Control States and Complete Agent Architectures

Dr. Darryl N. Davis

Neural, Agent and Emergent Technologies Research Group

Department of Computer Science

The University of Hull

Kingston upon Hull

HU6 7RX

UK

Email: D.N.Davis@dcs.hull.ac.uk

Web: http://www2.dcs.hull.ac.uk/NEAT/index.html

Tel: +44 (0)1482 466469

Fax: +44 (0)1482 266666

Abstract

This paper presents a developing concept of mind defined in terms of external and internal niches. This perspective on mind is described primarily in terms of the niche space of control states and the design space of processes that may support such phenomena. A developing agent architecture, that can support motivation and other control states associated with mind, is presented. Different aspects of agent research are discussed in terms of three categories of agents. Each agent category is characterized primarily in terms of their task-related competencies and internal behaviors and discussed in terms of our taxonomy of control states. The concept of complete agents is then introduced. Goals are described in terms of their generation across a number of computational layers. Experimental analysis is provided on how these differing forms of behaviors can be cleanly integrated. This leads into a discussion on the nature of motivational states and the mechanisms used for making decisions and managing the sometimes-competitive nature of processes internal to a complex agent. The difficulty of evaluating complete agents is discussed from a number of perspectives. The paper concludes by considering future directions related to the computational modeling of emotions and the concept of synthetic mind.

Keywords:

Agent architectures, synthetic mind, control states, design space, niche space.

1. Introduction

This paper reports on an exploration of the different categories of agent architecture for the modeling of motivational and other information-rich control states. This research is based on a consideration of the theoretical, design and computational limitations of a theory of mind (Sloman 1993). In pursuing this research, the conviction is that the strengths, inadequacies and oversights of a theory of (synthetic) mind and its resulting computational designs become apparent through producing plausible implementations. The motivations for research in this area are long term, and the expectation is that of making slow progress. However in the short term, by producing plausible computational models of synthetic agents, we can further our understanding of biological, psychological and social agents, and the theories and models that attempt to describe them. It is open to argument what are the appropriate evaluation criteria for any such endeavor. Hence the success of any such project can be difficult to assess. However, we are finding that the emerging designs provide building blocks suitable for applications requiring intelligent systems, for example decision support systems that build on expert system and agent technology (Davis 1999a; Davis et al 2000) and agents capable of game-playing (Davis et al 1999).

The contention is that a highly interconnected four-layer architecture is required for a complete (cognitive) agent, making use of reflexive, reactive, deliberative and reflective behaviors. A central theme to this argument is that implicit (tropistic) representations and related processing are sufficient only to model certain types of agent behavior. To reason about past events, make predictions about future events and act in a pro-active manner, a persistent (hysteretic) representation of at least some parts of the environment and past behaviors is required. This requires the use of computational processes that can manipulate representational structures. Of particular interest are the architectural structures and mechanisms involved in the production and processing of motivators and similar phenomenological states. These processes do not exist in isolation but are related to an agent's internal and external environments. Reflective processes which monitor processes internal to an agent need to be of a specific type if recursively higher processing strata are to be avoided at the theoretical, design and implementation stages. How these differing processes can be integrated is described in a discussion on the nature of motivational states and the mechanisms used for making decisions in simple but changing and incompletely known environments.

2. Niche Space and Design Space

The nature of this research, and the areas it addresses, is informed through the use of philosophical and psychological perspectives on natural and synthetic agents. One particular perspective, that of niche space and design space, enables a broad and deep view of the research. Sloman (1998) describes both design space and niche space as discontinuous, multi-level interacting spaces. A niche is not (necessarily) a physical environment but a collection of more or less specific requirements and constraints. For example two organisms may share the same physical environments but inhabit different niches, for instance predator and prey scenarios. Both classes of agents exist in the same environment. Requirements for navigating an uninhabited environment are effectively the same. The drives that compel these agents to explore different parts of an inhabited environment differ. Prey agents direct their behavior to flee predators and seek vegetation. Predator agents seek both vegetation and prey agents. Hence, the niche spaces for these two agents overlap but differ. The study of niche space provides a means to facilitate the investigation of collections of requirements and constraints for the design of systems for specific niche spaces. Design space provides a means for the study and analysis of alternative and competing design possibilities for niche spaces at different levels of abstraction. Design space is a collection of more or less specific methodologies, formalisms, architectures, mechanisms, algorithms and (virtual) machines. Mappings between these two spaces represent more or less well-fitted matches between requirements and designs. Any specific agent design fulfils the requirements and constraints of one specific niche point to some degree of efficacy. Furthermore, any specific agent design may function, with greater or lesser efficacy, in more than one niche space. A dragonfly is an effective insect predator. A mayfly fulfils a specific insect niche very effectively but makes an ineffective insect predator. There can be many compromises associated with the mappings between the two spaces and there may be no unique criterion of optimality. Furthermore a novel design for any set of requirements and constraints in niche space may create a change in that niche space.

Figure 1, for example, depicts a relatively straightforward trajectory through an epistemological niche space and an associated computational design space. In these spaces, data is defined as facts from which information may be inferred and information as intelligence given data, for example hypotheses about patterns of data. There can be degrees of belief associated with information. Knowledge is assured belief; i.e. that which is known, and provides the means by which information can be applied to tasks in any specific domain. The structural complexity of niche space increases in moving from data through to knowledge. Here niche space describes the increasing epistemological complexity associated with moving from data through information to knowledge that facilitates decision-making. An associated design space describes the increasing design sophistication of computational systems capable of supporting activities in this niche space. The advent of a new computational system (the data-mining agent) not only changes design space but also affects the niche space(s) if it discovers new knowledge. If this new knowledge requires new types of representations, this will alter niche space in a more dramatic fashion. In turn these changes in niche space will require changes in design space, for example new types of decision support tools. At a more abstract level, hinted at by the broad encompassing arrows in figure 1, the cognitive frameworks used in studying these niche and design evolve as the original trajectory shifts in response the changes affected in these spaces. Of course these cognitive frameworks themselves may, in turn, be described in terms of niche space and design space.

Niche and design spaces provide a means of investigating the theatre of environments that encompasses the range and diversity of time, space, nature and mind. The niche and design space(s) that drive the current work are those that minds (whether natural or synthetic) inhabit. The analysis of niche space enables the production and modification of the theory of mind. By designing agent architectures based on different theories of the mind, a better understanding of the strengths and inadequacies of those theories is achieved. Through such evaluations, a better understanding the niche spaces studied is achieved. Again, as for the simple computational trajectory described above, as we progress with producing tentative designs for cognitive agents, the descriptions we use, in evoking the niche spaces we are studying, change with our understanding of them. It is argumentative, and a matter of philosophical debate, what it is that does change. Do the niche spaces themselves change or merely the means by which we access and think about them, and hence the aspects that we can indeed access. One particular trajectory through the niche and design spaces related to mind is that of control states.

3. Mind and Control States

Mind can be considered to be a dynamic structure of asynchronous data, information and knowledge processing mechanisms and the information-rich control states they support. Control states and their underlying processes directly relate to or reflect the different mental phenomena as discussed in the psychological and philosophical literature. A control state is a behavior, pervasive and encompassing in its nature, internal to an agent. A control state need not necessarily manifest itself as external behavior. Control states can influence activity and may require different modes of processing at different layers in the mind. A control state can be associated with a specific set of cognitive processes, which act as a locus for that control state. This is analogous to the modular approach to describing cognition. However, these loci can also act as explicit and implicit contexts for other processes. Such an effect can cause antithetic processes to stall or cause the emergence of further control states, whether perturbant or harmonious. Control states need not be Boolean or exclusive, and more than one control state can be extant at any particular time. Control states can be temporally brief, even instantaneous, or temporally more expansive. The taxonomy of control states (see figure 2) is discussed in depth elsewhere (Sloman 1990, 1993). This in turn builds on the work of Simon (1967). Here a few of these control states of particular relevance to the current work are described.

Beliefs are internal models of the world, possibly inferred from perceptual acts (e.g. something red, to the left and moving away to the right) or the result of internal processes (e.g. what was the last red thing thought about). Rational beliefs are those that can substantiated through means of deductive processes or empirical evidence. These are the assured beliefs used in the definition of knowledge in the previous section. Beliefs in general however need not have a rational basis, for example a child's belief in the existence of monsters under the bed.

Emotions can be described as "*a state usually caused by an event of importance to the subject. It typically includes (a) a conscious mental state with a recognizable quality of feeling and directed towards some object, (b) a bodily perturbation of some kind, (c) recognizable expressions of the face, tone of voice, and gesture (d) a readiness for certain kinds of action*" (Oatley and Jenkins 1996). In short emotion can be defined as a quality of mind arising from valenced expectations and reactions to events in the world and an agent's perceptions of those events and other related categories of cognitive acts. Many emotions can be related to motivators particularly if parts (a) and (d) of this definition are highlighted.

Imaginings can be characterized as hypothetical expectations or conjectures that make use of other epistemological states to perform problem solving, planning or other forms of reasoning. Imaginings and irrational beliefs can be interrelated through internal processes such as non-rational deliberations. For example, a child can experience imaginings about adventures related to the irrational belief in monsters under the bed.

Impulses are related to spontaneous behavior, for example suddenly leaving the cinema during the screening of a film or making a rash purchase. They are associated with the instantaneous formation of an idea, perhaps unrelated to current cognitive context, and can cause a temporary or more persistent re-focus of mind.

Motivators are dispositions to assess situations and respond to those situations and assessments in a certain way. They provide a context and impetus for reasoning about environmental and epistemological events, and provide the substantive basis for certain types of goal-directed behavior.

Reflexes are survival-related behaviors typically manifested as ballistic mappings from input to output. Jumping out of the way of unexpected obstacles in a dynamic environment typifies these fast behavioral responses to perceptual events. Reflexes can result in disruptive cognitive activity, for example attention being repeatedly or involuntarily drawn to task-irrelevant perceptual or cognitive stimuli.

Many of these control states, for example motivators, can take several sub-forms.

Attitudes are pre-dispositions to respond to specific sensory or cognitive cues in specific ways. For example, an agent could generate pro-active goals to investigate a hapless agent based on an altruistic standard (an attitude) and a set of beliefs about the capabilities of that agent.

Desires underpin goals and other purposeful behavior, whether internal or external. The use of this term is not analogous to its use in BDI agent architectures. Here desires need not be realistic or achievable or describable in logic. Repeated attempts to achieve unrealistic desires, which necessarily result in failure, may lead to perturbant and disruptive control states.

Goals, whether reactive or pro-active, can take a variety of forms. *Quantitative* goals are those types of goals talked about in control theory and reinforcement learning, for example (Maes 1989). *Qualitative* goals are similar to most artificial intelligence goals (as used in for example PRS or described by Nilsson (1994)) and involve relations, predicates, states and behaviors.

A major concern has been with how motivators are generated, activated, reactivated and managed. In this work the term *generactivation* is used**.** Motivator generation refers to the production of a new motivator. Activation refers to the (typically deliberative) processing of a motivator. Generactivation refers to the generation of motivator, if it does not already exist, or to its activation if it does. Motivational states and structures are complex and dynamic in their nature. They can exist across a number of layers in a computational agent, requiring different types of processing at those different layers.

Sloman and colleagues (Beaudoin 1994; Sloman et al 1994) suggest that motivator structures need to include at least 12 components. The current research extends this generic framework, with for example emotional attributes, and is summarized in table 1. Not all components need be instantiated and components can take different values over the lifetime of a motivator. This research addressed how motivators are processed including how new ones attract attention, how their importance is assessed, what sorts of time constraints are in place and how these are assessed and utilized. These investigations have also considered how conflicts between motivators are dealt with, how plans are formed, selected, and executed, and how new events can affect existing motivators (Davis 1996). At a design and implementation levels the management of resource-limited processes has also been investigated. For example how attention is controlled when an agent has too many currently active motivators and the forms of self-monitoring (or self-awareness) required for such activities. The latter, in part, drives the contention that a model of mind requires meta-management or reflective processes.

A particularly contentious area of research (Wright et al 1996) considered how processes involving conflicts and motivation can lead to control states that are at least partly analogous to some of the states described as *emotional* in humans. More recent work (Davis 1999b, 2000, 2001) is looking to deepen the emotion aspect of this computational model of mind, and provide an emotional capacity for motivators. For, example fear can be defined as a physical or social threat to self, or a valued role or goal (Oatley and Jenkins 1996). If a specific motivator to reach a point of safety in an increasingly dangerous environment cannot be achieved, the motivator management processes may re-evaluate this motivator and set its dynamic state from *active* to *unachievable*. As a side effect the alarms associated with the emotional correspondence for the motivator trigger the experience of fear in the agent. This in turn will cause a not necessarily rational re-evaluation of current motivators, and possibly the abandonment of the very motivator that would save the agent.

4. An Architecture for Mind

A primary aim of this research is to develop a theory of synthetic mind. An acceptable theory for a synthetic mind is one that maps onto a computational design and (theoretically at least) onto at least one implementation. It is taken as read that this endeavor is philosophical plausible – see (Chalmers 1996). Through the development of an acceptable theory of synthetic mind, better theories of natural mind can be developed. The distances between niche spaces for natural and synthetic mind are such that the development of designs for synthetic mind will enable the development of theories that will explain many (but not all) aspects of natural mind. While a specific set of designs may be capable of supporting a synthetic mind, there will be qualitative differences between that design and one nearby in design space that is capable of supporting natural mind. One of many reasons for this is the difference in their information processing substrates.

Figure 3 presents a stylized view of a developing architecture for synthetic and natural mind (Sloman 1993; Beaudoin 1994; Davis 1996, 1998); an alternative version is given in figure 6. The figure emphasizes four-layers interacting within an integrated whole. Each level of the architecture, and the processing within and between layers should be viewed as concurrent, albeit asynchronous, activities. The model is quite general, and the effect of altering the relative size and importance of the layers is an open issue. This model differs from for example Newell's (1990) theory in that it presents a broader picture of the mind, with high level (or *cognitive*) and low level (or *conative*) processes co-existing and interacting in a holistic manner. Hence motivator processing, planning, decision-making and other cognitive processes are not merely abstract but exist in relation to other automatic, adaptive and pre-conscious processes. The two lower layers relate to *pre-attentive* (conative) processes and are capable of supporting innate and learnt environmental competencies and behaviors. Perception of and action upon the external environment is mediated primarily through these two layers. The third (deliberative) layer relates to the types of things discussed in most cognitive science, for example (Newell 1990). This does not preclude a non-symbolic implementation of this layer. The fourth layer, the reflective meta-cognitive qualities, serves to monitor the overall behavior of the agent. In particular, the role of the reflective layer is to identify and act on out-of-control behaviors, whether internal, external, deliberative or reactive. This meta-management level processing is considered to be the most abstract level of processing. The computational nature of the fourth layer needs to be reactive, and non-deliberative. If not there is a requirement for the reflective processes to be monitored in turn. This in effect would lead to an infinite regress requiring meta-meta-deliberation to monitor the meta-deliberative processes, and yet further levels monitoring them.

The lowest level, *reflexes,* are very fast, inherently parallel, pre-attentive processes that allow a direct response to perceptions, similar to the mapping from the visual pathway to the limbic and motor systems in many biological systems. However, this does not preclude these processes from provoking or triggering events at other levels. For example, a reflexive process may trigger motivational states that require processes of a qualitatively different nature. The reactive layers extend the reflexive layer and allow for adaptive behavior. This includes the reactive planning activities associated with agent architectures (Kaelbling 1989). Other automatic processes are similarly pre-attentive, but *necessitate* the generactivation of conscious control states at the deliberative and possibly reflective layers to achieve their goals. In ascending the architecture, the processing layers, if they are to correspond to psychological phenomena, become slower and more serial in nature. The deliberative layer represents those processes typically studied in thinking and human problem solving. Other kinds of processes at this level include those used in the management of low level actions, for example the high level behavioral components associated with driving along a busy road in treacherous conditions. Many learnt processes and behaviors come into existence at this level, but are typically subsumed into the lower layers of the architecture. As deliberative processes are resource-limited, they require attention filters to protect them from disturbance and unwanted interruptions. In turn, they require meta-cognitive reflective processes to regulate and direct them. In some scenarios the resource limits restricting parallelism in high level processes may lead to emotional and other characteristically human states involving partial loss of control of attention and thought processes (Sloman 1993). Recent work (Davis 1999b, 2000, 2001) extends this model with affective states. The reflective processes serve to monitor both internal and external behavior. In effect they act as *inner perception* and *inner action* processes. Reflective processes are triggered in response to events occurring elsewhere in the architecture, whether positively or negatively valenced, and serve to (re-)direct the cognitive focus of the agent. This meta-deliberation arises from an interaction of the overall behavior of the agent in its (internal and external) environments, and its designated (or acquired) niche role(s). The processes that have their loci at this level therefore provide a high-level context for the overall processing of the agent.

The attention filter acts as process protector and a facilitator of cognitive focus-of-attention. It can be context loaded and/or respond to signal strength. Some low level actions (or alarms), for example life threatening events requiring fast responses at the pre-attentive level, can effectively override the filter and current deliberative processing purely on the basis of their signal strength and cause a refocus of the majority (if not all) processes across all the layers. A surfaced control state is one that the agent is now conscious of, and may require attention at the deliberative level. In terms of context loading the filter accepts information that is related to ongoing (or stalled) deliberative processes and so allow low-level activity to prompt a possible re-evaluation of these processes. The attention filter's function, however, is not simply to acquiesce to pertinent goals but to also protect the deliberative processes from being overloaded with currently unimportant information pertaining to the agent's environment. In some design and implementation experiments there has been a perceptual component to this filter whereby it acts as a percept block. In other situations, the filter accepts agent-related (external) events to trigger deliberative processes in a manner analogous to the cocktail party phenomena (Cherry 1957). Here the attention filter acts as a contextual filter that deflects or permits the surfacing of control states. A more sophisticated filter will be capable of modeling for example divided and focused attention.

Although this description is presented in a hierarchical way, it does not suggest that motivational (and other control) processes at the reflective layer always override processes at the other layers. The behavior of an intelligent cognitive agent is not controlled by any of these layers in isolation. Behaviors at the reactive level may preclude processes at or actions motivated by the deliberative or reflective layers. Processes over any specific combination of layers may arise as a result of an agent attempting to manage control states originating in any of the layers. Where decision processes related to possibly antagonistic behaviors are not cleanly integrated, there is the very real possibility that the agent will experience cognitive perturbance. When the agent cannot learn from these situations, or only learns slowly, the agent runs the risk of experiencing prolonged dysfunction.

5. Categorizing Agents

The aim of this section is not to provide a review of all research on agent architecture (impossible within the remit of this paper), but to consider exemplars of different types agent architectures according to a categorization that fits with the overall perspective of this paper. The contention, that the computational architecture for a complete (cognitive) agent needs four separate yet highly interrelated layers, necessitates the consideration of a number of ongoing arguments in agent research. One influential but limiting view, given the focus on complete agents, is the argument that classical artificial intelligence type architectures are wholly inappropriate for agents. A classical (typically hierarchical) architecture performs a lot of internal processing of explicitly represented (perceptual) information before any action selection and subsequent behavior is performed. The key idea underpinning this rejection is the relation between intelligence and situatedness and embodiment. In effect this states that intelligence is situated in interactions with the world, and not as a disembodied entity placed in, for example, planners and theorem-provers. This argument, taken to an extreme, would therefore preclude abstract thought, expectations, dreams or other cognitive behavior, of which most humans are capable, as being categorized as intelligent. Genesereth and Nilsson (1987) describe pure (reactive) behavior based agents as tropistic; they keep no representation of past events. Other agents are hysteretic – they keep representations of past events that enable them to reason about current and future events. Intelligent behavior relates to effective and novel use of the results of interactions with internal and external worlds; current and past, real and hypothetical. As such intelligent behavior can be generated in any number of ways, including explicit reasoning of the type that symbolic AI proposes. We do not buy into the exclusiveness of the situatedness argument. The argument that intelligence is, to some extent, an emergent property of certain types of complex systems and their interactions with their external environment and not some innate isolated property is considered when we return to the mapping between an agent's internal and external environments.

It is possible to produce endless definitions of what an agent is, from different perspectives through the use of a seemingly endless list of attributes and qualities (Wooldridge and Jennings 1995; Ferber 1999; Franklin and Graesser 1996). Here four different types of agents are defined using nine attributes or qualities. Simple agent systems, which equate to the notion of weak agency, should demonstrate the following qualities:

autonomy*:* agents operate without direct interaction of other entities, i.e. they have some form of control over their actions and internal states.

flexibility**:** agents adapt and react to changes in their environments. In effect agents should behave appropriately in less than ideally circumscribed circumstances.

reactivity*:* agents perceive their environment, which may be the physical world, an abstract representation of some information domain or a totally synthetic environment, and respond in an expedient manner to changes in that environment.

social ability*:* agents interact in a competitive and/or cooperative fashion.

More intelligent agents may demonstrate further attributes and characteristics, for example:

communication: agents can communicate via some (agent) communication language.

learning: internal capabilities and external competencies are not static but can be modified to the agent's needs through experience.

There are any number of agent designs that may display some or all of these qualities. Comparisons across these different agents are only possible if the research objectives associated with those agent designs are considered. A stronger notion of agency requires us to consider notions such as belief, intention and motivation. Brustoloni's work (Brustoloni 1991; Franklin 1995) presents a hierarchy of agent types based on different kinds of motivational drives. The simplest agent type are regulatory agents capable of no planning but with sufficient capabilities to cope with a limited range of stimuli in their environment. A more sophisticated category (adaptive agents) consists of those agents that are capable of learning new behaviors in pursuit of their drives but remain (even at the design level) situated within a specific domain. Here a different categorization is used for classes two and three. The regulatory class still serves as a base level for agent categorization. Class two agents are those agents, whether planning or adaptive, that are focussed, whether at the design or implementation level, on one specific domain, and typically a sub-set of the possible tasks within that domain. Class three agents are those which are intended to be capable of extending their knowledge and behaviors, whether internal or external, in order to traverse the space of problem spaces within a domain. The intention is not to provide an extensive review of the whole field, but to consider some of the important work in agent architectures from the perspective of control state theory and these categorizations.

5.1. Category 1: Regulatory Agents

A regulatory agent is defined as *an integrated (computational) entity with intentionality and some degree of autonomy*. The behavior based approach, for example (Brooks 1991), is suitable for the design and implementation of such agents. The penguin in Pengi (Agre and Chapman 1987) makes use of a simple horizontal architecture with a number of competing behaviors. Arguably, it demonstrates intentionality - the desire to stay alive and avoid its opponent (the killer-bee). It reacts to changes in its environment, for example it attacks the killer-bee if the environmental situation is suitable, otherwise it flees. This approach to agent design seems to work well for robots (e.g. Herbert) and agents that model insect-like or limited task fulfilling behaviors built from control states such as impulses and reflexes.

An alternative architecture for this category of agent makes use of cellular automata and artificial life principles, for example (Davis et al 1999). This agent, by a means of multiple communities of homogeneous cellular automata, demonstrates reflexive and reactive behaviors appropriate for playing the game of Go. Instead of directly imposing human models of how to play the game of GO, integer-valued Cellular Automata (CA) represent all the available positions on a GO board. These simple agencies communicate to each other about their state and their local environment. By placing a stone on the board, the state of the board as a whole, at a number of levels of abstraction, is changed. This change is a result of the interaction of all the agencies that change state as a result of the stone placement. Part of the communication between agencies can be seen as landscaping the board space. This simple model has been developed to allow communicating communities of automata representing different aspects of the game. This work highlighted the inadequacy of the regulatory agent as a model of cognition. The emergent and beneficial complexity of behavior arising from interacting simply modeled agents can only be utilized in agents whose computational architecture allows them to recognize and utilize these patterns. Overall this category of agent precludes many of the control states, for example beliefs, motivators and imaginings.

5.2. Category 2: Domain Specific Agents

Agents in this category extend the capabilities of the regulatory agents, through the ability to manipulate representations (typically related to plans) or select between behavior sets, in order to achieve goals that persist over time. Kaelbling's reactive planning architecture (Kaelbling 1989) builds upon the principles of modularity and robustness, ensuring perceptual and behavioral levels of competence, and awareness. This type of agent is locked into a continual *perceive-the–environment* cycle, regardless of the agent's capability in assessing the resulting perceptual information. This can cause two major problems for a slow-acting planning system. The agent may not be able to plan fast enough and, secondly, by the time a plan is formulated its context is at odds with the agent's current situation. Kaelbling solves these by resorting to a hybrid architecture that permits hierarchical mediated behaviors. This does address some of the shortfalls of the pure behavior approach and points to the way to designing more robust heterarchical agent architectures. However this tropistic architecture presents an inappropriate model for complete agents which need not only to *react intelligently* in respect to their external environment, but also reason about past, current and possible future events.

A different type of agent within this category is that exemplified by PRS (Georgeff and Lansky 1986). This is a reactive planning class agent architecture that supports attention shifts, changing intentions and the suspension and resumption of external and internal processes. 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. Where novel top-level goals need to be generated at run-time in more complex environments this leads to either an unusable generality of plans or an explosion in the number of specific plans. PRS does not allow the asynchronous generation of new top-level goals, which is a flexible way of making a system reactive to new problems and opportunities. All top-level goals and their associated plans need to be preformed. This is a limitation for partially unknown environments. Although PRS can redirect its activity while pursuing a plan, it must commit itself at the outset to a plan, and cannot defer execution of plan after partial expansion – there are limits to its behavior suspension and resumption capabilities. PRS can support a subset of the control states, in particular beliefs and goals, and attitudes and standards are implicit in the goals and the plans of a PRS implementation. However it is inadequate for non-goal-oriented control states, such as impulses, reflexes and imaginings. PRS makes use of a first order logic which ensures rational beliefs, but also requires that a complete model of the environment is required to produce this. It is unclear how a non-shallow design for a system supporting computational emotions fits into the PRS design philosophy.

With Touring machines, Ferguson (1995) builds upon the ideas expounded by Kaelbling, making use of three independently motivated layers. A reactive layer provides a set of behavioral competencies that allow the agent to cope with a partially structured, dynamic, multi-agent world. Sat above this is a planning layer. This is similar to the Kaelbling architecture discussed above. The third layer provides a modeling layer that enables the agent to construct belief models of its observed environment. This permits the agent to reason about possible future events in its environment – a Touring Machine is a hysteretic agent. Further modules provide a perception sub-system, an action sub-system and a control context. The latter is responsible for activating and deactivating potential behaviors across the three layers using context-activated control rules. This architecture will support many of the control states, including beliefs, imaginings and goals. However, the use of the context-switching module means it is not possible with this architecture for low-level (instinctual/survival) mechanisms to override the control (focus-of-attention) mechanisms.

5.3. Category 3: Problem Space Agents

Problem space agents are those capable of many types of processing at various levels, but most importantly attempt to model more general problem-solving behaviors of use in more than one domain or category of task in one domain. Category 3 agents, at least at the theoretical and design level, are computational or robotic systems that aim to display generalized capabilities. Problem space agents are capable of effective functioning in a number of closely related niche spaces. They need to determine the kind of niche space they inhabit, and be able to make predictions about the effect of actions in their current problem (or niche) space; in short they need to be *pro-active*. Such agents do not simply respond to events but take initiatives upon their (internal and external) environment so that it becomes more conducive to the agent's long-term aims. The behavior-based approach does not allow for persistence of events in and about the world, for example memories of perceptual, episodic, semantic or *cognitive* events that are necessary for pro-active behavior. Nor does it allow for rational thought, for example, the establishing of a set of assured beliefs. Hence tropistic agents are not considered in this section. The Touring Machine architecture discussed above provides a boundary between categories two and three. However, its capabilities need to be generalized to provide an insight into how it will cope with different domains and problem spaces.

The SOAR architecture (Newell 1990) makes use of explicit representations that supports both reactive and run-time planning. As the agents developed within this research community incorporate some learning capabilities, for example chunking, at the minimum it represents a more general exemplar of the class 2 agent and to some extent points the way to the most sophisticated class of agents. The theory underpinning this architecture in attempting to unify perspective on cognition provides a framework for category 3 agents. However, many of the SOAR domain implementations are category two agents, for example the TACAir-SOAR (Tambe et al 1995). The theoretical framework underpinning SOAR assumes all motivators arise at the deliberative level, and can be solved as goals in problem space. It fails to explain how conative behaviors and affective states are incorporated into cognitive acts, and how behaviors at reactive and deliberative levels interact and segue; exactly those capabilities displayed by the Kaelbling and Ferguson agents. It fails to answer the question: why and from where do the different categories of (deliberative) cognitive behavior arise. It struggles to explain the emotive content of many control states, and to provide a substantive basis for many of the control states outlined in figure 2.

A further possibility is the integration of the behavior-based approach with the blackboard approach (Hayes-Roth 1993b, 1995). For example, a number of concurrent behaviors can be given access to sensory data, each (possibly) producing its own potential action. The behaviors can range from very simple, e.g. continue performing the same simple task, to more abstract behaviors, e.g. an agent wondering whether it has been in a similar situation previously, requiring some reasoning about current and past events and how they relate to ongoing and potential internal control states. This type of architecture (for example the Guardian implementation) has demonstrated its capability of managing reactive and timely behaviors in complex problem spaces. Hayes-Roth suggests that peripheral reflexes can be incorporated into the Guardian architecture, along with reactive and deliberative processes. However it remains unclear how conflicts between high and low level behaviors are resolved. Furthermore it is unclear how the control state metaphor maps onto the architecture without compromising much of the design philosophy.

5.4. Complete Agents

Franklin (1998) describes a complete agent as a system structurally coupled to its environment in pursuit of its own agenda. This builds on the embodiment of the behavior-based approach but allows hysteretic designs. Such an agent is a close relative of the category three agents as described above. A criticism of the above architectures is that while they represent exemplars of the three agent categories and demonstrate some or all of the qualities introduced above for agents and intelligent agents. They fall short of the aim in pursuing the design of complete agents. For Sloman (Sloman 1990) a complete agent is a design and computational metaphor that allows the exploration of principles and mechanisms associated with the concept of mind, and is analogous to what Bates and associates (Bates et al 1995) term broad agents. In some of this research, complete agent and cognitive agent are synonymous. This paper will use complete agent so as to reduce confusion with the type of cognitive agent exemplified by the theory behind the SOAR architecture (see Ferber 1999:pp20). Such cognitive agents rely on non-embodied models of cognition, being focussed on memory mechanisms, explicit and symbolic knowledge representation and rational decision-making. The heterarchical (complete) agent architecture described in this paper has a broader cognition and affect remit.

To pursue the broad design of complete agents, in addition to the agent characteristics already introduced, further important qualities need to be introduced:

holistic designs: cognition is not a disembodied activity. It is related to and incorporates non-deliberative processing and not only in the sense of reactive (external) competencies. The impetus for much of cognition depends on activities occurring at non-cognitive (conative) levels.

meta-cognition: complete agents have the ability to monitor their own internal control states and modify or change them to further their aims or achieve them in a more effective manner.

self-actuation: a true complete agent can modify its behaviors, capabilities and environment in pursuit of its given or self-developed roles. Such an agent can define its own niche space, and therefore redefine its current problem space.

The second of these attributes requires, for example, that the agent can reason about its motivations, goals and plans and produce, what are in PRS terms, top-level goals or reorganize its existing goals so that their underlying motivators are more likely to be achieved. Such a capability means an agent is able to sense the emergence of cognitive states and so redirect internal (and possibly external) processes to better manage these emergent states. An agent displaying the third attribute should possess the capability to reason about its internal worlds and the processes that maintain and change it. An agent, whether natural or synthetic, displaying all these qualities is limited only by *its own* imagination. At present no such synthetic agent exists.

To explore the possibilities that agent architectures offer for the development of synthetic minds and the modeling of motivation and other control states, we are presented with a number of design options. For example we may want to produce a *deep* design for one or more components of the architecture, whereby we implement a system with a rich and accurate collection of capabilities capable of coping with a challenging environment, or believably display a range of psychological phenomena. Alternatively, we may produce a *broad* design (Bates et al 1991) for an agent architecture encompassing as many different kinds of *high-level* functionality as possible. The broad approach necessarily requires an initially shallow approach in designing computational agencies, with many issues addressed at a relatively coarse grain. In the short term there is no option but to produce a broad and shallow agent and attempt to iteratively deepen aspects of the design and implementation.

6. Using Agents to Implement the Model

Our experiments with agents in pursuing the strengths and inadequacies of this architectural theory of the mind have made use of a number of simulations in a small number of domains. It seems obvious, to us interested in complete agents, to develop hybrid architectures. Such agents fuse the strengths of the behavior-based approach, appropriate for many lower-level processing mechanisms, with the deliberative symbolic architectures useful for reasoning about the world and actions upon it. There are a number of interesting questions that need to be addressed. What makes a complete agent? What kinds of motivation and related processing structures need a complete agent have? What are the appropriate architectures and control issues for such an agent? Is it necessary for complete agents to make use a number of levels of processing (perhaps of differing modalities of operation) and if so, how can they be integrated? Does sensory data need to be restructured for different levels of internal processing (behaviors) in complete agents? What are the appropriate decision mechanisms to use in determining which among (possibly conflicting) behaviors are to be preferred in certain circumstances? What are the different learning algorithms necessary for the different levels within a complete agent?

To simplify initial development work, and ensure that the layers to the architecture function in their own right, each architectural layer has been investigated in turn using agents of increasing sophistication. These investigations into implementing the agent architecture have involved using a complete agent looking after a number of simpler agents in a dynamic, and possibly hazardous, synthetic environment. Alternative perspectives have also been explored. Reflexive and reactive (artificial-life based) processes have been explored using the niche spaces offered by the game of Go (Davis et al 1999; Davis 1999b), TileWorld (Hanks et al. 1993), and predator-prey game scenarios (Davis 2001). The combination of deliberative and reactive layers has been investigated using in decision-making scenarios (Davis 1999a) and e-commerce frameworks (Davis et al 2000).

Many of the experiments, described here, made use a single room environment. Other multiple room environments have been used. The environments used represent a set of possible factory floors with agents (of differing sophistication) performing different kinds of tasks; ranging from simple conveyance to collaborative maintenance. A typical task domain for a complete agent involves monitoring the more primitive agents in the environment and minimizing certain types of disruptive or harmful behavior. Environments are defined in terms of *Walls*, *Doors*, *Rooms* (defined in terms of Walls and Doors), *Hazards* and *Agents*. Doors can be connections between adjacent rooms or exit and entry points into the environment. An energy source, which the agents must find when their energy levels are becoming depleted, is typically present. Normally it is static, but when it needs recharging it needs to leave and re-enter the environment. As it does so, it constitutes a dynamic hazard capable of diminishing the processing power of nearby agents. All agents can visually sense the moving energy source, and so attempt to avoid collisions. Other hazards, which entail physical damage, typically consist of features such as ditches into which agents can fall and collisions with walls, objects or with other agents. In the most sophisticated of these experiments, an environmental monitor oversees the agents, their tasks and the environment. This is an abstract (at times disembodied) entity with its own representations of the world and tasks within that world. It models the deliberative and reflective layers of the architecture and can make use of reactive agents to affect changes in the observed environment. It is responsible for the monitoring of the environment and the initiation of actions for the agents that it subsumes. It contains its own private representational schemes, for example an explicit model of the environment, descriptions of current behaviors and tasks related to specific agents. It makes use of reactive and deliberative agents: to provide perceptual information about the environment; to develop and maintain motivational states; and to perform actions upon the world. It this agent (and its many sometimes distributed parts) that embodies the architecture depicted in figure 3.

Reflexive agents

This base level agent is a kind of mobile frictionless plinth modeling the perception-reflex-action loop of figure 3. Reflexive agents combine internal processing with perceptual and action processes to perform impulse behaviors and avoid collisions with agents and other objects within the environment. This tropistic class of agent is given a degree of autonomy and must navigate the environment without resource to persistent representations or higher level deliberative processes. A pure behavior-based design is appropriate for this class of agent.

A reflexive agent can be given up to three senses so that it can negotiate its way through the environment. This allows experimentation with processes that fuse data and information from several sensory sources. Auditory and memory danger senses have a 360° field, while vision is restricted to 180°, centered on the current direction of the agent. These agents move in one of four directions (north, south, east and west) in a continuous environment and are given an initial energy level, velocity and direction. Any change in velocity or direction causes energy to be consumed. When the energy level is reduced below a certain level, they must be recharged. The need to do this can be modeled as an implicit motivator. When their energy level reaches zero, they become static until *rescued*. Environmental competencies (for example stop, start and turn) define how the reflexive agents move around within their environment. Further reflexive behaviors such as accelerate, reverse and wander build upon this set of environmental competencies. The impulse to perform a specific behavior, in a purely reflexive agent, is activated entirely by perceptual information. For example, if an agent currently senses objects in front and to the right, an impulse to turn left will be initiated. If the agent senses objects in front and to the left and right, then a stop impulse will be initiated. The default behavior causes the agent to continue moving in the current direction with the current velocity - if stationary, the agent remains stationary. In the more sophisticated (reactive and deliberative) agents, the perceptual impulse to perform certain behaviors is extended with reactive and motivator initiated impulses. In such agents the environmental competencies can be extended to include the use of effectors.

Reactive agents

The second class of agents build on the mechanisms used in the reflexive agents but are given more flexible behaviors, and control mechanisms, similar to the architecture described by Kaelbling. This allows them to perform a wider range of more complex tasks. These reactive agents subsume the reflexive plinths and include extra perceptual pathways and mechanisms for integrating decision making and behaviors across the reactive-reflexive interplay of intended actions. These regulatory agents model all the stand-alone pre-attentive processes of our model. As in the reflexive agents, there is no persistent explicit representation of the external environment. However goals, their status and their processing structures persist over time. A reactive planner, similar to PRS, provides a design model for goal-selection mechanisms that make use of processing related to explicitly modeled motivational (or goal-oriented) states. The goals of these agents are typically self-centered reactions to internal (low-charge) and external (danger) states. However, reactive agents can be given low-level goals (for example, to perform simple conveyance type tasks) by deliberative agents, in the form of propositional instructions, e.g.

Move(Position1), *Grasp(Claw, Object*), *Move(Position2)*. The environmental competencies associated with these explicit goal-oriented motivational states are built from the more primitive (reflexive) actions already described. For example, to recharge itself the agent needs to move to a specific location (the energy source) while avoiding collisions using the stop, start, turn and default behaviors. Two subtypes, with differing preferences in the population density of their immediate environment, have been used to experiment with further motivational behaviors. For example the reclusive agent prefers less populated areas, seeks such areas when in more heavily populated areas and subsequently may perform less ably, due to goal conflicts, when unable to do so. The gregarious agent provides the antipathetic personality. Measurements such as goal persistence over time, number of actions to achieve goal and agent life expectancy provide evaluation criteria for these different agent subtypes.

Deliberative agents

The reactive class of agent, in turn, provides a computational platform for the deliberative agents. Deliberative agents combine the mechanisms used in the reflexive and reactive agents with deliberative processes, and allow an investigation into how these different categories of processing can be integrated. They extend the motivational states associated with the reactive agents to include more selfless goals such as rescue a trapped reclusive agent in a heavily populated room, recharge energy-less agents and investigate non-moving agents. This hysteretic class of agent maintains a persistent representation of the environment that includes beliefs and memories. Deliberative processes include explicit planning, management of goals and the selective focus of the agent's processes. The latter includes the contextual loading of the attention filter. Deliberative agents resolve behavioral conflicts arising from the proposed actions of the reflexive and reactive processing layers through the explicit consideration of surfaced goals. Surfaced goals are reactive goals that the agent must attend to at the deliberative level. These can arise from internal or external events. Internal event goals may be generated at any level but are only expanded and processed explicitly at the deliberative layer. Some goals related to attitudes and standards are generated at the deliberative layer from the agents' models of its environment. The agent reasons over its representation of the environment to generate these (typically selfless) goals. The memory model consists of simple propositional statements with an associated memory strength parameter. Memory strengths can take Boolean true or false values allowing the agent a set of assured beliefs. Memory strengths can also be real number valued. This provides a means for instantiating doubt into an agent when it needs to reason about its beliefs. This ensures the agent experiences a different form of internal conflict, and needs to reason over its memories in order to build a consistent or rational internal model of its external environment.

Reflective agents

This class of agent builds on the mechanisms present in the other three classes of agents. Their primary role is to monitor and help to control the internal processes arising from the interaction of these mechanisms and set high level contexts. In a full implementation, these agents should have the ability, through the management of internal processes or agencies, to control emergent properties of the agent architecture and so modify unwanted behavior and propagate beneficial states, whether designated or emergent. More recent research on this architecture (Davis 2000, 2001) uses a computational model of emotion to drive the reflective layer. In this model emotions exist in different computational guises over the different layers of the architecture (see figure 6). This provides reflective modes of access to the other layers that are related to the underlying metaphor of the reflective layer.

7. Computational Structure of Control States

Motivators do not exist in isolation but are related to control states and other phenomena. Control states can be explicit and associated with motivators or be implicit. Implicit control states are distributed among co-existing processes and memory structures. Control states such as desires and alarms (for example to replenish energy levels) can be made implicit within procedural triggers or object-oriented methods. The activation of such control states can make use of impulses at the reactive and reflexive levels and may give rise to explicit motivators at the reactive and deliberative levels. Pro-active desires, at the deliberative level, are implemented in an explicit manner as a set of rules and supporting methods. However the underlying standards and attitudes, for example to care for hapless agents in the environment, are implicit. These agents cannot, as yet, reason about social ethics!

A (semi-)formal notation, similar to McCarthy (1995), is used in designing and specifying epistemic events and helps to ensure that a consistent view of such events is maintained, at least at the design level. An agent-oriented mixture of objects, methods and symbolic rule-sets describe internal and external behaviors in all the agents. This combination of techniques provides a transparent means for representing behaviors and process oriented control states, for example beliefs, emotions, motivators and goals. There are rule-sets for the management of incoming information; the selection and activation of external behaviors; the generation, instantiation and management of motivators; planning; decision-making; and the management of memories. An explicit motivator structure (see table 1) is used at the reactive and deliberative layers of the computational model and draws together many aspects of the control state taxonomy. While many aspects of motivator management, for example attention thresholds, motivator rationale and resource management, are not addressed within the reactive agents, the same computational structure are used for motivators at the reactive level as at the deliberative layer where those aspects are considered.

The core of a motivator is a descriptor. This can be a single attitude, e.g. *make true, keep false,* towards a single proposition, e.g. *Avoid(ObjectA), Move(X,Y).* In many cases there is a need for multiple attitudes towards (possibly multiple) propositions. This can lead to the relaxation of constraints associated with the full generactivation of a motivator. For example, the emotional correspondence of a motivator may be activated through one of the propositions if other propositions remain false. Attitudes can also vary over the lifetime of a motivator. The management information component is used to maintain a memory of process information related to a motivator, for example the conditions that caused the motivator to be generated. The actors and entities component identifies factors such as sub-goals, plan set names, and the environmental objects and other agents referenced by a motivator. Importance, insistence, intensity and urgency are modeled using a set of qualitative values, e.g. *high, medium, low*. These values are calculated as needed and can also be modified during the lifetime of a motivator. For example, a recharge motivator has its urgency and importance calculated when generated at the reactive layer. A recharge motivator for an agent that cannot sense or is some distance from the energy source would have a high importance. An agent close to the energy source may generate a recharge motivator with a low importance. If the agent's energy level is low, the urgency value will be high. The insistence value takes the higher of these two components. The intensity of a motivator is calculated by a means of a comparison with other motivators and a consideration of the agent's current beliefs and control state context. Other components have internal effects on the motivator. A decay function can ensure that insistence decreases while a motivator remains passive or postponed. Such components respond to changes held as status information such as commitment status (e.g. *unknown, adopted, rejected, ignored*), and dynamic state (e.g. *passive, postponed, active, failed, successful, unachievable*). For example, for an adopted but passive recharge motivator, the insistence value changes in response to both the decay function and the increasing urgency and importance of the motivator.

Figure 4 shows a set of reactive plans for the motivator associated with the recharge behavior. Other goal-oriented motivators (for example rescue trapped agent, or investigate static agent) have similarly defined plans. At their simplest, these plans provide an exclusive serial choice, similar to rules in production systems. For example if *plan1* can be applied, satisfaction processes for that motivator are activated. If *plan1* is inapplicable but *plan2* can be applied then the agent will (continue to) recharge itself. Further serial choices require that the agent generate a further motivator to either move towards the energy source (*plan3*) or explore the environment and find the energy source (*plan4*). In certain situations, none of plans can be used. The generactivation of supportive generators can cause repetitive behavior in the reactive agents where motivators can be only partially expanded. The deliberative motivator management processes make use of memories related to past events (internal to a motivator) to catch these cyclic behaviors, and subsequently make use of a generic mechanism to activate a *plans-failed* process. This may activate an *abandon-motivator* process or call for runtime planning, and the generation of further motivators. Motivators generated as part of currently active goals are initially given the same insistence, importance, urgency etc. as their parent motivator. It is possible for circular or infinite motivator regression to occur, particularly where run-time planning is used or subtle differences between motivators bypass the fail-safes of the deliberative motivator management processes. In the majority of occasions such motivators bottom out as external behavior, typically as actions associated with explore and investigate goals. Circular motivator generation, which resemble an obsession with trying to achieve an impossible goal, also typify emergent internal behaviors that the reflective layer attempts to manage.

8. Decision-Making Processes

Different types of decision-making occur at the different layers of the computational model and in the different classes of agent. This approach to modeling agents, with competing independent behavior modules, means an agent needs some means of deciding between conflicting potential actions at the various levels in the architecture. For example, an agent cannot simultaneously turn left and right, but can look for the energy source, turn left and stop. However, decision-making needs to be made consistent across the different layers too. For example, an agent should not activate the expansion of motivators unrelated to the energy source at the deliberative layer, while choosing recharge behaviors at the reactive and reflexive layer. This does not prevent an agent wanting to pursue incompatible motivations.

Weights, or signal strengths, are associated with environmental competencies. The reflexive decision behavior module (in figure 5) contains rules that define compatible behaviors. It uses a *winner-takes-all* strategy for determining the next action. Potential behaviors are posted onto a reflexive behavior stack. All permissible combinations of currently posted behaviors are then added to the stack with their combined weights. The stack is sorted in terms of the behavior weight and their original position on the stack if of equal weight. More recent postings (i.e. higher placed behaviors) are preferred – these are more likely to be in tune with the state of the agent's internal and external environments. The potential action with the highest stack position is preferred. An alternative approach is to use neural networks trained to recognize appropriate environmental conditions for behaviors. Again a simple additive rule can be used to define a combination of actions (Davis et al 1995).

For agents with a number of levels of processing a more sophisticated approach is required. If no explicit motivators are generated, then even the most sophisticated agent acts as if it were a reflexive agent. For reactive agents or agents acting as if they were reactive with one or more reactive-level motivators, not all motivators need to be expanded at the deliberative level, the highest ranked motivator on the reactive motivator stack is chosen. The same winner-takes-all strategy as used at the reflexive level ranks motivators on the basis of their insistence value and recency. Motivator plans, involving action in the environment, are then executed at the reactive level. Where the mapping of plans onto low-level actions occurs, actions sanctioned by plans are given a real-number value reflecting the insistence value of the motivator. The external behavior actions of the selected motivator(s) are then posted to the reflexive behavior stack where further decisions may be required. An agent need not necessarily prefer the potential actions related to reactive (or deliberative) motivators to actions generated at the reflexive level. The agent decides from its set of (possibly conflicting) potential external actions at the reflexive level, as described above. To avoid the situation described above the agent must ensure that behaviors associated with a motivator are more heavily weighted. With a suitably insistent motivators this occurs as a matter of fact.

For deliberative agents, if a reactive motivator is generated and its insistence value is higher than the threshold of the attention filter, it is placed on the deliberative motivator stack. A pro-active motivator generated at the deliberative level is simply added to the motivator stack. An executive process, on noticing changes in the deliberative motivator stack, ranks the motivators in terms of their intensity, which can be calculated at this juncture. The most intense motivator can replace the current motivator. The motivator database is managed in other ways too. Deadlines, detailed from within the representational structure for motivators, can cause motivators to be deleted from the database. Motivators can also be stricken from the deliberative levels if they are abandoned during execution. Using Nilsson's (1994) teleological planning technique it is possible to combine and co-ordinate the execution of multiple motivators. For example an agent can recharge itself and another agent at same time, without pre-compiled plans to do both.

If a motivator is subsequently adopted, the attention filter threshold is set to the motivator's intensity value and the filter's deliberative context is changed to reflect the motivator context. In this way, very intense goals, requiring a total focus of the agent's deliberative processing, can effectively filter out all but very insistent or urgent goals from surfacing at the deliberative layer. The adoption of a motivator will include an evaluation of its current plans and whether it is currently achievable - a motivator can be accepted but further processing deferred until more urgent or important tasks achieved. Depending upon the current internal state of the agent, and what it knows, this may spawn further nested motivators or map onto actions to be performed on itself or other agents and objects in the environment. It is possible for an accepted ongoing motivator to provide an overriding context for all the processing of the agent. This occurs as the side-effect of the processing required for the internal components of the motivator. For example the emotional correspondence of a motivator may necessitate that an agent becomes fixated on that motivator or a motivator can carry a complex semantic content that relates not only to the agent's external environment but also internal processes. Motivators (and other control states) need not be related to an agent's external environment – they can be related to internal processes and representations (e.g. memories and beliefs). The plans associated with an adopted and active motivator, if related to actions on the external environment, are executed at the reactive level as described above. If further reactive motivators are generated (in subsequent time intervals) and their insistence value is greater than the attention filter threshold or their semantic content is supportive, then they are added to the motivator stack. Potential motivators not meeting these criteria do not surface at the deliberative layer and are eventually forgotten at the reactive level due to the decay function.

Another form of decision making relates to the management of memories and belief sets. Memories in deliberative agents are propositional statements extended with a real-numbered strength. As new information is added to memory, it reinforces similar information but weakens older disagreeing information. A variable quantitative threshold for forgetting, set by the memory manger processes at the deliberative level, means that sufficiently weak memories are deleted. Our agents cannot recollect lost memories. An expanded memory model, and fuller implementation, would also allow for qualitative thresholds and so allow more sophisticated memory-related cognitive capabilities to be investigated. Experimentation with an emotion-based strength for memories proceeds (Davis 2000). In the experiments described here, the agent needs to reason over possibly disagreeing memories to produce a consistent belief set. A consistent belief set is not always required. The belief set and the agent's memories are used in the generactivation of the pro-active motivators, and the agent can generate and manage incompatible motivators. The memory strengths provide a means for allocating and updating the insistence values of motivators.

9. Overview of Experiments

A number of different experiments were run to investigate design issues and different means of implementing control states and decision-making in the different agents and layers within any one agent. A number of these experiments involved placing an agent in a specific scenario with specific internal data and investigating how the agent made a decision and what structures it generated. Other experiments allowed the agents greater autonomy and provided metrics describing their behaviour in these experiments. Here we present details of two experiments used to determine the effectiveness of different action-selection mechanisms using the reactive and part of the deliberative agents. These experiments predate the full extent of the agent processing described in the previous two sections, but were instrumental in addressing design issues that enabled progress to be made.

These experiments make use of four types of agents and an energy source. The energy source can be static and benign or moving and damaging. The fundamental agents used in these experiments are reactive agents with no explicitly represented motivational states. These agents need to maintain energy levels and avoid colliding into agents and other objects within the environment through the use of ten competencies:

Default: If no other action is initiated the agent simply continues with it current behavior.

Stop: a base level behavior that brings the agent to an immediate halt.

Start: a base level behavior that causes the agent to move.

Turn: one of two base level behaviors that cause a 90° change in direction, either to the left or right.

Accelerate: a second level behavior causing an increase in velocity.

Decelerate: a second level behavior causing a decrease in velocity.

Reverse: a second level behavior causing a 180° change in direction.

Wander: a third level behavior causing the agent move around in the environment through a combination of reversing, turning, acceleration and deceleration.

Recharge: a third level behavior causing the agent to move to the energy source and increase its energy level.

These behaviors are activated through the use of perceptual information. For example, if an agent senses objects in front, behind and to the right, turning left will be the most appropriate behavior. If the agent senses objects to the front, left and right, but not behind, reversing will be an appropriate behavior. Wandering occurs if no object is in close proximity. Mreactives are reactive agents that make use of an extra (perceptual, internal and action-related) processing level for the explicit modeling of motivator states. There are two subclasses of Mreactive agents; one (Mreactive1) that explicitly models only the recharge drive (using the described motivator structure); and a second that has a further motivator. The second subclass of Mreactive agents consists of two subtypes: Mreactive2 that steer clear of the moving energy source; and Mreactive3 that seek areas of the environment populated by Mreactive2 agents. The inclusion of more than one explicit motivator allows us to experiment with conflict a between motivators. In single agent experiments, the Mreactive3 agents should perform similarly to the Mreactive1 agents as they will never sense any Mreactive2 agents and hence their second motivator should never be generated. Mreactive agents should perform better than the pure reactive agents - they possess the capability to autonomously defer some action, such as approaching the energy source when it is dangerous to do so.

The first experiment involves siting the energy source at the center of the room, with the single agent randomly placed in the room. The scenario is run until the agent dies and statistics for its age at death, number of behaviors lost and number of successful recharges are kept. The agent can die from three causes. Its energy level reduces to zero - it is unable to recharge. Its set of possible behaviors is reduced to zero - it has got too close to the moving energy source too many times, or it is destroyed outright by the moving energy source. The agent might not be able to recharge for at least three reasons. An agent might never sense and move to the energy source. An agent can sense the energy source but cannot reach it, because the energy source is moving at a greater velocity than the agent. Or in the case of the Mreactive2 agent the flee-dangerous-energy-source goal overrides the recharge goal sufficiently often that the agent's energy source is reduced to zero.

The experimental results given in table 2 show a number of interesting things. There is very little difference in the (mean) age of Mreactive1 and Mreactive3 agents as expected. The Mreactive2 agents however do not live as long. The plain reactive agents survive for an even shorter period. In the reactive agents, no matter how low their energy state, other behaviors (e.g. turn or reverse) can be preferred to selecting the recharge behavior. This can explain the poorer age when compared to the Mreactive1 agent, which has an explicit recharge goal which can override other behaviors. The Mreactive2 architecture, while having a better survival rate than the reactive agent, does not perform as well as the other two reactive agents on either survival rate or feeding occurrences (in fact it never fed in any of the 1000 runs!). This is an effect of the flee goal overriding the recharge goal and reducing its life expectancy. The behavior loss statistic supports this interpretation; the ratio of behavior loss to age (approximately 2:9) is lower than for the other Mreactive agents (approximately 5:8). We can therefore conclude the decision process used in the reactive agents is unsuitable for all types of behavior, and that by simply making explicit some high level behaviors (as in Mreactive1 agents) a better survival rate follows. It is a mute question what are the appropriate motivational control states best represented explicitly.

In a second set of experiments, five agents and the energy source are located in one room. The behavior set for all agents is slightly extended here, in that they are allowed to leave the room once they have reached a specific age. For the plain reactive agents this is just another third level behavior. For the Mreactive agents this is modeled as a further explicit goal. In all cases, four agents of one type (i.e. reactive or one of Mreactive subtypes) are placed near a fifth agent which acts as the experimental subject. Again statistics are collected for its age (either when it dies or leaves the experimental room). Each of these scenarios is run 1000 times. Table 3 displays the mean and variance for age (for when the agent either left the environment or died) for all these situations plus the percentage of agents that managed to leave the room. Notice how the life expectancy varies across the different scenarios for the Mreactive agents, but not for the plain reactive agents. For example, in the situation with four surrounding reactive agents, age (for the Mreactives) increases over that obtained in the single agent experiments for the Mreactives. It is reduced for the plain reactive agents over the single agent experiments presumably because the greater overall energy consumption rate increases the danger level of the environment - the energy source reaches its recycle threshold sooner and more often. While the life expectancy for the Mreactive agents is always greater than for the plain reactive agents, there is a dramatic decrease when more than one Mreactive agent is used. There must be some interplay between the effect of the different goals being pursued by the agents and the overall demands placed on the environment. No reactive-type agent ever manages to leave the environment. Although the age varies very little for the basic Mreactive agent over the three situations involving just Mreactive-type agents, there is more variation for the multiple-goal Mreactives.

The (Mreactive) approach of deliberative motivator processes tightly integrated to the reactive behaviors enabled us to more fully consider the nature of motivational processes, and further extend the characteristics and structures used for representing and handling goals. It raised a number of challenges related to the integration of decision mechanisms across the different layers of the model. The nature of the decision-making processes meant that actions chosen at the reflexive-reactive layer might not actually be associated with the currently active deliberative goal. The current deliberative goal would then be considered to have failed, while in actuality the decision-making at other layers were overriding its intended actions.

10. Discussion

One very real challenge with the type of research described here is that of evaluation. The previous section detailed some early experiments, but the type of metrics used in those experiments are of only limited use. The metrics approach used is flawed. Hanks and colleagues (1993) have disagreeing opinions on its suitability for the simpler categories of agents. What are the alternatives?

There are many ways of approaching this problem – although none are totally satisfactory. From one perspective, our model is working when all the control states can be identified in the computational agent and the internal behavior of the different layers in the agent, when in a specific set of control states, correspond to supportive processes for those control states. Similarly our agent can be said to be a complete agent when it demonstrates all the qualities (attributes) given in the definitions of the different categories of agents. Again, this approach to evaluation is unsatisfactory for a number of reasons. If our agent really is autonomous (and pro-active) we may have to wait a very long time before it demonstrates all the control states. It may never demonstrate them if we have provided it with deliberative and reflective processes appropriate for managing the more perturbant states. There is a deeper misgiving about this approach to evaluating complete agents. What are the indicators that distinguish the different control states? We need to become machine or agent psychologists and interpret the behavior and actions of the agents in order to answer such questions.

Consider the following non-exhaustive list of qualities, taken from (Johnson-Laird 1993) that a mind can be said to possess. A mind perceives the world and learns. A mind remembers and controls actions. A mind cogitates and learns new ideas. A mind controls communication with others. A mind creates the experience of feelings, intentions and self-awareness. We can repeat the checklist approach outlined above and consider which of these our agent implementations and designs can perform. If we did we would find that yes our agent can definitely perform some (say 60%), definitely not do others (say 20%), but again the remaining qualities are very difficult to evaluate.

At present, there does not seem to a wholly satisfactory evaluation methodology for this type of research - a point made in a review of Newell's work (Hayes-Roth 1993a). The AIS research team (Hayes-Roth 1995) is attempting to resolve this question by focussing their agent implementations in domains and situations that require human-like expertise. The suggestion from that and others research is that evaluation needs to be empirical. Architectures for intelligent agents need to meet a sufficiency criterion, and then be judged on architectural parsimony. Our implementations, our models and designs have yet to *completely* meet the sufficiency criteria for a complete (cognitive) agent. Put simply, there is no agreed sufficiency criteria for a complete agent!

An alternative is to measure research progress against the motivations for the research. Firstly, by producing plausible computational models of simulated agents we can further our understanding of biological, psychological and sociological agents. Through the design and implementation of agent architectures based on different theories of the mind, we can better understand the inadequacies and strengths of these theories. By developing working agent architectures, we can further our theories and models of control mechanisms for use in dynamic environments. We have yet to address all the issues but the nature of this research is very long term on which we can expect to make slow progress. We have yet to provide a full and integrated implementation of our control state taxonomy, but the research described here and in other papers (Wright et al, 1996, Davis 1996, Davis 1999a, Davis et al. 1999, Davis 2000, Davis 2001) addresses many of the states. We have made some progress towards our stated aims, but need to provide an integrated implementation to further the theory.

11. Future Directions

We have shown that it is possible to provide computational implementations based on a psychology and philosophy of the mind. However, there is a considerable amount that is missing in the computational agent, for example semantic memory, learning mechanisms for different types of internal and external behaviors, and the complete repertoire of control states. This developing theory does provide for architectures that allow an integration of reflexive, reactive, deliberative and reflective behaviors. Even though we pursue a broad approach, we are limited in what research issues we can tackle at any particular point. Some inadequacies in the work to date can be traced to these omissions. Nothing in the work, so far, has really undermined the four-layer principle to a theory of mind. Furthermore, the control state approach is still valid, though not necessarily an exhaustive description of what could be occurring within a mind. The taxonomy presented needs to be reviewed in light of further philosophical work (Chalmers 1995; Wollheim 1999).

The design and implementation experiments highlighted one of the major problems in complete agent research, as identified by Franklin (1997); that of integrating learning. At the design level, it is possible to see how different categories of learning segue well with certain capabilities or levels of processing. For example, Q-learning or an artificial neural net will be appropriate for the decision mechanisms in the reflexive-reactive agents. However, such techniques will not be appropriate across the range of activities inherent in the architecture. For example, inductive learning may be more appropriate for improving motivators and belief models at the deliberative level. However, we need to integrate learning in these multi-level computational models. Some form of (cross-representational scheme) hybrid learning, analogous to the work of Sesito and Dillon (1994), may allow the higher level processes to represent learnt behaviors and also allow the low level behaviors to optimize behavioral requirements requested by higher level processes. The problem is analogous to the experimental work of Lashley (1963) who addressed questions related to the central location of memory of learnt behaviors in biological agents (rats in his case). The design and attempted implementation of a centralized learning mechanism seems analogous to Lashley's (fruitless) search for the engram (i.e. the neurophysiological basis of memory). No one single process (however connected) could account for these mechanisms at a theoretical, design or computational level. Any attempt to do so is almost bound to arrive at the problems highlighted here and also by Franklin.

Current and future research will build on these experiences and look to dynamic agent architectures, allowing heterogeneous communities of agencies that mirror multiple cognitive (and affective) activities. Not only can we specify the initial capabilities and behaviors of an agent, but also the conditions for how its constituent agencies may create new agencies tuned to the requirements of specific tasks and their related control states. This may lead to co-operative and competitive cliques of emergent processing and there is the possibility that these communities may overwhelm the processing of the overall agent, leading to undesired control states. Learning mechanisms in this scenario are agent frameworks within which certain classes of agencies, and perhaps communities of agents, have their mechanisms, processes and knowledge re-calibrated to suit changing environmental or niche-role pressures.

The concept of niche space and design space is still found to be useful. Further analysis of the niche space for complete agents is extending both the breadth and the depth of the agent architectures in design space. Current work (Davis 2000, 2001) presents a central emotion-based core for the theory and across all layers of the agent architecture (see figure 6). This agent-oriented internal environment will allow a reflective (temporally global) navigation of deliberative processes and control state related communities of (reactive and reflexive) processes. As the agent monitors its interactions within itself and relates these to its tasks in the external environment, the impetus for change within itself (i.e. a need to learn) is manifested as a combination of emotive states. Such a control state can lead to the generactivation of a motivator requiring the agent to modify its behavior or processes in some way. The modification of an agent's internal environment is then situated in terms of an emotion (control-state) mapping between its internal and external environments.

12. References

Agre, P. and D. Chapman, 1987, PENGI: An implementation of a theory of activity. *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87), pages 268-272*, Seattle, WA.

Bates, J., A.B. Loyall and W.S. Reilly, 1991, Broad agents, *SIGART BULLETIN*, 2(4).

Beaudoin, L.P., 1994, *Goal Processing in Autonomous Agents*, Ph.D. Thesis, School of Computer Science, University of Birmingham.

Brooks, R.A., 1991, Intelligence without representation, *Artificial Intelligence*, 47.

Brustoloni, J.C., 1991, *Autonomous Agents: Characterisation and Requirements*, Carnegie Mellon University Technical Report, CMU-CS-91-204.

Chalmers, D., 1995, The Conscious Mind : In Search of a Fundamental Theory, Oxford University Press.

Cherry, E.C., 1957, *On human communication: A review, a summary and a criticism*, MIT Press.

Davis, D.N., 1996, Reactive and motivational agents. In: *Intelligent Agents III*, J.P. Muller, M.J. Wooldridge and N.R. Jennings (Editors), Springer-Verlag, Heidelberg.

Davis, D.N., 1998, Synthetic Agents: Synthetic Minds? Frontiers in Cognitive Agents, *IEEE Symposium on Systems, Man and Cybernetics, San Diego.*

Davis, D.N., 1999a, Agent-Based Decision Support Framework For Water Supply Infrastructure Rehabilitation *International Journal of Computers, Environment and Urban Systems, Vol 24, 1-18*.

Davis, D.N., 1999b, Computational Emergence and Computational Emotion, *IEEE Symposium on Systems, Man and Cybernetics*, Tokyo, Japan.

Davis, D.N., 2000, Agents, Emergence, Emotion and Representation, *IEEE International Conference on Industrial Electronics, Control and Instrumentation, (IECON2000), Nagoya, Japa*n.

Davis, D.N., 2001, Multiple Level Representations of Emotion in Computational Agents, *Emotion, Cognition and Affective Computing, AISB2001:Agents and Cognition*, University of York.

Davis, D.N., A. Sloman and R. Poli, 1995, Simulating Agents and Their Environments. *AISB Quarterly*.

Davis, D.N., T. Chalabi, and B. Berbank-Green, 1999, Towards An Architecture for A-life Agents:II, In: M. Masoud (Editor), *New Frontiers in Computational Intelligence and Its Applications*, ISO Press.

Davis, D.N., K. Liu & Y. Luo, 2000, A multi-agent framework for stock trading, 16th IFIT World Computing Congress, Beijing, China.

Ferguson, I.A., 1995, Integrated Control and Coordinated Behavior: A Case for Agent Models In *Intelligent Agents ECAI-94 Workshop on Agent Theories, Architectures and Languages, pp 203-218, Springer-Verlag*.

Ferber, J., 1999, *Multi-Agent Systems*, Addison-Wesley.

Franklin, S.P., 1995, *Artificial Minds*, The MIT Press, Cambridge, Mass.

Franklin, S.P., 1997, Autonomous Agents as Embodied AI, *Cybernetics and Systems, Vol. 28, No. 6*.

Franklin, S.P. and A. Graesser, 1996, Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. *Intelligent Agents III*, J.P. Muller, M.J. Wooldridge and N.R. Jennings (Editors), Springer-Verlag, Heidelberg.

Genesereth, M.R. and N.J. Nilsson, 1987, *Logical Foundations of Artificial Intelligence*, Morgan Kaufman.

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

Hanks, S., M.E. Pollack, and P.R. Cohen, 1993, Benchmarks, Test-beds, Controlled Experimentation, and the Design of Agent Architectures, *AI Magazine, 14(4):17-42.*.

Hayes-Roth, B, 1993a, On building integrated cognitive agents: A review of Newell's Unified Theories of Cognition. *Artificial Intelligence*, 59.

Hayes-Roth, B., 1993b, Intelligent control. *Artificial Intelligence*, 59.

Hayes-Roth, B., 1995, An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence*, 72.

Kaelbling, L.P., 1989, An Architecture for Intelligent Reactive Systems, *Readings in Planning*, Morgan Kaufmann.

Lashley, K., 1963, *Brain Mechanisms and Intelligence*, New York.

McCarthy, J., 1995, Making Robots Conscious of their Mental States. *Machine Intelligence 15*, Oxford.

Maes, P., 1989, How to do the right thing, *Connection Science*, Vol. 1.

Newell, A., 1990, *Unified Theories of Cognition*, Harvard University Press.

Nilsson, N.J., 1994, Teleo-reactive programs for agent control, *Journal of Artificial Intelligence Research* 1.

Oatley, K. and J.M. Jenkins, 1996, *Understanding Emotions*, Blackwell.

Ortony, A., G.L. Clore and A. Collins, 1988, *The Cognitive Structure of Emotions*. Cambridge University Press.

Sesito, S. and T.S. Dillon, 1994, *Automated Knowledge Acquisition*, Prentice-Hall.

Simon, H.A., 1967, Motivational and emotional controls of cognition, *Models of Thought*, Yale University Press.

Sloman, A., 1990, Motives mechanisms and emotions. M.A. Boden (Editor), The Philosophy of Artificial Intelligence, Oxford University Press.

Sloman, A., 1993, The mind as a control system, In: *Philosophy and the Cognitive Sciences*, C. Hookway and D. Peterson (Editors), Cambridge University Press.

Sloman, A., 1998, The Semantics of Evolution: Trajectories and Trade-offs in Design Space and Niche Space? *IBERAMIA'98*, Lisbon, Springer-Verlag

Sloman, A., L. Beaudoin and I. Wright, 1994, Computational modeling of motive-management processes, *Proceedings of the Conference of the International Society for Research in Emotions*, N. Frijda (Editor), ISRE.

Tambe, M, W.L. Johnson, R.M. Jones, F. Koss, J.E. Laird, P.S. Rosenbloom and K. Schwamb, Intelligent Agents for Interactive Simulation Environments, 1995, *AI Magazine*.

Wollheim, R., *On The Emotions*, Yale University Press, 1999.

Wooldridge, M. and N.R. Jennings (Editors), 1994, *Intelligent Agents*. Springer-Verlag.

Wright, I.P., A. Sloman, and L. Beaudoin, 1996, Towards a Design-Based Analysis of Emotional Episodes, *Philosophy Psychiatry and Psychology, Voume 3*.

Figure Captions

Figure 1. A relatively simple IT trajectory through niche space and design space.

Figure 2. Taxonomy of control states.

Figure 3. A plausible architecture of a mind based on the work of (Sloman 1990).

Figure 4. A set of plans and related processes associated with the self-centered recharge goal.

Figure 5. A four layer architecture for the partial implementation of the design in figure 3.

Figure 6. Current experimental architecture for work on computational emotion.

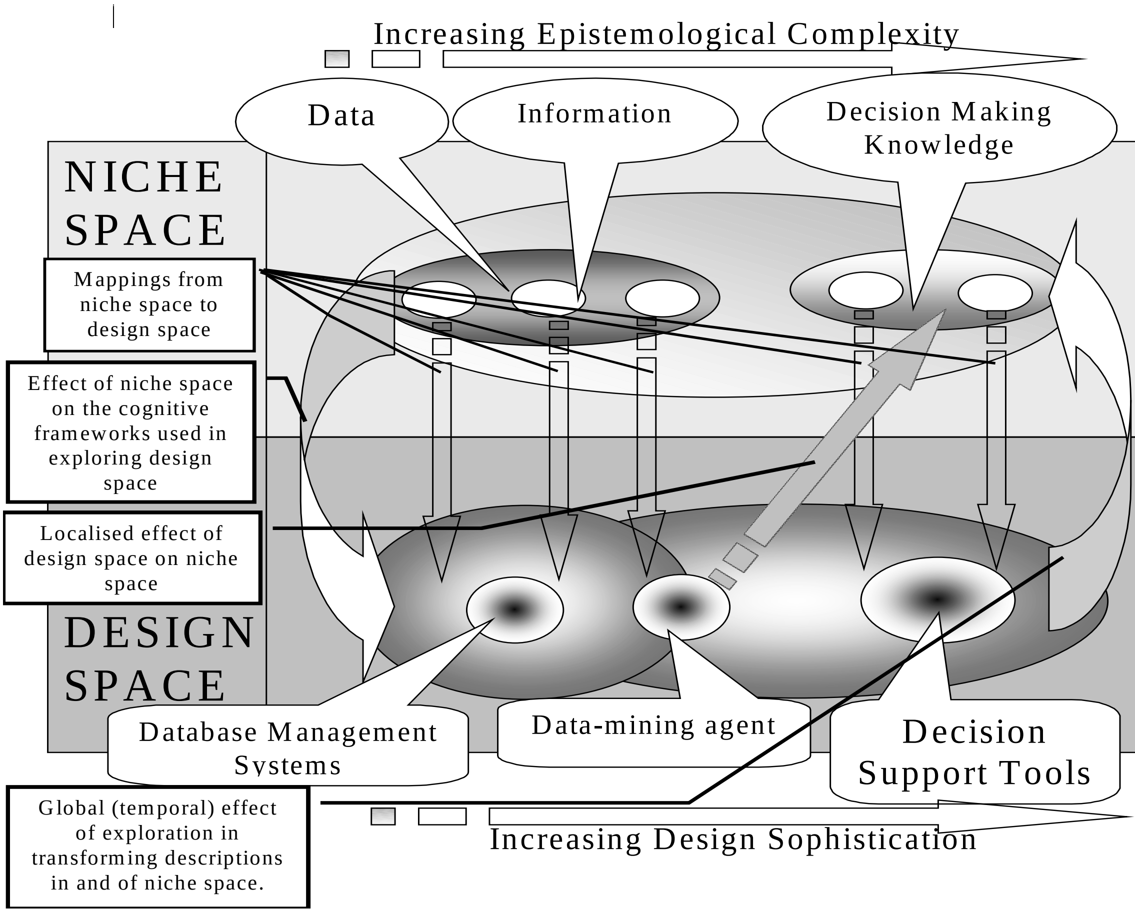

Figure 1. A relatively simple IT trajectory through niche space and design space.

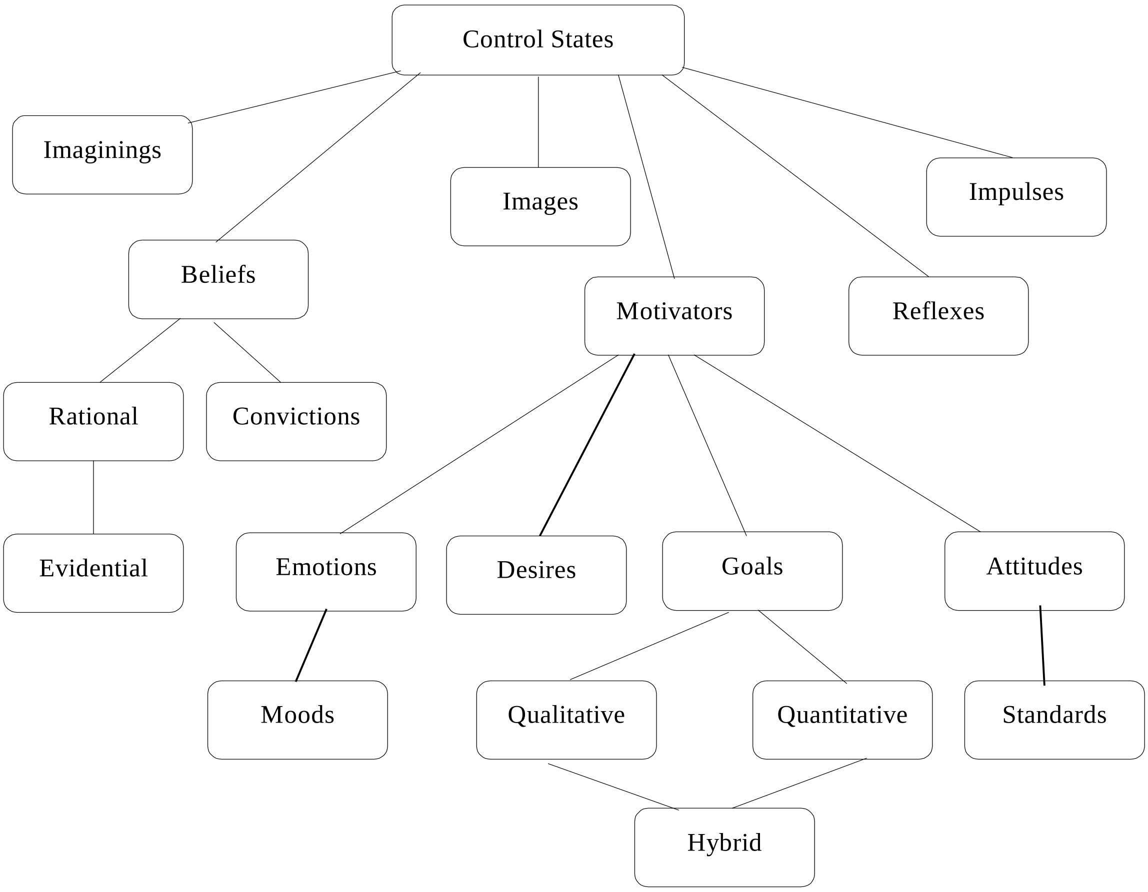

Figure 2. Taxonomy of control states.

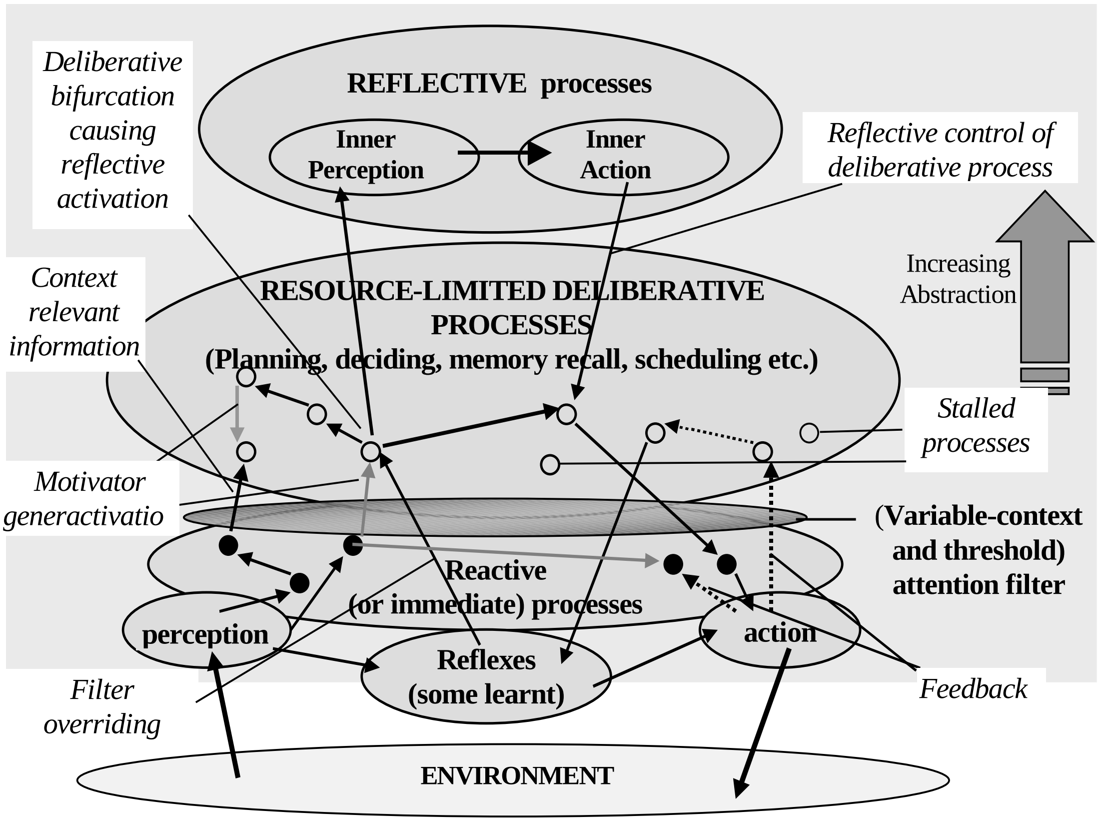

Figure 3. A plausible architecture of a mind based on the work of (Sloman 1990).

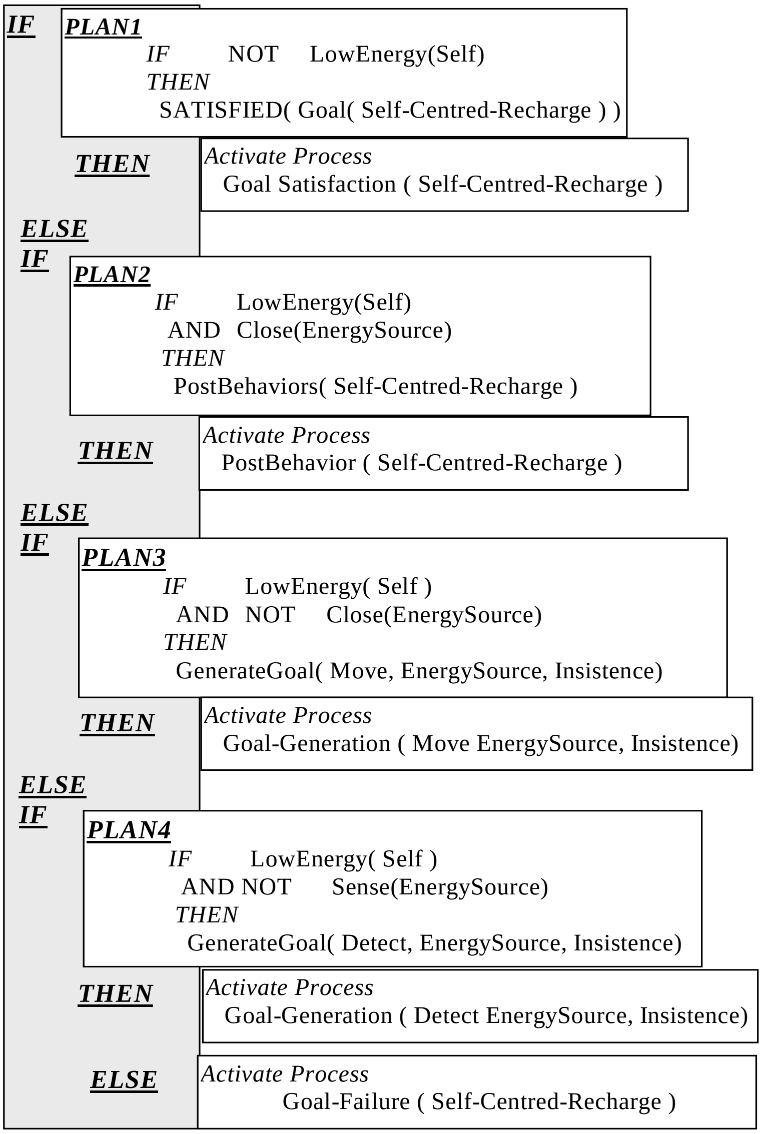

Figure 4. A set of plans and related processes associated with the self-centered recharge goal.

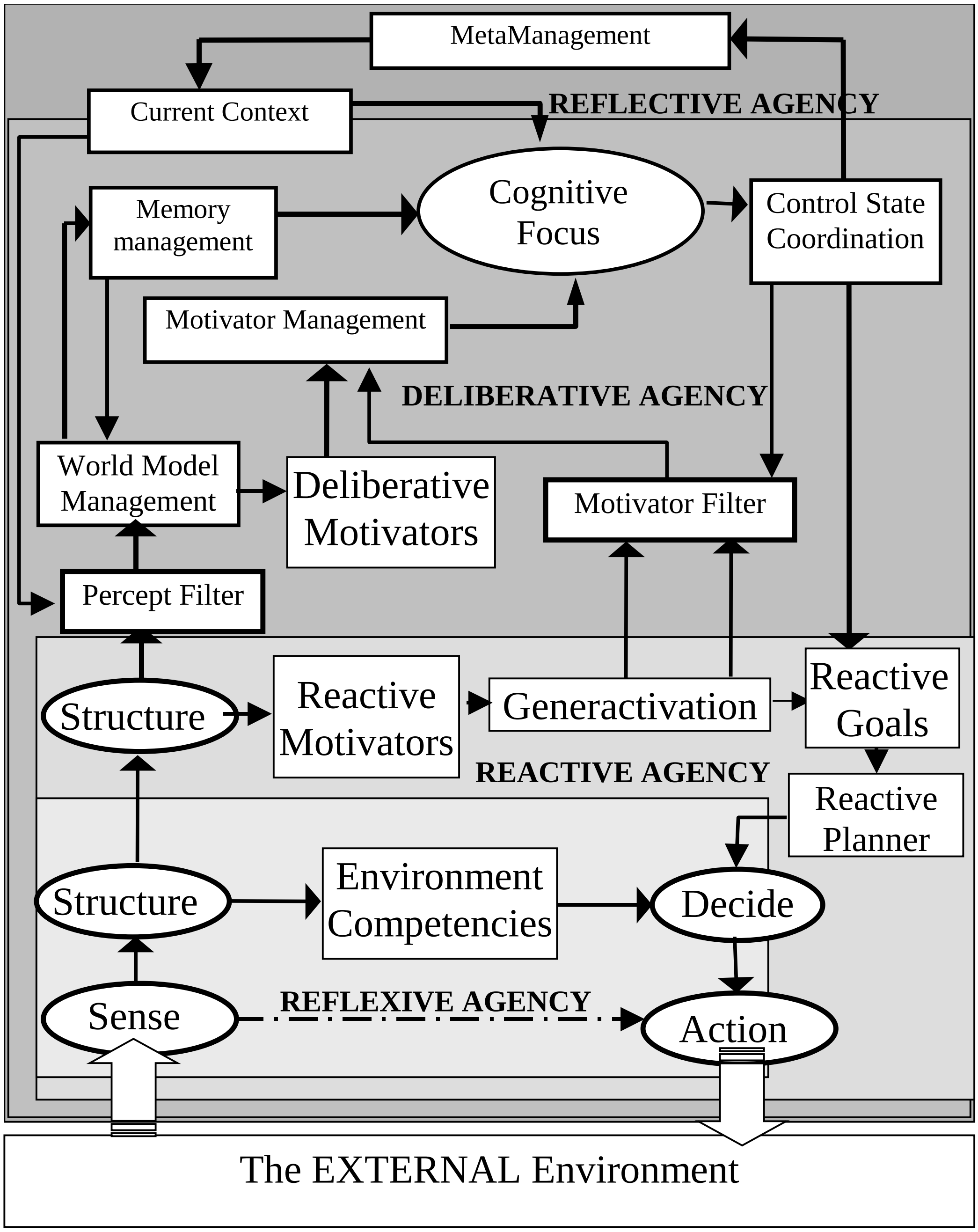

Figure 5. A four layer architecture for the partial implementation of the design in figure 3.

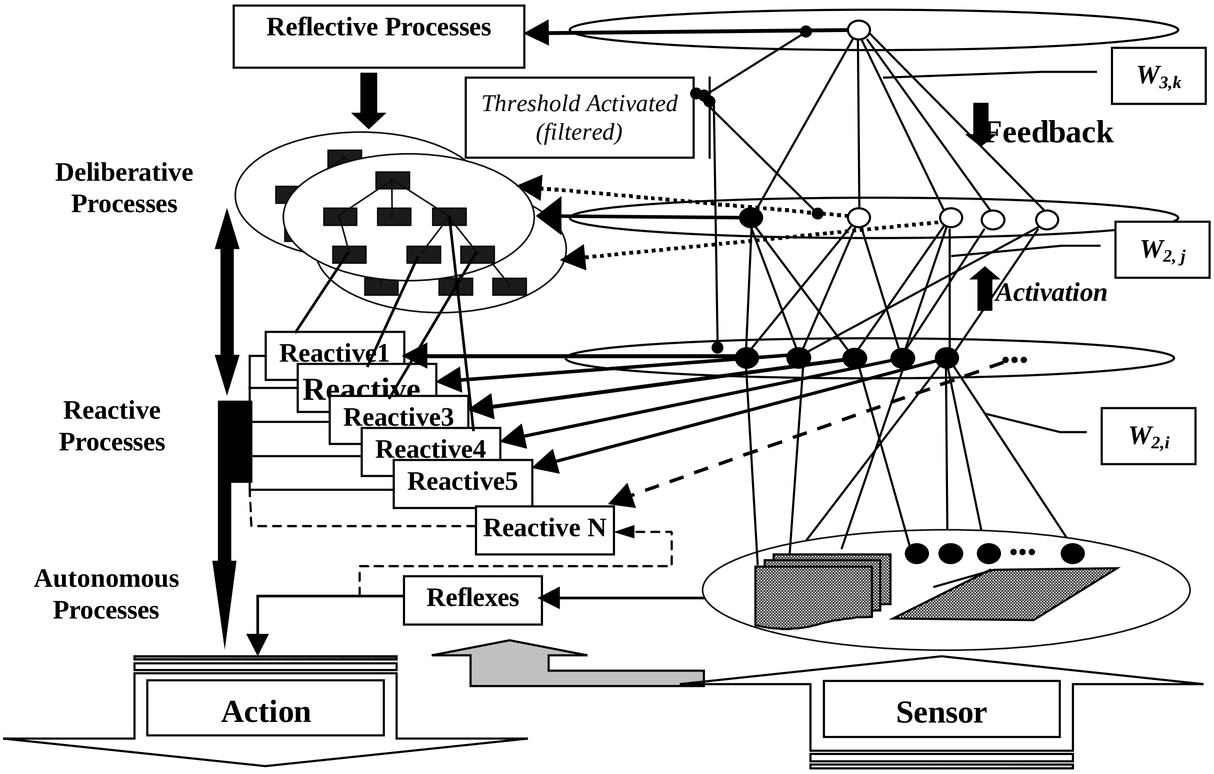

Figure 6. Current experimental architecture for work on computational emotion.

Table 1. The main components associated with motivator structures.

Component	**Meaning**
Actors and Entities	Other agents (actors) and objects (entities) referenced by this motivator
Belief Indicator	Indication of current belief about the status of P: e.g. *true, nearly true* etc.
Commitment Status	The current status of the motivator, e.g. *adopted, rejected, undecided, interrupted, stalled* etc.
Decay Function	Defines how insistence decreases while motivator remains unadopted.
Dynamic State	The process state of the motivator e.g. being considered, nearing completion etc.
Emotional Correspondence	The emotions and their situational triggers associated with the motivator, *e.g. If Unsuccessful Then Activate(Fear) . If Successful Then Activate(Happy)*
Importance Value	Importance (e.g. neutral, low, medium, high, unknown). This may be intrinsic or based on an assessment of the consequences of doing or not doing
Insistence Value	Heuristic determining interrupt capabilities. This should correspond to a combination of the motivator's importance and urgency.
Intensity	This influences the likelihood of (continuing) to being acted on.
Management Information	The state of relevant management and meta-management processes.
Motivational Attitude	The motivator's attitude to P : *make true, keep true, make false* etc.
Plan Set	Possible plan or set of plans for achieving the motivator.
Rationale	If the motivator arose from explicit reasoning - motivators need not.
Semantic Content	A proposition P denoting a possible state of affairs, which may be *true* or *false*
Urgency Descriptor	How urgent is this descriptor – this may be qualitative (e.g. *high*, *low*) or quantitative (for example a time-cost function).

Table 2. Recharge Experiment Results. (Each run one thousand times).

Architecture	Attribute	Min	Max	Mean	Variance
Reactive	Age	3	28	10.14	2.27
Reactive	Losses	0	4	0.06	0.35
Reactive	Recharges	0	2	0.05	0.22
Mreactive1	Age	5	166	23.94	17.83
Mreactive1	Losses	0	15	1.62	1.76
Mreactive1	Recharges	0	6	0.19	0.73
Mreactive2	Age	5	31	18.28	4.86
Mreactive2	Losses	0	4	1.00	0.57
Mreactive2	Recharges	0	0	0.0	0.0
Mreactive3	Age	3	155	23.85	14.64
Mreactive3	Losses	0	15	1.65	1.63
Mreactive3	Recharges	0	5	0.14	0.53

Table 3. Results for multiple agent experiments. (Each run 1000 times).

Subject	**Surround**	**Min Age**	**Max Age**	**Mean Age**	**Var Age**	**% of agents left**
Reactive	Reactive	3	42	7.36	3.64	0
Reactive	Mreactive1	3	23	7.41	2.91	0
Reactive	Mreactive2	3	23	7.30	2.89	0
Reactive	Mreactive3	3	31	9.14	3.47	0
Mreactive1	Reactive	5	205	55.84	29.33	3.2
Mreactive1	Mreactive1	5	163	28.01	15.46	1.2
Mreactive1	Mreactive2	5	159	9.31	15.75	1.6
Mreactive1	Mreactive3	5	91	29.0	8.30	1.2
Mreactive2	Reactive	5	122	54.51	27.20	1.4
Mreactive2	Mreactive1	5	127	17.88	8.12	0.1
Mreactive2	Mreactive2	5	51	17.47	7.02	0.0
Mreactive2	Mreactive3	7	76	25.68	10.94	4.8
Mreactive3	Reactive	5	197	55.37	27.59	3.3
Mreactive3	Mreactive1	5	187	28.03	15.23	0.7
Mreactive3	Mreactive2	11	107	21.05	4.20	0.0
Mreactive3	Mreactive3	27	107	31.67	4.62	0.0