Synthetic Minds

Aaron Sloman

School of Computer Science
The University of Birmingham
Birmingham, B15 2TT
England
Phone: +44 121 414 4775 (Sec 3711)
Fax: +44 121 414 4281
A.Sloman@cs.bham.ac.uk

Abstract ID: A150
Category: paper
Application Area: synthetic agents (including robots)
Technical Issues: agent architectures, meta-modeling of an agent by itself, models of motivation (and emotions),

Abstract

This paper is about the conditions under which some of the "higher level" mental concepts applicable to human beings might also be applicable to artificial agents. The key idea is that mental concepts (e.g. "believes", "desires", "intends", "mood", "emotion", etc.) are grounded in assumptions about information processing architectures.

An architecture generates a family of interrelated concepts describing the states and processes that can occur in that architecture. Which concepts are applicable to a system will depend on the architecture of that system. For example certain sophisticated architectures support notions of self-control and loss of self-control, such as occurs in some emotional states, whereas others do not.

McCarthy has discussed some of the reasons why we shall need to be able to describe intelligent robots in mentalistic terms, and why such a robot will need some degree of self consciousness. He has made suggestions regarding both the notation that we might use to describe its mental states and the notation the robot might use to describe its own states. This paper extends that work by focusing not on the notation but on the underlying "high level" architectures required.

[Note to reviewers: I apologise for not being able to finish revising this paper in time for the deadline. If it is accepted I shall try to compress several portions and try to be more precise about the relationships between specific mentalistic concepts and the underlying architecture, instead of simply giving some examples. I shall, in any case, continue working on it.]

1 Introduction: how should synthetic agents be described?

People often wish to describe programs using language that has hitherto been restricted to descriptions of human beings or possibly other animals. In particular, it is often convenient to describe a machine as “choosing”, “exploring”, “deciding”, “inferring”, etc. These are *intentional* states and processes, since they have semantic contents. This paper explores the idea that such mentalistic concepts implicitly assume a certain sort of high level information processing architecture. That is, what Dennett [4] calls the *design stance* is implicit in much of our ordinary thinking about ourselves. If this is correct, similar architectural requirements will need to be satisfied by artificial agents if the mentalistic terms are to be applicable to them.

Our mentalistic concepts have two aspects. First they describe some of the information stored in, used by, missing from, or investigated by an agent. Secondly the concepts implicitly refer to what can loosely be described as “control functions”, namely how such information is produced or changed, and how it affects what happens in the agent, which might include changing other control states or information states.

For example, the state of wanting a holiday in Portugal includes both some sort of reference to Portugal and the agent being there at a future date, and also a *prima facie* tendency to take steps relevant to bringing about that situation, including both internal steps such as thinking about possible dates, possible means of travel, finances, etc. and also external steps, such as visiting a travel agent. The tendency to take these steps may be overridden by all sorts of factors, e.g. lack of opportunity to travel, choosing to do something more important, being too busy to think about the holiday, concern about costs or risks, and so on.

This leads to the notion of a variety of stores of information, of many different kinds, with different causal powers and different causal relationships, which interact with each other and themselves, and also with external interfaces such as sensors and motors or muscles and limbs. The precise configuration of such interacting collections of information defines a high level information processing architecture.

Many mentalistic predicates (e.g. “believes”, “desires”, “prefers”, “intends”, “fears”) have several aspects. They include *semantic content*. They include an attitude to that content, which may be accepting it as true, wanting it to be true, wanting it to be false, etc. Such attitudes involve *dispositional* properties of the state. Dispositions may or may not be realised in actual occurrences. Some of the occurrences will be observable behaviour, while others are changes within the architecture, including changes of the contents of information structures or changes in dispositions, or both. So as Ryle [13] noted many of the dispositions and tendencies and capabilities relate not to observable behaviour but the creation, removal, or modification of other *mental* states.

It is extremely difficult to specify the precise manner and conditions in which various mental states manifest themselves, whether in internal processes or in external behaviour. That is because almost any potential effect can be overridden by another state that undermines its potential. (Conditional expressions in computer programs are just one example, inhibitory and excitatory links in neural nets are another.) Thus all generalisations about the external manifestations of mental states require a somewhat ill defined *ceteris paribus* qualification.

If we move to a slightly lower level of description, such as might be used by a software engineer designing a complex agent, then we can talk about the architecture in a manner that is less vague and indeterminate, and which explains the possibility of all the phenomena referred to in mentalistic descriptions. I’ll call this lower level “the information level” and say more about it below.

2 Transferring mental states to artificial agents

The concern about applying mentalistic descriptions to artificial agents raises the following questions:

(a) Under what conditions (if any) would it be appropriate to apply these mentalistic descriptions to artificial agents?

(b) Will the mentalistic descriptions that are appropriate for human beings be sufficient for all kinds of artificial agents or will we eventually require new kinds of mentalistic concepts, e.g. for distributed agents, or agents that explore information networks?

(c) Which (if any) of the descriptions will be appropriate only for physically situated agents with their own bodies and which will also be appropriate for agents that inhabit only software environments or possibly virtual realities that mirror physical environments?

Some philosophers insist on the following constraints on applicability of mentalistic concepts.

The history constraint: the language for describing mental states is applicable only to agents that have a biological history and are products of evolution.

The physical embedding constraint: mental states with semantic content are possible only for an agent that possesses a physical body with transducers giving it direct access to physical reality.

These constraints are inherently arbitrary for the following reason. Consider predicates such as "understands English", "likes mathematics", "is careless", "is good at chess". Whether or not they do have the above two constraints, they also say a great deal more than that an individual satisfies those constraints. This additional (diverse) content is far richer than the two constraints. So if we consider predicates just like these, but not subject to those constraints, then we can talk about complex systems for which we would otherwise lack suitable terminology.

The first constraint makes it difficult to talk about a very sophisticated robot assembled in a factory. This is an arbitrary restriction if all the other requirements for the predicates are satisfied. The second constraint rules out agents that inhabit software worlds or virtual reality systems, or agents all of whose mental processes are directed to such tasks as investigating number theory, or exploring relationships between games like Go or Chess. Since an agent inhabiting such a world may share a significant proportion of the information processing capabilities of a physically embedded agent there is no good reason for not allowing the same mentalistic descriptions to be applied to it, as long as they are taken to refer to the agent's capabilities, not the nature of the environment. For more on this see [15, 12].

If it is objected that it is *impossible* for agents without a biological history, or without physical embedding, to satisfy the remaining "internal" conditions, then that needs to be argued. Merely stipulating it as a defining condition is no argument.

Like [12], I shall assume from now on that such *extraneous* criteria are irrelevant. Provided the system can *do* the right sorts of things, now and in the future, the mentalistic predicates will be assumed to be applicable to it. But what are the capabilities, and what sorts of machines can have them?

3 Two extreme approaches to description

Those who work on artificial agents need a vocabulary with which to describe the systems they design. There are two extreme options, the first of which is more common:

(a) free use of terms from ordinary language, so that robots and software agents are described in terms of having beliefs and desires, taking decisions, making plans, perceiving things, learning things, and so on; and

(b) using precisely defined technical terms describing only the algorithms and data structures used, along with the sensory inputs and visible movements.

The problem with option (a) is that developers often find it hard to see the limitations of what they are doing, and may therefore exaggerate its significance by over-interpreting it in terms of familiar categories. McDermott (in [9]) criticised AI researchers who describe their programs using words and phrases from ordinary language as "wishful mnemonics", e.g. "Goal", "Understand", "Planner", "General Problem Solver", where the programs do not support this.

An example of this sort of mistake is to have variables called "love", "hate", "anger", and the like, which are given numerical values which are modified by the program and which affect the program's subsequent behaviour.[1]

Over-interpreting systems using such variables is analogous to assuming that the degree of overloading of a computing system can be thought of as a numerical variable in the system whose value controls the speed with which the system responds to commands. In fact, overloading is something that emerges from the interactions between the components of the system and the programs that are running in it. The types of overloading that are possible will depend on the architecture (as illustrated below). Likewise, the types of mental states that are possible will depend on the architecture.

Option (b), describing systems solely in terms of the details of implementation, using the technical terminology of the implementation discipline, may be more accurate, but has problems of its own. First it makes the work sound far less interesting, and may therefore not attract additional funding or sales!

More seriously, restricting descriptions to low level details may fail to capture some of the most interesting features of the system, just as attempting to describe human beings entirely in terms of their neural structures and their physical interfaces with the environment would leave out aspects that we know to be important, and make it impossible for us to describe some of the more abstract common features of groups of individuals, e.g. some are ambitious, some are aggressive, some are good at philosophy, some are superb composers, some find mathematics difficult to learn, some love their country, some enjoy poetry, some believe they are immortal, and so on.

In some cases, a description only at the implementation level will make it impossible for people to use or interact with the system – like describing a word processor only in terms of what it does with bits, bytes, and various internal data-structures, rather than describing it in terms of letters, words, lines, sentences, paragraphs and so on. (Compare [6].)

In addition there are many episodic descriptions of mental phenomena which play a role in explaining particular actions. "He turned round because he heard a footstep and wondered who it was." "He glared angrily because he thought he was being accused of theft." "He looked at her gratefully because she had not given away his secret." And so on. We may need similar mentalistic descriptions to explain the behaviour of human like robots or software agents: e.g. "It notified me of Fred's message because it assumed it was from Fred Bloggs".

[1] I have no objection to the use of such controlling variables where the objective is entertainment or shallow simulation, as in the design of "believable" agents, e.g. [1].

4 Bridging levels

Can we build a bridge between the levels, and so get the best of both worlds, by providing the power and convenience of mentalistic descriptions, whilst providing precise justifications for their use? This will not necessarily involve *reducing* mentalistic concepts to lower level concepts. For instance, it is commonplace in science to introduce theoretical explanatory constructs that are not *defined* in terms of pre-existing constructs. One mechanism for doing this is the use of what Carnap in [3] called "meaning postulates".

This can be done by adding mentalistic descriptive concepts to the implementation vocabulary along with a lot of statements linking new concepts to the lower level descriptions, to one another, and to visible behaviour and external conditions, but without ever giving necessary and sufficient conditions. This is analogous to presenting something like an architecture diagram indicating flow of information and control. The predicates describing relatively long term states will correspond to enduring nodes in the architecture. Others may correspond to short term memory contents, such as percepts, plans, new motives.

We can call this "explicating" mentalistic concepts in terms of concepts used to specify the (high level) functional architecture. This requires showing how global capabilities, states, events and processes are explained in terms of the functional roles of the components and their capabilities, states, events and processes. On the basis of a good theory of the architecture we may be able to produce a useful taxonomy of the states and processes that can occur at the mentalistic level of description.

An example: "overloaded"

I'll first illustrate this with a non-mentalistic, but high level, functional concept. Suppose we wanted to analyse the notion of a machine being "overloaded". Understanding this requires grasping (a) that within a computing system there are various kinds of resources, (b) that during the functioning of the system these resources can sometimes turn out to be inadequate, and (c) that as a result performance is impaired in various ways, including some tasks taking longer, some programs no longer being capable of running, errors or inaccuracies occurring, etc.

The particular sorts of overloading that can occur will depend on the precise architecture, e.g. the number and power of the processing units, whether they have memory caches, the cache sizes, the size of RAM, the amount of disk space available, the speeds of the data buses and disk controllers, and so on. We can then say that there are different sorts of overloading, depending on which of these components is being used to its full capacity and therefore blocking some programs from getting things done and perhaps causing others to crash, e.g. for lack of memory. The same machine might be overloaded in different ways on different occasions.

For instance on one occasion it may simply have so many concurrent programs running on it that the queue of waiting processes is always very long and each user therefore gets a very slow response. If the CPU were faster it could keep the queue much smaller. This sort of overloading does not include any wasted effort, only delays. It might be reduced by providing more CPUs or speeding them up.

On another occasion, in the same machine, there could be a smaller number of processes, and a shorter job queue. If each process is very big then switching between them, and switching between locations within a process may require a lot of disk traffic caused by swapping and paging, as a result of which the system spends a lot more of its CPU time on these administrative tasks than on running user programs. This sort of overloading involves wasted effort, as well as delays. It could be reduced by providing more memory or by speeding up the disks, disk controllers and disk interfaces. This is what is meant by describing a system as “thrashing”. (Note that this is a state of the whole system, not part of the system.)

Yet another kind of overloading may be caused by a high proportion of cache misses, which could be reduced by increasing the cache size. On the other hand, in a multi-processor system where each processor has its own cache, the larger caches may cause more interactions between processes that write to the same memory location, leading to different sorts of delays.

These are all examples of internal system states that arise not because the system is *designed* to produce them, but because the architecture supports functionality, which, if applied in certain situations, can produce unexpected emergent states.

If the system also has the ability to monitor its own states and processes a new variety of descriptions becomes applicable. We could describe a system as learning that various kinds of overloading can occur, as reporting aspects of its state, as asking for help, or taking remedial action itself. What sorts of actions are available (e.g. killing processes, switching some process to a different machine, disallowing new logins till the load drops, changing the priorities so as to favour a subset of processes) will depend on the system’s architecture.

Just as we can use the term “overloaded” to refer more or less loosely to a variety of functionally describable phenomena in computing systems the same is true of mentalistic labels, e.g. “harassed”, “careless”, “afraid”, “excited”, etc. However the architectural presuppositions of these descriptions are more sophisticated and far more complex architectures are needed. The work in the Birmingham Cognition and Affect group over the last few years has been attempting to elaborate on these ideas. (See [26, 16, 17, 18, 20, 2, 25, 27]. Compare [14] and [10]).

5 An example: “careless” and “careful”

McCarthy [6, 7] adopts a similar viewpoint and attempts to identify and define axiomatically a class of predicates describing mental states. In the first of the two papers, he wrote (in [6]):

> To ascribe certain beliefs, knowledge, free will, intentions, consciousness, abilities or wants to a machine or computer program is legitimate when such an ascription expresses the same information about the machine that it expresses about a person.

This of course begs the question when it is legitimate to ascribe these things to persons, and what information such ascriptions express. I claim that when we use mentalistic language to talk about ourselves or others, we are actually implicitly making assumptions about the architecture of human minds, namely we are assuming that there are various coexisting interacting subsystems with different functional roles, for instance, perceptual subsystems, various types of memory, various skill stores, motivational mechanisms, various problem solving capabilities. There is no reason why we should not transfer these predicates to artificial agents, provided they have a sufficiently rich architecture to make these descriptions apt.

For example, describing X as “working carelessly” implies (a) that X had certain capabilities relevant to the task in hand, (b) that X had the ability to check and detect the need to deploy those capabilities, (c) that the actual task required them to be deployed (e.g. some danger threshold was exceeded, which could have been detected, whereupon remedial action would have been taken),

(d) that something was lacking in the exercise of these capabilities on this occasion so that some undesirable consequence ensued or nearly ensued. (It need not actually have ensued: carelessness in a car driver can cause a near miss.)

Different sorts of architectures could satisfy these conditions in different ways. And within a single architecture how the conditions are satisfied would depend on which task was in question. For a particular task there are various ways in which carelessness could occur, including the following:

• X forgets the relevance of some of the checks (a memory failure),

• X does not focus attention on the data that could indicate the need for remedial action (an attention failure),

• X uses some shortcut algorithm that works in some situations and was wrongly judged appropriate here (a selection error),

• X does not process the data in sufficient depth because of a misjudgement about the depth required (a strategy failure),

• X failed to set up the conditions (e.g. turning on a monitor) that would enable the problem to catch his attention (a management failure).

Each of these can be unpacked in terms of things done or not done by components of X's architecture. This is not meant to be an exhaustive analysis, merely an indication of how a simple and familiar mentalistic description can relate to a design architecture, i.e. a collection of coexisting interacting capabilities within X.

Law courts are often concerned with finding out exactly which of these sorts of things went wrong, e.g. when someone is accused of negligence. The result of such investigations is sometimes acquittal or reduction of punishment, even when it is agreed that the agent did what he was accused of. It is not usually noticed that the courts (including juries) are thereby implicitly using theories about the architecture of a mind.

Similar comments are relevant to the description of X as "careful". This is not an appropriate description if X does not have the sort of architecture which provides opportunities for carelessness to arise. A clock cannot tell the time carefully or carelessly.

These architectural assumptions in ordinary language are vague, ill defined, and only implicit in mentalistic concepts. Which ones are applicable may vary somewhat from individual to individual or culture to culture, and they are not necessarily always true of the agent in question, for instance when there is some brain abnormality or underdeveloped collection of skills caused by abuse in childhood.

A task for agent theorists then is to devise a more accurate and explicit theory of the types of architecture to be found in human minds (and perhaps others) and use the architectures as a framework for defining families of descriptive concepts that are applicable to different sorts of humans (including for instance infants, and people with various kinds of brain damage) and different sorts of artificial agents.

There need not be just one architecture. Humans differ from one another, and the same human develops through infancy, childhood, adolescence and so on. For instance, very young children who have a strong desire for something (e.g. sweets visible in a supermarket) may not be able spontaneously to direct attention back to that desire after they have been distracted by a wise parent. A few months or years later, the architecture develops so that motivational states can remain active and regain attention after temporary diversions.

Similarly, both naturally occurring alien intelligences and artificial human-like agents may turn out to have architectures that are not exactly like those of normal adult humans. A human psychopath may be lacking in some of the mechanisms required for generating strong motivations relating to the needs or concerns of others. The normal "altruistic" motivational mechanisms are not necessarily simple instantiations of some general mechanisms of intelligence, for, as many biologists have remarked, they could be directly planted by genetic mechanisms which have evolved to ensure that the needs of a gene pool are not always overridden by the needs of an individual. Asimov's "laws of robotics" were based on a similar design requirement for artificial agents.

6 Design space

As there are possible differences in architecture, there may also be semantic indefiniteness in the terminology used to describe states in the architecture, owing to inevitable imprecision regarding architectural presuppositions. For instance talking about what a young child with limited reasoning capability and a restricted set of concepts knows or believes may have different connotations from talking about what a normal adult knows or believes. Likewise different robots or software agents. One advantage of developing a theory of a space of possible architectures [6, 7, 19, 21, 22, 23] is that it provides a framework for more precise specifications of alternative families of mentalistic concepts.

In discussing the use of mentalistic descriptions of machines McCarthy [6] writes:

> It is useful when the ascription helps us understand the structure of the machine, its past or future behavior, or how to repair or improve it. It is perhaps never logically required even for humans, but expressing reasonably briefly what is actually known about the state of a machine in a particular situation may require ascribing mental qualities or qualities isomorphic to them. ... Ascription of mental qualities is most straightforward for machines of known structure such as thermostats and computer operating systems, but is most useful when applied to entities whose structure is very incompletely known.

Whilst agreeing with most of this, I am trying to extend McCarthy's ideas with the following suggestion: which concepts are applicable will depend in a systematic way on the functional architecture of the machine or organism in question. This appears to go against his last sentence, but the apparent conflict can be resolved by pointing out that knowing something about the *high level* functional organisation of a system is consistent with very incomplete knowledge, or even complete ignorance about details of the implementation and details of the contents of most of the sub-systems.

In [8] McCarthy defines taking the "functional stance" towards something as treating it as having a particular sort of function without knowing anything about how is implemented. I am talking about a combination of the "design stance" [4] and the functional stance: an individual studied is thought of has having a complex architecture, with many components to which we takes the functional stance. Because this stance ascribes semantic contents to many of the components I call this the "information level" of description. (The components need not be physically separable: they could be aspects of a virtual machine.)

Example: the rechargeable robot

For example, we can use the information level when we think about a robot (or complex software system) X in the following way:

- X includes a priority-based resource-allocation mechanism R,
- R assigns a high priority to keeping X in a fully functional state
- R has the information that being fully functional requires the battery charge level not to be low
- R has access to information about the charge level of X's batteries
- the charge level of X's batteries is now very low,

This can be used to predict (though never with complete certainty) that X is likely to take action to replace or recharge its batteries. The prediction is not certain because we don't have information about the priorities assigned to other things, we don't know that all the sub-mechanisms are currently fully functional (e.g. low battery charge may cause some of the self-monitoring to become unreliable), we don't know that all the reasoning that the relevant sub-systems are capable of performing will be performed, we don't know which other internal processes can interfere with the resource allocation, and so on.

Thus both prediction and explanation in this sort of situation are subject to assumptions of "normality". This means that statements about the architecture are not generally falsifiable on the basis of predictions about external behaviour. Only by some more direct examination of the implementation can one tell whether statements about the architecture are true or false, and if one has not actually designed the system that may be very difficult, or even impossible in practice.

7 The information level

In [22] this functional, information-based, level of description, often used by software engineers, was described as "the information level". In some ways this is close to the sort of reasoning which Dennett [4] refers to as taking the "intentional stance" and the level of description that Newell [11] describes as "the knowledge level". However, both of them claim that this level of description presupposes that the agent being described is rational (or nearly rational), whereas I claim that something deeper is presupposed which requires the "design stance", namely that the agent has a certain sort of functional architecture which explains observable capabilities, but which may or may not always function perfectly. This level of description is below Newell's knowledge level and above his physical and symbol levels.

It is an instance of Dennett's design stance, but already assumes semantic contents, which he restricted to objects viewed from the *intentional* stance. (Apparently he unwittingly switched to the design stance and information level descriptions in [5]).

The information level requires no assumptions about rationality. Whether the whole system is rational or not will depend on how the functional components are combined and on how they cope with the exigencies of particular situations encountered. This is why we cannot assume human beings, or intelligent robots, will be always, or even mostly, rational.

Analysis at the information level presupposes that there are various stores of information (not necessarily physically distinct, and not necessarily expressed in any particular type of language or formalism, as explained in [24]), which have functional roles defined by (a) where the information comes from, (b) how it is stored, (c) how it is processed or transformed before, during and after storage, (d) whether it is preserved for a short or long time, (e) how it can be accessed, (f) which other components can access it, (g) what they can do with the information, and so on.

I claim that the ordinary notions of having a belief, desire, intention, hope, fear, attitude, etc. all relate to the existence of diverse information stores with diverse contents and functional roles within the architecture. But they do not presuppose rationality because there are all sorts of ways in which interactions between these components of the functional architecture can produce irrational decisions or actions. Even within the framework of folk psychology we can make allowance for impulses, obsessions, memory lapses, various kinds of carelessness, temporary misjudgements of relative importance, and so on. With a richer theory of the architecture we can account for far more types of occurrence.

If we know more about the architecture we can also allow the possibility of a wide variety of forms of processing which do not produce what we would regard as rational decisions and actions. This could be either because of malfunction of components or because of high level consequences of interactions between properly functioning components. Some of these might be thought of as due to hardware or software "bugs", whereas others may simply be "natural" consequences of the complexity of information absorbed by the system, like the occurrence of thrashing or other manifestations of overloading in a computer.

An example might be an agent who frequently reacts with others in such a way as to make them dislike him, simply because he has learnt the wrong collection of social skills, or perhaps because he rightly believes that it is impossible to detect whether others intend to harm him and lacking any sensible strategy for dealing with that, invokes behaviours partly at random which do not serve his interests.

All this could be regarded as a paraphrase of some of the implicit assumptions of psychoanalysis. Freud's theory of the unconscious, including his explanations of slips of the tongue, constituted an important, though largely unsuccessful, attempt to expand the commonsense architectural presuppositions. He was ahead of his time and did not have adequate design concepts for the task.

8 Architecture-based vs logic based concepts

The fact that architecture can be important as a basis for ascribing mental states is implicitly acknowledged in some of what McCarthy writes, for example in his detailed discussion of the reasons why a machine may require a form of self-consciousness, where he explicitly refers to internal structures and their states.

His main method is to introduce new primitive predicates and define them by specifying axioms. I.e. they are defined implicitly by their mutual relationships, as described above.

The programme outlined here is more ambitious in that it requires description of (very high level) machine architectures and definition of states and processes within those architectures in terms of which we can define new global states of the whole system. Moreover I think it is premature to make any assumptions about whether agents use logic or other forms of representation (diagrams, neural nets, etc.), whereas McCarthy [7] also states:

> There will usually be other data structures and programs, and they may be very important computationally, but the main decisions of what do are made by logical reasoning from sentences explicitly present in the robot's memory. Some of the sentences may get into memory by processes that run independently of the robot's

decisions, e.g. facts obtained by vision.

9 Conclusion

This paper has begun to sketch some ideas about the relationships between architectures and high level mentalistic descriptions, whether applied to natural or artificial agents.

One reason for much of the vagueness is that we still do not know what kinds of architectures are possible nor which ones will provide global capabilities that are similar to those of humans and other animals. It is very difficult to settle these issues empirically because most relevant aspects of the information processing architecture in an organism do not map in any simple way onto either observable behaviour or the physically distinguishable structures and relationships within brains. Moreover, we may not even have the right concepts for specifying architectures with the right sorts of capabilities.

I believe the only way to make real progress is to allow different types of exploration to proceed in parallel, including philosophical analysis, psychological and neurophysiological studies of humans and other animals, experiments with a variety of working designs for synthetic agents, and experiments with artificial evolutionary processes which might throw up important types of architectures that we would not think of designing ourselves. In our laboratory we have built a toolkit for exploring a variety of different types of agent architectures[2] but we have to collaborate with researchers adopting other approaches. [3]

References

[1] J. Bates, A. B. Loyall, and W. S. Reilly. Broad agents. In *Paper presented at AAAI spring symposium on integrated intelligent architectures*, 1991. (Available in SIGART BULLETIN, 2(4), Aug. 1991, pp. 38–40).

[2] L.P. Beaudoin. *Goal processing in autonomous agents*. PhD thesis, School of Computer Science, The University of Birmingham, 1994.

[3] R. Carnap. *Meaning and necessity: a study in semantics and modal logic*. Chicago University Press, Chicago, 1947.

[4] D. C. Dennett. *Brainstorms: Philosophical Essays on Mind and Psychology*. MIT Press, Cambridge, MA, 1978.

[5] D. C. Dennett. *Consciousness Explained*. Penguin Press, Allen Lane, 1991.

[6] J. McCarthy. Ascribing mental qualities to machines. In M. Ringle, editor, *Philosophical Perspectives in Artificial Intelligence*, pages 161–195. Humanities Press, Atlantic Highlands, NJ, 1979. (Also accessible at http://www-formal.stanford.edu/jmc/ascribing/ascribing.html).

[2] Described in http://www.cs.bham.ac.uk/~axs/cog_affect/sim_agent.html

[3] For more on our work see the Cognition and Affect Project ftp directory: **ftp://ftp.cs.bham.ac.uk/pub/groups/cog_affect**

[7] J. McCarthy. Making robots conscious of their mental states. In *AAAI Spring Symposium on Representing Mental States and Mechanisms*, 1995. Accessible via http://www-formal.stanford.edu/jmc/consciousness.html.

[8] J. McCarthy. What has artificial intelligence in common with philosophy? In *Proc. 14th International Joint Conference no Artificial Intelligence*, 1995. (Accessible via http://www-formal.stanford.edu/jmc/aiphil/aiphil.html).

[9] D. McDermott. Artificial intelligence meets natural stupidity. In John Haugeland, editor, *Mind Design*. MIT Press, Cambridge, MA, 1981.

[10] M. L. Minsky. *The Society of Mind*. William Heinemann Ltd., London, 1987.

[11] A. Newell. The knowledge level. *Artificial Intelligence*, 18(1):87–127, 1982.

[12] R.A.Young. The mentality of robots. *Proc. Aristotelian Soc.*, 1994.

[13] G. Ryle. *The Concept of Mind*. Hutchinson, 1949.

[14] H. A. Simon. Motivational and emotional controls of cognition, 1967. Reprinted in *Models of Thought,* Yale University Press, 29–38, 1979.

[15] A. Sloman. What enables a machine to understand? In *Proc 9th IJAI*, pages 995–1001, Los Angeles, 1985.

[16] A. Sloman. Motives mechanisms and emotions'. *Emotion and Cognition*, 1(3):217–234, 1987. Reprinted in M.A. Boden (ed), *The Philosophy of Artificial Intelligence*, 'Oxford Readings in Philosophy' Series, Oxford University Press, 231–247, 1990.

[17] A. Sloman. On designing a visual system (towards a gibsonian computational model of vision). *Journal of Experimental and Theoretical AI*, 1(4):289–337, 1989.

[18] A. Sloman. Prolegomena to a theory of communication and affect. In A. Ortony, J. Slack, and O. Stock, editors, *Communication from an Artificial Intelligence Perspective: Theoretical and Applied Issues*, pages 229–260. Springer, Heidelberg, Germany, 1992.

[19] A. Sloman. The mind as a control system. In C. Hookway and D. Peterson, editors, *Philosophy and the Cognitive Sciences*, pages 69–110. Cambridge University Press, 1993.

[20] A. Sloman. Prospects for ai as the general science of intelligence. In A. Sloman, D. Hogg, G. Humphreys, D. Partridge, and A. Ramsay, editors, *Prospects for Artificial Intelligence*, pages 1–10. IOS Press, Amsterdam, 1993.

[21] A. Sloman. Explorations in design space. In *Proceedings 11th European Conference on AI*, Amsterdam, 1994.

[22] A. Sloman. Semantics in an intelligent control system. *Philosophical Transactions of the Royal Society: Physical Sciences and Engineering*, 349(1689):43–58, 1994.

[23] A. Sloman. Exploring design space and niche space. In *Proceedings 5th Scandinavian Conference on AI, Trondheim*, Amsterdam, 1995. IOS Press.

[24] A. Sloman. Musings on the roles of logical and non-logical representations in intelligence. In Janice Glasgow, Hari Narayanan, and Chandrasekaran, editors, *Diagrammatic Reasoning: Computational and Cognitive Perspectives*, pages 7–33. MIT Press, 1995.

[25] A. Sloman. What sort of control system is able to have a personality, 1995. Available at URL ftp://ftp.cs.bham.ac.uk/pub/groups/cog_affect/ Aaron.Sloman.vienna.ps.Z, (Presented at Workshop on Designing personalities for synthetic actors, Vienna, June 1995).

[26] A. Sloman and M. Croucher. Why robots will have emotions. In *Proc 7th Int. Joint Conf. on AI*, Vancouver, 1981.

[27] I.P. Wright, A. Sloman, and L.P. Beaudoin. Towards a design-based analysis of emotional episodes, 1996(to appear). Available at URL ftp://ftp.cs.bham.ac.uk/pub/groups/cog_affect in the file Wright_Sloman_Beaudoin_grief.ps.Z.