, 11, 341-364.

Pelavin, R. N. (1991). Planning with simultaneous actions and external events. In J. F. Allen, H. A. Kautz, R. N. Pelavin, & J. D. Tenenberg (Eds.), Reasoning about plans (pp. 128-210). San Mateo, CA: Morgan Kaufmann Publishers.

Pennebaker, J. L. (1989). Stream of consciousness and stress: levels of thinking. In J. S. Uleman & J. A. Bargh (Eds.), Unintended thought (pp. 327-350). New York: Guilford Press.

Peters, R. S. (1958). The concept of motivation. London: Routledge.

Peterson, D. R. (1989). Interpersonal goal conflict. In L. A. Pervin (Ed.), Goal concepts in personality and social psychology (pp. 327-361). Hillsdale, NJ: Lawrence Erlbaum Associates.

Pfaffmann, C. (1960). The pleasures of sensation. Psychological Review, 67, 253-269.

Piattelli-Palmarini, M. (1989). Evolution, selection and cognition: From "learning" to parameter setting in biology and the study of language. Cognition, 31(1), 1-44.

Pollack, M. E. (1992). The uses of plans. Artificial Intelligence, 57, 43-68.

Popper, K. R. (1956/1983). Realism and the aim of science. Totowa, NJ: Rowman and Littlefield.

Popper, K. R. (1959). The logic of scientific discovery. New York: Basic Books.

Power, R. (1979). The organisation of purposeful dialogues. Linguistics, 17, 107-152.

Powers, W. T. (1973). Behavior: The control of perception. Chicago: Aldine.

Prosser, P. (1989). A reactive scheduling agent. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 2 (pp. 1005-1009). Detroit, MI: IJCAI.

Pryor, L., & Collins, G. (1991). Information gathering as a planning task. In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society, . Chicago, IL: Lawrence Erlbaum Associates.

Pryor, L., & Collins, G. (1992a). Planning to perceive: A utilitarian approach. In Proceedings of the AAAI 1992 Spring Symposium on Control of Selective Perception.

Pryor, L., & Collins, G. (1992b). Reference Features as guides to reasoning about opportunities. In Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society, . Bloomington, IN: Lawrence Erlbaum Associates.

Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI-Architecture (Technical Note No. 14). Australian Artificial Intelligence Institute, 1 Grattan Street, Carlton, Victoria 3053, Australia.

Reason, J. (1984). Lapses of attention in everyday life. In R. Parasuraman & R. Davies (Eds.), Varieties of attention (pp. 515-549). New York: Academic Press.

Reid, L. D., Hunsicker, J. P., Lindsay, J. L., & Gallistel, C. R. (1973). Incidence and magnitude of the "priming effect" in self-stimulating rats. Comparative and Physiological Psychology, 82(2), 286-293.

Rosenbloom, P. S., Laird, J. E., Newell, A., & McCarl, R. (1991). A preliminary analysis of the Soar architecture as a basis for general intelligence. Artificial Intelligence, 47, 289-325.

Russell, S., & Wefald, E. (1991). Do the right thing: Studies in limited rationality. Cambridge, MA: MIT Press.

Russell, S. J., & Zilberstein, S. (1991). Composing real-time systems. In Proceedings of the 12th International Joint Conference on Artificial Intelligence, (pp. 212-217). Sydney, Australia: IJCAI.

Ryle, G. (1954). Dilemmas. Cambridge: Cambridge University Press.

Ryle, G. (1956). The concept of mind. London: Penguin Books.

Sartre, J.-P. (1947). Existentialism (B. Frechtman, Trans.). New York: Philosophical Library.

Scheier, M. F., & Carver, C. s. (1982). Cognition, affect, and self-regulation. In M. S. Clark & S. T. Fiske (Eds.), Affect and cognition: The seventeeth annual carnegie symposium on cognition. (pp. 157-183). Hillsdale, NJ: Lawrence Erlbaum Associates.

Schneider, W., Dumais, S. T., & Shiffrin, R. M. (1984). Automatic and control processing and attention. In R. Parasuraman & R. Davies (Eds.), Varieties of attention (pp. 1-27). New York: Academic Pres.

Schoppers, M. J. (1987). Universal plans for reactive robots in unpredictable environments. In Tenth International Joint Conference on Artificial Intelligence, (pp. 1039-1047). Milan:

Shallice, T. (1988). From neuropsychology to mental structures. Cambridge: Cambridge University Press.

Shallice, T., & Burgess, P. W. (1991). Deficits in strategy application following frontal lobe damage in man. Brain, 114, 727-741.

Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. Psychological Review, 84(2), 127-190.

Shultz, T. R. (1982). Rules of causal attribution. Monographs of the society for research in child development, 47(1, Serial No. 194).

Shultz, T. R., Fischer, G. W., Pratt, C. C., & Rulf, S. (1986). Selection of causal rules. Child Development, 57, 143-152.

Shultz, T. R., & Kestenbaum, N. R. (1985). Causal reasoning in children. In G. J. Whitehurst (Ed.), Annals of Child Development (pp. 195-249). JAI Press.

Simon, H. A. (1959). Theories of decision-making in economics and behavioral science. The American Economic Review, 49(3), 253-283.

Simon, H. A. (1967). Motivational and emotional controls of cognition. Psychological Review, 74, 29-39.

Simon, H. A. (1993). Artificial Intelligence: An experimental science. Pre-publication abstract of a key-note lecture presented at National Conference on Artificial Intelligence, Washington, DC.

Slany, W., Stary, C., & Dorn, J. (1992). Vague data management in production scheduling applied to high-grade steelmaking. In J. Hendler (Ed.), Artificial intelligence planning systems, (pp. 214-221). College Park, Maryland: Morgan Kaufmann Publishers.

Sloman, A. (1978). The computer revolution in philosophy: Philosophy, science, and models of mind. Atlantic Highlands, NJ: Humanities Press.

Sloman, A. (1984). The structure of the space of possible minds. In S. Torrance (Ed.), The mind and the machine: philosophical aspects of Artificial Intelligence Chichester: Ellis Horwood.

Sloman, A. (1985a). Real time multiple-motive expert systems. In M. Merry (Ed.), Expert systems 85 (pp. 1-13). Cambridge: Cambridge University Press.

Sloman, A. (1985b). Why we need many knowledge representation formalisms. In M. A. Brammer (Ed.), Research and development in expert systems (pp. 163-183). Cambridge, UK: Cambridge University Press.

Sloman, A. (1986). Robot nursemaid scenario (draft project proposal). (Unpublished manuscript).

Sloman, A. (1987). Motives, mechanisms and emotions. Cognition and Emotion, 1, 217-234.

Sloman, A. (1988). Why philosophers should be designers. Behavioral and Brain Sciences, 11(3), 529-530.

Sloman, A. (1989). On designing a visual system: towards a Gibsonian computational model of vision. Journal of Experimental and Theoretical AI, 1(4), 289-357.

Sloman, A. (Mar 30, 1992a). (Personal communication).

Sloman, A. (1992b). Prolegomena to a theory of communication and affect. In A. Ortony, J. Slack, & O. Stock (Eds.), Communication from an Artificial Intelligence Perspective: Theoretical and Applied Issues (Proceedings of the 1990 NATO Advanced Research Workshop on "Computational theories of communication and their applications: Problems and prospects") (pp. 229-260). Heidelberg, Germany: Springer.

Sloman, A. (1993a). Introduction: Prospects for AI as the general science of intelligence. In A. Sloman, D. Hogg, G. Humphreys, A. Ramsay, & D. Partridge (Ed.), Prospects for Artificial Intelligence (Proceedings AISB-93), . Birmingham: IOS.

Sloman, A. (1993b). The mind as a control system. In C. Hookway & D. Peterson (Eds.), Philosophy and the Cognitive Sciences (pp. 69-110). Cambridge: Cambridge University Press.

Sloman, A. (4 Jan, 1994a). (Personal communication).

Sloman, A. (Apr 6, 1994b). (Personal communication).

Sloman, A. (1994c). Explorations in design space. In A. G. Cohn (Ed.), Proceedings of ECAI94, (pp. 578-582). Amsterdam: John Wiley.

Sloman, A., & Croucher, M. (1981). Why robots will have emotions. In Proceedings of the Seventh International Joint Conference on Artificial Intelligence, (pp. 197-202). Vancouver:

Smyth, M. M., Morris, P. E., Levy, P., & Ellis, A. W. (1987). Cognition in action. London: Lawrence Erlbaum Associates.

Stellar, J. R., & Stellar, E. (1986). The Neurobiology of Motivation and Reward. New York: Springer-Verlag.

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. Journal of Experimental Psychology, 18, 643-662.

Sussex University. (1993). Poplog [Computer Program]. Brighton: Integral Solutions Ltd.

Sussman, G. J. (1975). A computer model of skill acquisition. New York: American Elsevier.

Swagerman, J. (1987) The Artificial Concern REalisation System ACRES: A computer model of emotion. Doctoral dissertation, University of Amsterdam.

Tait, R., & Silver, R. C. (1989). Coming to terms with major negative life events. In J. S. Uleman & J. A. Bargh (Eds.), Unintended thought (pp. 351-382). New York: Guilford Press.

Taylor, D. W. (1960). Toward an information processing theory of motivation. In M. R. Jones (Ed.), Nebraska symposium on motivation, (pp. 51-79). Lincoln: University of Nebraska Press.

Teasdale, J. D. (1974). Obsessional states. In H. R. Beech (Ed.), Learning models of obsessional-compulsive disorder London: Methuen.

Toates, F. (1990). Obsessional thoughts and behaviour. London: Thorsons.

Toates, F. M. (1986). Motivational systems. Cambridge: Cambridge University Press.

Toates, F. M. (1988). Motivation and emotion from a biological perspective. In V. Hamilton, G. H. Bower, & N. H. Frijda (Eds.), Cognitive perspectives on emotion and motivation Dordrecht: Kluwer.

Toda, M. (1962). The design of a fungus-eater: a model of human behavior in an unsophisticated environment. Behavioral Science, 7, 164-182.

Tomkins, S. S., & Messick, S. (Eds.). (1963). Computer simulation of personality. New York: Wiley.

Toomey, C. N. (1992). When goals aren't good enough. In J. Hendler (Ed.), Artificial intelligence planning systems, (pp. 311-312). College Park, Maryland: Morgan Kaufmann Publishers.

Trigg, R. (1970). Pain and emotion. Oxford: Clarendon Press.

Tversky, A. (1969). Intransitivity of preference. Psychological Review, 76(1), 31-48.

Tversky, A., & Kahneman, D. (1983). Extensional versus intuitive reasoning: The conjunction fallacy in probability judgement. Psychological Review, 90(4), 293-315.

Uleman, J. S., & Bargh, J. A. (Ed.). (1989). Unintended thought. New York: Guilford Press.

Vere, S. A. (1983). Planning in time: Windows and durations for activities and goals. IEEE PAMI, 5(3), 246-267.

Warnock, G. J. (1989). J. L. Austin. London: Routledge.

Wegner, D. M., & Schneider, D. J. (1989). Mental control: the war of the ghosts in the machine. In J. S. Uleman & J. A. Bargh (Eds.), Unintended thought (pp. 287-305). New York: Guilford Press.

Wegner, D. M., Schneider, D. J., Knutson, B., & McMahon, S. R. (1991). Polluting the stream of consciousness: the effect of thought suppression on the mind's environment. Cognitive Therapy and Research, 15(2), 141-152.

Weir, S. (1978). The perception of motion: Michotte revisited. Perception, 7, 247-260.

White, A. R. (1964). Attention. Oxford: Basil Blackwell.

White, P. A. (1989). A theory of causal processing. British Journal of Psychology, 80, 431-454.

White, R. W. (1959). Motivation reconsidered: The concept of competence. Psychological Review, 66(5), 297-333.

Wilensky, R. (1980). Meta-planning: Representing and using knowledge about planning in problem solving and natural language understanding (No. UCB/ERL/M80/33 Berkeley electronic research laboratory.

Wilensky, R. (1983). Planning and understanding: A computational approach to human reasoning. Reading, MA: Addison-Wesley.

Wilensky, R. (1990). A model for planning in complex situations. In J. Allen, J. Hendler, & A. Tate (Eds.), Readings in Planning (pp. 262-274). San Mateo, CA: Morgan Kaufmann. (Reprinted from Cognition and Brain Theory, 1981, 4(4)).

Wilkins, D. E. (1985). Recovering from execution errors in SIPE. Computational Intelligence(1), 33-45.

Wilkins, D. E. (1988). Practical planning: Extending the classical AI planning paradigm. San Mateo, CA: Morgan Kaufmann.

Winterfeldt, D. v., & Fischer, G. W. (1975). Multi-attribute utility theory: Models and assessment procedures. In D. Wendt & C. Vlek (Eds.), Utility, probability, and human decision making (pp. 47-85). Dordrecht: D. Reidel.

Zilberstein, S., & Russell, S. J. (1992). Efficient resource-bounded reasoning in AT-RALPH. In J. Hendler (Ed.), Artificial Intelligence planning systems, (pp. 260-266). College Park, Maryland: Morgan Kaufmann Publishers.