The "Semantics" of Evolution: Trajectories and Trade-offs in Design Space and Niche Space.

Aaron Sloman
School of Computer Science, Univ. of Birmingham, Birmingham,
B15 2TT, UK
A.Sloman@cs.bham.ac.uk
http://www.cs.bham.ac.uk/~axs/

June 16, 1998

Abstract

This paper [1] attempts to characterise a unifying overview of the practice of software engineers, AI designers, developers of evolutionary forms of computation, designers of adaptive systems, etc. The topic overlaps with theoretical biology, developmental psychology and perhaps some aspects of social theory. Just as much of theoretical computer science follows the lead of engineering intuitions and tries to formalise them, there are also some important emerging high level cross disciplinary ideas about natural information processing architectures and evolutionary mechanisms and that can perhaps be unified and formalised in the future.

1 Introduction: Exploring Design Space

AI can be construed as the exploration of the space of possible designs for (more or less) intelligent agents, whether natural or artificial [4, 13]. Designs are not all static. Some systems change aspects of their own design: they modify themselves, through learning, adaptation, or architectural development, e.g. from embryo to infant, and from infant to adult. Brain damage or disease can also produce design changes with deleterious consequences.

All these changes move a machine or organism from one region of design space to another. Possible routes through design space can be thought of as *trajectories* in the space.

Some regions of design space are not linked by possible trajectories for individual development. An acorn can transform itself into an oak tree, and by controlling its environment you can slightly modify what sort of oak tree (e.g. how big). But no matter how you try to train or coax it by modifying the environment, it will never grow into a giraffe. The acorn (a) lacks information needed to grow into a giraffe, (b) lacks the architecture to absorb and use such information,

[1] Invited talk for IBERAMIA'98 Lisbon, October 1998. http://www.springer.de/comp/lncs/index.html

and (c) lacks the architecture required to modify itself into an architecture that can absorb the information.

Trajectories that are possible for an individual which adapts or changes itself will be called *i-trajectories*. Different sorts of i-trajectories could be distinguished according to the sorts of mechanisms of change, e.g. innately determined development, reinforcement learning, facilitation by repetition, and various kinds of self-organising processes partly driven by the environment.

Trajectories which are not possible for an individual machine or organism but are possible across generations of individuals subject to a particular type of evolutionary development will be called *e-trajectories*. Examples include development of humans and other animals from much simpler organisms and modifications of software structures by genetic algorithms. Conjectured e-trajectories leading to human minds are discussed in [5] and [18].

Whether two designs are linked by an e-trajectory or not will depend on the type of evolutionary mechanism available for manipulating genetic structures and the type of ontogenetic mechanism available for producing individuals (phenotypes) from genotypes. In biological organisms the two are connected: the ontogenetic mechanism can also evolve. Lamarckian inheritance would allow i-trajectories to form parts of e-trajectories.

There are also some changes to individuals that are possible only through external intervention by another agent, e.g. performing repairs or extensions, or debugging software. Such changes follow *r-trajectories* (repair-trajectories).

Viewing a species as a type of individual, e-trajectories for individuals form i-trajectories for a species, or a larger encompassing system, such as an ecology. Researchers in AI, Alife and evolutionary computation all contribute to the study of such trajectories. This study is in its infancy.

2 "Semantics of Evolution"

In computer science "semantics of computation" refers to abstract, mathematical, properties of programming languages, data structures and the processes which can occur in virtual machines of various sorts. Likewise a study of the most general features of evolutionary trajectories in design space addresses a topic that could be called "semantics of evolution" (though both differ from the more common use of the word "semantics" in linguistics).

Milner [9] noted that theoretical computer science follows the lead of engineering intuitions and tries to formalise them. Since the intuitions are often very subtle and complex the process of formalisation can lag far behind. Likewise attempts to study and formalise the space of possible designs and the various trajectories in design space will lag behind intuitive understanding gained from empirical research in biology, psychology, and computational explorations. This paper attempts to identify some of the phenomena to be formalised.

Many have attempted to formalise features of evolution, individual learning, development etc. Kauffman [7] describes mathematical laws which constrain biological mechanisms and processes in surprising ways. The ideas discussed below deal with phenomena which at present are too ill defined for mathematical formulation and computational modelling.

3 Generalising Fitness Landscapes

Evolutionary trajectories are often represented in a “fitness landscape” where a fitness value is associated with various locations in design space. If a class of designs can be specified by two parameters (e.g. length and stiffness of a spring), then there is a 2-D space of designs, and adding a measure of fitness of the spring for a particular purpose, produces a 3-D fitness landscape. Typically, design spaces are far more complex than this, and cannot be specified by a fixed number of parameters, e.g. designs for Prolog compilers vary in structure and complexity. Moreover, many designs have no single fitness measure: Prolog compilers vary according to their portability, the speed of compilation, the speed of compiled code, the size of compiled code, the kinds of error handling they support, etc.

A set of constraints and requirements for a design can be called a “niche”. Fitness of designs for organisms and artefacts may be compared in relation to a niche. Requirements for a compiler can vary just as the requirements for a plant or insect can. So there is a space of possible niches: “niche space” ([13, 15, 17]).

Designs and niches are both abstract and can have different concrete instantiations. Designs don’t presuppose a designer and requirements (niches) don’t presuppose a requirer. There are different ways actual requirements can be generated: e.g. engineering goals vs biological needs and pressures.

Two insects in the same physical location can be in different niches, so niches are not determined by physical location. Neither are they simply in the eye of a beholder: niche-pressure can influence movement of individuals or a species in design space. This can happen in different ways. If the individual has adaptive capabilities it may move along an i-trajectory so as to fit a niche better. Alternatively the pressure may cause a gene pool, or a subset of a gene pool, to move along an e-trajectory.

There are many different sorts of causal relations: within an architecture, between architectures, between architectures and niches, between niches. Niches can interact with one another by producing pressure for changes in designs, which in turn can change niches, as in co-evolution of organisms. So there are trajectories in niche space as well as design space.

Where independent changes in different dimensions are possible, a complex niche can cause parallel design changes, e.g. making an organism both physically stronger and better able to recognize complex physical structures. Problems arise when the changes are not independent: e.g. increasing agility may conflict with increasing strength. Where there are pressures for incompatible changes, which one actually occurs may depend on subtle features of the total context, and two identical individuals may be pushed along divergent trajectories because of slight differences in context.

Feedback loops occur where changes in one individual or group alter the niche, and therefore the causal influences on another individual or group. Such feedback can lead to continual change, to oscillations, or to catastrophes.

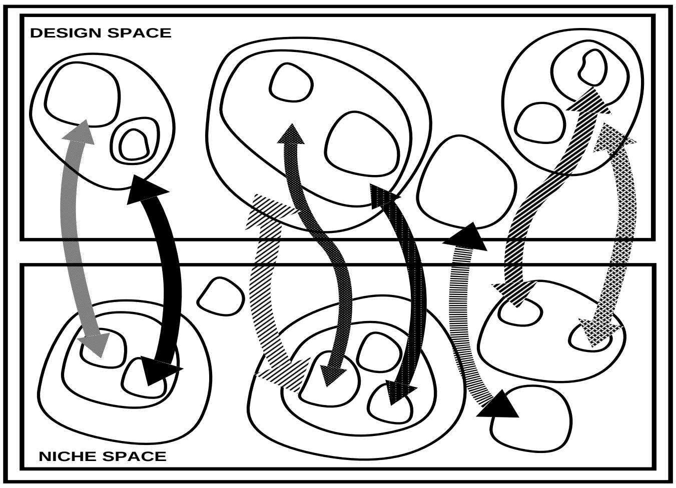

Figure 1: Design space and niche space: Relations between designs and niches are complex and varied "fitness" descriptions, not numeric values. Trajectories are not shown.

4 Simple and Complex Fitness Relations

In the simplest case a design either fits or does not fit a niche. More generally the relation is more complex, and different designs may fit the same niche to different degrees. Besides being simply better or worse, designs may fit a niche in different ways – e.g. different combinations of speed, robustness, flexibility, etc. The different styles of arrows in Figure 1 are intended to indicate this.

If the fitness of designs solving a particular problem vary only in degree, then the search for a solution can be thought of as the search for a location in design space where the fitness value is highest. This is the familiar notion of a fitness landscape. A learning mechanism or an evolutionary process might pursue a trajectory in which the design is guided towards a fitness peak in the landscape. High peaks can be very hard to find.

We can now generalise this idea in a number of ways.

(a) Instead of a fixed niche determining the evaluation we consider a space of niches as well as the space of designs. A design can then be assessed in relation to many possible niches, so that it does not have only one fitness value. Kauffman [7, pp 221ff] allows for this by mentioning that the fitness values for particular designs, and therefore the landscape, can change if objects in the environment change. If those objects also have fitness landscapes then there are *coupled* fitness landscapes, each causing changes in the other. His notion of a fitness landscape changing corresponds to our notion of a design having different fitness descriptions in relation to different regions of niche space.

(b) Instead of each niche determining an evaluation function which yields a numeric fitness, or even a total ordering of designs, it may determine a collection of incommensurable criteria for assessing designs (like speed and error handling in a compiler, or protection from predators and from cold in a house). In general the comparison of a niche and a design will yield not a number but a description of the match ([8, 11]). In simple cases this could be a vector of values.

Sometimes there is a partial ordering of the descriptions, and sometimes not even that, because there is no way to combine the different dimensions of comparison. Design A might be better than B in one respect, B better than C in another and C better than A in a third. (Compare consumer reports.) Engineers are very familiar with such tradeoffs between designs.

(c) Different designs and different niches can vary in complexity. So also do the descriptions of fitness. Assessment of a spelling checker will be much simpler than assessment of an operating system.

(d) If fitness values are non-scalar, trajectories no longer lead uphill, downhill or horizontally. A path can lead to improvements in some respects and degradation in others. "Selection" becomes a problematic notion.

(e) Having separate fitness values allows different sorts of selection to occur in sequence, simplifying the evolutionary design task by decomposing it, as an engineer might. When the changes required to improve different features are not independent there may be no useful decomposition. The "divide and conquer" approach is not always applicable: sometimes a creative new design is needed.

(f) If identical individuals inhabit slightly different niches (e.g. because of different social roles or different neighbours) reproductive success might be favoured by different traits. E.g. in some farming communities physical strength may be more important than intelligence, whereas in a nearby industrialised region intelligence is more useful for acquiring resources to raise a family. Thus different e-trajectories can be explored in parallel within a population exposed to different niches. Functional differentiation within a social system can accelerate this. Since motivation and performance are linked we can expect diversity of tastes as well as abilities. This may explain why we have individuals both able and willing to be concert pianists, steeplejacks, brain surgeons, etc.

(g) Since designs have complex structures, a niche can change simply because of a change *within* a design, without anything changing in the environment. E.g. a change which increases running speed may alter energy requirements. So some aspects of a design determine requirements for other aspects. What you need is partly determined by what you've got. This can generate positive feedback loops driving designs along e-trajectories without any environmental changes.

(h) The effects of niche pressures change in character when organisms develop cognitive abilities which enable them to recognize their own and others' needs and abilities. If they can assess in advance the relevance of different physical characteristics or behaviours to filling those needs, then improvements in a useful trait may be selected both directly through differing abilities to provide for offspring, and indirectly through recognition of the trait by potential mates. Eugenic social policies are similar. Cognitive abilities can also influence co-evolution: if predators can tell in advance which prey are easier to catch, they can select victims in a herd. Thus recognition of signs of weakness can accelerate the elimination of weaker traits. So cognitive processes in organisms can make evolution and co-evolution more like a process of design.

From the standpoint described here, genetic algorithms which use a scalar "fitness function" are simply a special case. Moreover, in artificial evolution the designer often adds a separate selection function which uses the output of the fitness function. In natural evolution (and some

Alife scenarios) selection and fitness are related in more subtle and varied ways.

We have seen that when organisms have cognitive abilities this can make evolution more like a design process. Perhaps we should think of the biosphere as a sort of super-organism struggling to develop itself and through the intelligence of its products becoming more like a designer.

5 Dynamics, Discontinuities and Inhomogeneities

Since niches and designs interact dynamically, we can regard them as parts of virtual machines in the biosphere consisting of a host of control mechanisms, feedback loops, and information structures (including gene pools). All of these are ultimately implemented in, and supervenient on physics and chemistry. But they and their causal interactions may be as real as poverty and crime and their interactions.

The biosphere is a very complex abstract dynamical system, composed of many smaller dynamical systems. Some of them are evanescent (e.g. tornados), some enduring but changing over diverse time scales (e.g. fruit flies, oak trees, ecosystems). Many subsystems impose constraints and requirements to be met or overcome by other subsystems: one component's design is part of another component's niche. Through a host of pressures, forces and more abstract causal relations, including transfer of factual information and control information, systems at various levels are constantly adjusting themselves or being adjusted or modified. Some of the changes may be highly creative, including evolution of new forms of evolution, and new mechanisms for copying and later modifying modules to extend a design.

These ideas may seem wild, but they are natural extensions of ideas already accepted by many scientists and engineers, e.g. [7, 2].

Discontinuous e-trajectories. Both design space and niche space have very complex topologies, including many discontinuities, some small (e.g. adding a bit more memory to a design, adding a new step in a plan) some large (adding a new architectural layer, or a new formalism). Understanding natural intelligence may require understanding some major discontinuities in the evolutionary history of the architectures and mechanisms involved. This in turn may help us with the design of intelligent artefacts.

I suspect discontinuities in design space occur somewhere between systems that are able merely to perform certain tasks, and others which can use generalisations they have learnt about the environment to create new plans, i.e. between reactive and deliberative architectures. Discontinuities in e-trajectories can occur when an old mechanism is copied then modified: e.g. a mechanism which originally associates sensory patterns with appropriate responses could be copied then used to associate sensory patterns with predicted sensory patterns or with a set of available responses.

Discontinuities might also be involved in the evolution of the "reflective" abilities described below: not only being able to do X but having and being able to use information on how X was done, or why X was done, or why one method of doing X was used rather than another. (Compare [17, 20].) What sorts of niche pressures in nature might favour such e-trajectories is an interesting biological question.

Design space and niche space are "layered": regions within them are describable at different levels of abstraction and for each such region different "specialisations" exist. Some specialisations of designs are called implementations. The philosopher's notion of "supervenience" and the engineer's notion of "implementation" (or realisation) seem to be closely linked, if not identical.

Both are inhomogeneous spaces: local topology varies with location in the space, since the minimal changes possible at various locations in the same space can be very different in type and number. Consider designs of different complexity: there are typically more ways and more complex ways, of altering a complex design than a simple design. So they have neighbourhoods of different structures. By contrast, in most multi-dimensional spaces considered by scientists and engineers (e.g. phase spaces), each point has the same number of dimensions, i.e. the same number and the same types of changes are possible at all points (unless limited by equations of motion).

Discontinuous i-trajectories. A system which develops, learns or adapts changes its design. I-trajectories, like e-trajectories can be discontinuous (e.g. cell division) and link regions in inhomogeneous spaces. The most familiar examples are biological: e.g. a fertilised egg transforming itself into an embryo and then a neonate. In many animals, including humans, the information processing architecture seems to continue being transformed long after birth, and after the main physiological structures have been established: new forms of control of attention, learning, thinking, deliberating, develop after birth. Ontogeny may partly recapitulate phylogeny: but cultural influences may change this.

Humans follow a very complex trajectory in design space throughout their lives. A good educational system can be viewed as providing a trajectory through niche space which will induce a trajectory in design space in self-modifying brains. A culture provides a set of developmental trajectories.

In general, following a trajectory in design space also involves a trajectory in niche space: the niches for an unborn foetus, for a newborn infant, a schoolchild, a parent, a professor, etc. are all different. Moreover, an individual can instantiate more than one design, satisfying more than one niche: e.g. protector and provider, or parent and professor. To cope with development of multi-functional designs we can include *composite niches* in niche space, just as there are composite designs in design space.

6 Trajectories for Virtual Machines in Software Systems

A distinction between i-trajectories and e-trajectories can be made for evolvable software individuals inhabiting virtual machines. A word processor which adapts itself to different users may or may not be capable of turning itself into a compiler through a series of adaptations. As in nature, there may be e-trajectories linking software designs that are not linked by i-trajectories.

Whether an e-trajectory exists from one software design to another in an artificial evolutionary system depends on (a) whether there is a principled way of mapping the features of the designs onto genetic structures which can be used to recreate design instances via an instantiation (ontogenetic) function, and (b) whether the structures can be manipulated by the permitted operators (e.g. crossover and mutation), so as to traverse a trajectory in "gene space" which induces a trajectory in design space via the instantiation function. Whether some sort of eval-

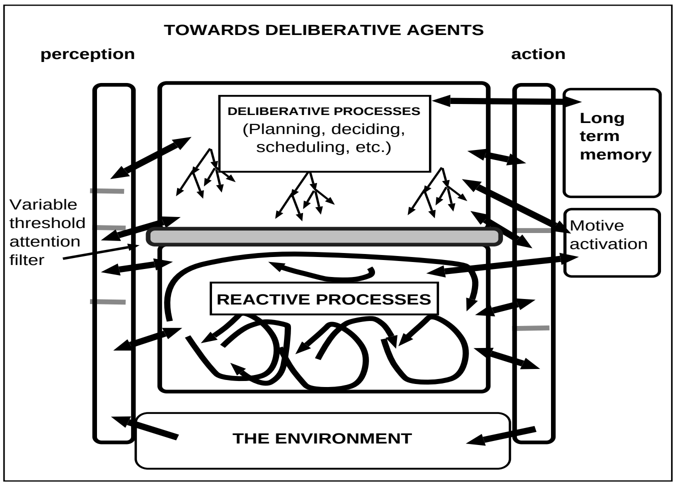

Figure 2: A hybrid reactive and deliberative architecture. A global "alarm" mechanism could be added. See text.

uation function or niche pressure can cause the traversal to occur is a separate question [10]. E-trajectories can exist which our algorithms never find.

7 Evolution of Human-like Architectures

We have argued in [14] and elsewhere (*contra* Dennett's "intentional stance") that many familiar mental concepts presuppose an information processing architecture. We conjecture that it involves several different sorts of coexisting, concurrently active, layers, including an evolutionarily old "reactive" layer involving dedicated highly parallel mechanisms each responding in a fixed way to its inputs. These may come from sensors or other internal components, and the outputs may go to motors or internal components, enabling loops. Some reactive systems have a fixed architecture except insofar as weights on links change through processes like reinforcement learning. Insects appear to have purely reactive architectures implementing a large collection of evolved behaviours. As suggested in [18], sophisticated reactive architectures may need a global "alarm" mechanism to detect urgent and important requirements to override relatively slow "normal" processes. This can interrupt and redirect other subsystems (e.g. freezing, fleeing, attacking, attending).

A hybrid architecture, as shown in Figure 2, could combine a reactive layer with a "deliberative" layer which includes the ability to create new temporary structures representing alternative possibilities for complex future actions, which it can then compare and evaluate, using further temporary structures describing similarities and differences. This plan-construction requires a long term memory associating actions in contexts with consequences. After creating and selecting a new structure the deliberative system may execute it as a plan, and then discard it. Alternatively it may be able to modify itself permanently by saving some or all of the structure for future re-use. In humans the reactive architecture seems also to be extendable by re-use of plans, e.g. learning car driving or language comprehension may create new reactions.

As before, a global alarm mechanism may be needed for coping with dangers and opportunities requiring rapid reactions. In mammals this seems to involve the limbic system, and emotional processes [3, 6].

A deliberative mechanism will (normally) be discrete, serial, and therefore relatively slow, whereas a reactive mechanism can be highly parallel and therefore very fast, and may include some continuous (analog) mechanisms, possibly using thresholds. Resource limits in deliberative mechanisms may generate a need for an attention filter of some kind, limiting the ability of reactive and alarm mechanisms to interrupt high level processing.

By analysing tradeoffs we may be able to understand how niche-pressures can lead to development of combined, concurrent deliberative and reactive architectures in organisms.

Everything that can be done by a hybrid architecture could in principle be done by a suitably complex reactive architecture e.g. a huge, pre-compiled lookup table matching every possible history of sensory inputs with a particular combination of outputs. However, pre-requisites for such an implementation may be prohibitive: much longer evolution, with more varied evolutionary environments, to pre-program all the reactive behaviours, and far more storage to contain them, etc. For certain agents the universe may be neither old and varied enough for such development nor big enough to store all the combinations required to match a deliberative equivalent with generative power. Perhaps evolution "discovered" this and therefore favoured deliberative extensions for some organisms.

A deliberative mechanism changes the niches for perceptual and motor mechanisms, requiring them to develop new layers of abstraction, as indicated in Figure 2. Likewise, development of new, higher level, abstractions in perceptual and motor systems may change the niches for more central mechanisms, e.g. providing new opportunities for learning and simplified planning.

Meta-management. Reflection on and retrospective evaluation of actions can often lead to future improvements. This is also true of *internal* actions. Thus besides abilities to perceive the environment and how external actions change it, there is a use also for internal self-monitoring, self-evaluation, self-modification (self-control) applied to *internal* states and processes. This could explain the evolution of a third architectural layer, as indicated in Figure 3.

Sensory "qualia" arise in self-monitoring mechanisms with access to intermediate sensory information structures not normally attended to. Different kinds of sensory qualia depend on different perceptual abstraction layers. Such "self-knowledge" is distinct from normal perception providing knowledge about the environment. "Meta-management" capabilities produce other sorts of qualia related to thinking processes, deliberation, desires, etc.

Robots with these capabilities might begin to wonder how their mental processes are related to their physical implementation, just as human philosophers do. Some of them, not fully understanding the notion of virtual machine functionality and the varieties of forms of supervenience, might even produce spurious but convincing arguments that they have conscious processes which cannot be explained by or fully implemented in physical processes. They may wonder whether humans are actually *zombies* with all the behavioural capabilities of conscious robots, but lacking their consciousness. I believe this solves the so-called "hard" problem of consciousness, see [1]. (Earlier work exploring these ideas can be found in [12, 13, 14, 16, 17, 19])

Such agents (with a combination of reactive, deliberative and self-management sub- architectures) may combine to form social systems. Questions about trajectories in design space and niche space arise for social systems also. Human social systems develop information and

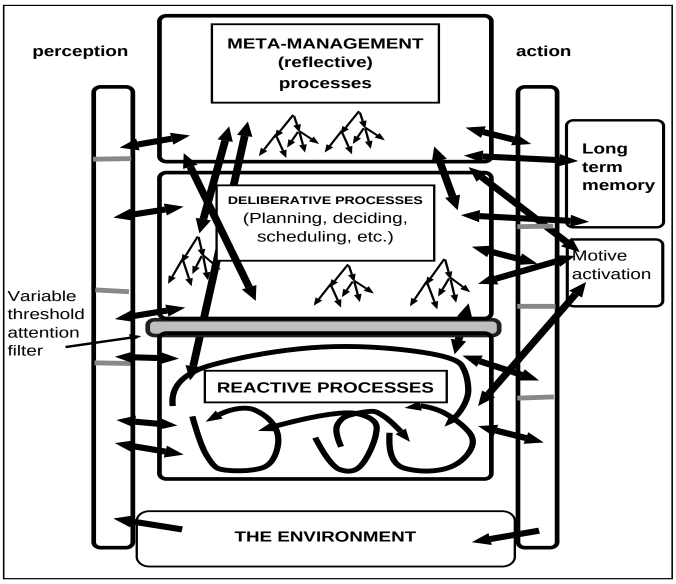

Figure 3: A meta-management layer adds self monitoring, self-evaluation, etc. Three classes of emotions correspond to alarm mechanisms (not shown, to save clutter) acting on the three layers.

rules which are transmitted to individuals, including rules that control meta-management (e.g. through guilt).

Accounting for all this needs a theory embracing computer science, theoretical biology, AI, psychology, neuroscience, anthropology, sociology, etc. Good software engineers and AI researchers are beginning to develop new intuitions about some of these things and it should be possible to find mathematical constructs that capture those intuitions, as computer science follows software engineering.

8 Further Work

We need to find more precise ways of describing architectures, designs, niches and their causal interactions, to improve on the high level concepts used only intuitively at present. This will involve both abstracting from domain specific details, so as to replace empirical concepts with mathematical concepts, and also enriching our understanding of the details of the processes, so that we can characterise and model the dynamics.

If the intuitive notions of niche, genotype etc. in biology can be made sufficiently precise to enable us to understand precisely the relationships between niches and designs for organisms, this may provide a better understanding of the dynamics and trajectories in biological evolution, including the evolution of evolvability.

This could lead to advances in comparative psychology. Understanding the precise variety of types of functional architectures in design space and the virtual machine processes they support, will enable us to describe and compare in far greater depth the capabilities of various animals. We'll also have a conceptual framework for saying precisely which subsets of human mental capabilities they have and which they lack. Likewise the discussion of mental capabilities of various sorts of machines could be put on a firmer scientific basis, with less scope for prejudice to determine which descriptions to use. E.g. instead of arguing about which animals, which machines, and which brain damaged humans have consciousness, we can determine precisely which sorts of consciousness they actually have.

We could also derive new ways of thinking about human variability and the causes and effects of mental illness, brain damage, senile dementia, etc. This could have profound practical implications.

Acknowledgements

The ideas reported here have been influenced by discussions with Margaret Boden, Chris Complin, Luc Beaudoin, Stan Franklin, William Langdon, Brian Logan, Riccardo Poli, Paul Marrow, Ian Millington, Rosalind Picard, Andy Pryke, Tim Read, Douglas Watt, Ian Wright, among many others.

References

[1] David J Chalmers. *The Conscious Mind: In Search of a Fundamental Theory*. Oxford University Press, New York, Oxford, 1996.

[2] Jack Cohen and Ian Stewart. *The collapse of chaos*. Penguin Books, New York, 1994.

[3] Antonio R Damasio. *Descartes' Error, Emotion Reason and the Human Brain*. Grosset/Putnam Books, 1994.

[4] Randall Davis. What are intelligence? and why? *AI Magazine*, 19(1):91–110, 1998. (Presidential Address to AAAI96).

[5] D.C. Dennett. *Kinds of minds: towards an understanding of consciousness*. Weidenfeld and Nicholson, London, 1996.

[6] Daniel Goleman. *Emotional Intelligence: Why It Can Matter More than IQ*. Bloomsbury Publishing, London, 1996.

[7] Stuart Kauffman. *At home in the universe: The search for laws of complexity*. Penguin Books, London, 1995.

[8] B. Logan and A. Sloman. Agent route planning in complex terrains. Technical Report CSRP-97-30, University of Birmingham, School of Computer Science, 1997.

[9] Robin Milner. Semantic ideas in computing. In Ian Wand and Robin Milner, editors, *Computing Tomorrow: Future research directions in computer science*, pages 246–283. Cambridge University Press, 1996.

[10] R. Poli and B. Logan. On the relations between search and evolutionary algorithms. Technical Report CSRP-96-07, School of Computer Science, The University of Birmingham, March 1996.

[11] A. Sloman. How to derive "better" from "is". *American Phil. Quarterly,*, 6:43–52, Jan 1969.

[12] 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.

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

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

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

[16] A. Sloman. Towards a general theory of representations. In D.M.Peterson, editor, *Forms of representation: an interdisciplinary theme for cognitive science*. Intellect Books, Exeter, U.K., 1996. ISBN: 1-871516-34-X.

[17] A. Sloman. What sort of control system is able to have a personality. In Robert Trappl and Paolo Petta, editors, *Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents*, pages 166–208. Springer (Lecture notes in AI), Berlin, 1997. (Originally presented at Workshop on Designing personalities for synthetic actors, Vienna, June 1995).

[18] A. Sloman. Damasio, descartes, alarms and meta-management. In *Proceedings International Conference on Systems, Man, and Cybernetics*. IEEE, 1998.

[19] A. Sloman. What sort of architecture is required for a human-like agent? In Michael Wooldridge and Anand Rao, editors, *Foundations of Rational Agency*. Kluwer Academic, 1998(forthcoming).

[20] A. Sloman and B. S. Logan. Architectures and tools for human-like agents. In Frank Ritter and Richard M. Young, editors, *Proceedings of the 2nd European Conference on Cognitive Modelling*, pages 58–65, Nottingham, UK, 1998. Nottingham University Press.