The Well-Designed Young Mathematician

Aaron Sloman*

School of Computer Science, University of Birmingham, UK

Abstract

This paper complements McCarthy's "The well designed child", in part by putting it in a broader context, the space of possible well designed progeny, and in part by relating design features to development of mathematical competence. I first moved into AI in an attempt to understand myself, especially hoping to understand how I could do mathematics. Over the ensuing four decades, my interactions with AI and other disciplines led to: design-based, cross-disciplinary investigations of requirements, especially those arising from interactions with a complex environment; a draft partial ontology for describing spaces of possible architectures, especially virtual machine architectures, for behaving systems (including our precursors); investigations of varied forms of representation and how they are suited to different functions; analysis of biological nature/nurture tradeoffs and their relevance to future machines; studies of control issues in a complex architecture; and showing how the states and processes possible in such an architecture relate to our (simplified) intuitive concepts of motivation, feeling, preferences, emotions, attitudes, values, moods, consciousness, etc. In 1971 I thought working models of human vision could lead to models of visual/spatial reasoning that would help to support Kant's view of mathematics, against Hume's. This has not yet happened, but I am still exploring requirements for such models, partly motivated by the hypothesis that human mathematical abilities are a natural extension of abilities produced by biological evolution that are not yet properly understood, and have barely been noticed by psychologists and neuroscientists. Some aspects of our ability to interact with complex 3-D structures and processes extend Gibson's ideas concerning action affordances, to include proto-affordances, epistemic affordances and deliberative affordances. Some of what a child learns about structures and processes starts as empirical then as a result of reflective processes can be transformed to the status of *necessary* (e.g. mathematical) truths. These processes normally develop unnoticed in young children, but provide the basis for much creativity in behaviour, as well as leading, in some, to development of an interest in mathematics. We still need to understand what sort of (possibly self-extending) architecture, and what forms of representation, are required to make this possible. This paper does not presuppose that all mathematical learners can do logic, though some fairly general form of reasoning seems to be required.

Key words: Altricial *vs.* Precocial species and competences, Architectures, Development, Empirical *vs.* Necessary, Evolution, Forms of representation, Geometry, Hume, Kant, Mathematics, Nature-nurture tradeoffs, Philosophy of Mathematics, Self-monitoring

The final version of this article will appear in *Artificial Intelligence*, accessible at http://dx.doi.org/10.1016/j.artint.2008.09.004

After this paper was written, a presentation on virtual machines, expanding some of the topics of sections 3-6 was given at a conference in November 2008. The presentation is available here (PDF) http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#wpe08

Email addresses: A.Sloman@cs.bham.ac.uk (Aaron Sloman)
URL: http://www.cs.bham.ac.uk/ axs/ (Aaron Sloman)

Contents

1. Introduction: Motivations for doing AI

Some people work in AI because they hope to produce useful or at least entertaining or impressive machines, often tailored to a particular application area, e.g. a personal assistant, an air traffic controller, a machine controller, an automated designer, a game character, or a tutor of some kind. Some aim to produce machines with human-like capabilities, without caring whether the mechanisms used have anything in common with biological information-processing mechanisms (e.g. brain mechanisms), on the assumption that such machines will have many applications.

In contrast, the primary interest of some AI researchers is science. Some of them aim to model in detail various aspects of biological intelligence, e.g. producing a machine that uses mechanisms that are biologically plausible and which has, as a result, ant-like, or lobster-like, or human-like competences. It is not always noticed that such goals are extremely ill-defined, since human-like systems would include systems like infants, toddlers, brain-damaged humans, schizophrenics, bricklayers, architects, composers, poets, murderers, people with dementia, people with autism, mathematicians, conjoined twins and even politicians. Others who are interested in AI as science include the study of non-biological mechanisms that are capable of producing various kinds of competence, including mechanisms thought to be on the path to so-called "Human-level AI", an ill-defined goal for the reasons mentioned, though made a little more precise by McCarthy, who explicitly prefers to focus only on desirable and useful aspects of human intelligence, for instance in "The well designed child".

This paper is an attempt to complement his discussion, in part by putting it in a broader context, the space of possible well designed young "self-extending" progeny, and in part by focusing on a particular kind of development: the development of mathematical competence. The broader scientific goal includes understanding the space of possible *requirements* for behaving systems and how those requirements relate to the space of designs for information processing systems that satisfy different sets of requirements more or less well. We could call this broad science the "Informatics of Intelligence" (IoI?) – a superset of AI as currently understood by most people. (My own interest in AI has always been based on my interest in IoI, though the label is new.)

The concept of a virtual machine including virtual machines made of other, interacting, concurrently active virtual machines, some of which may be discrete dynamical systems others continuous dynamical systems, will be shown to play an important role in specifying some of the more sophisticated designs, including designs for young individuals that develop self-awareness and self-control, and extend their information-processing architectures.

That will also help to bring out some of the relationships between this kind of science, and the physical sciences, including showing how it is possible for mental phenomena to play a causal role in a world where all mental phenomena are implemented in physical mechanisms.

Much of McCarthy's work, including "The Well-Designed Child" contributes to the huge task of collecting sets of requirements that need to be met by various interesting kinds of machine (including animals of various kinds).

Additional requirements are presented below, in the context of a theory of what the space of problems is like. In particular, a largely unnoticed aspect of the ability of an intelligent young animal or robot to grow its own mind will be shown to provide the basis of key mathematical competences in humans, and possibly future robots. Insofar as the mathematical knowledge gained by such learners is both non-empirical and substantive (i.e. not definitional and not purely logical) this will provide a new kind of support for something like the

philosophy of mathematics proposed by Immanuel Kant, in opposition to David Hume and many analytical philosophers. It points to a form of learning that is different from, and in some ways more powerful than, the essentially statistical Bayesian learning that has come to dominate AI research recently.

2. A broader view of AI as science

The goal of replicating functionality of one biological species whose members vary enormously is ill-defined, as explained above. Moreover, its scientific value is limited, for two reasons.

First, you can't really claim to understand something on the basis of a working model replicating its behaviour, if you don't know what difference it would have made if various aspects of the model had been different. Knowing how the functionality would have changed had various features of the design been different requires investigation of at least a neighbouring space of possible designs, and how those designs relate to different sets of requirements. This requires understanding design tradeoffs in a space of designs and a space of niches. Designs can be described at various levels of abstraction. For complex information-processing systems there are important reasons presented in Section 4.1 for considering designs at various *virtual* machine levels.

Second, the focus of all attention on human competences effectively ignores the fact that humans are one species among many, with a great many similarities and differences that need to be explained. Those species all have evolutionary histories, and the individual members have developmental trajectories. There is a vast space of structures and processes whose relationships and dynamics need to be explained if we are to understand the world we live in, and if the concepts, formalisms and theories of AI (or IoI) are to be deemed successful they should be able to account for the evolution, development and competences of that whole gamut of possible organisms with many different kinds of intelligence, or at least an interestingly varied subset of the gamut.

Even if biologists and psychologists wish to restrict their attention to the actual organisms produced by evolution on our planet, engineers and philosophers are motivated to explore a richer space of possibilities, including both organisms that might have evolved but did not, and possible machines of various kinds that might be produced in the future. From that point of view, AI as science looks incomplete unless it includes the study of the whole space of possible sets of requirements ("niche space") and the whole space of possible designs for behaving systems ("design space") and the relationships between the two spaces, including the tradeoffs between design alternatives relating to particular sets of requirements. See Figure 1.

I have been pursuing this broader goal for many years, in the context of AI,[1], most recently in collaboration with a biologist interested in animal cognition[2] and also working with AI colleagues in an EU-funded robotic project.[3] Over the years I have referred to this as "the design-based approach". I now prefer McCarthy's less cumbersome label "the designer-stance", though I do not know whether he accepts the implied requirement to explore spaces of designs and sets of requirements.

[1] See, for example, [43], [45], [46], [47], [49], [50], [51], [53], [54], [57], [55], [56], [59], [62], [67], [68], [66], [69]

[2] Jackie Chappell: [64], [4], [65]. We have now joined primatologist Susannah Thorpe in an attempt to analyse cognitive requirements for orangutan arboreal locomotion using compliant supports.

[3] The CoSy project: http://www.cs.bham.ac.uk/research/projects/cosy/

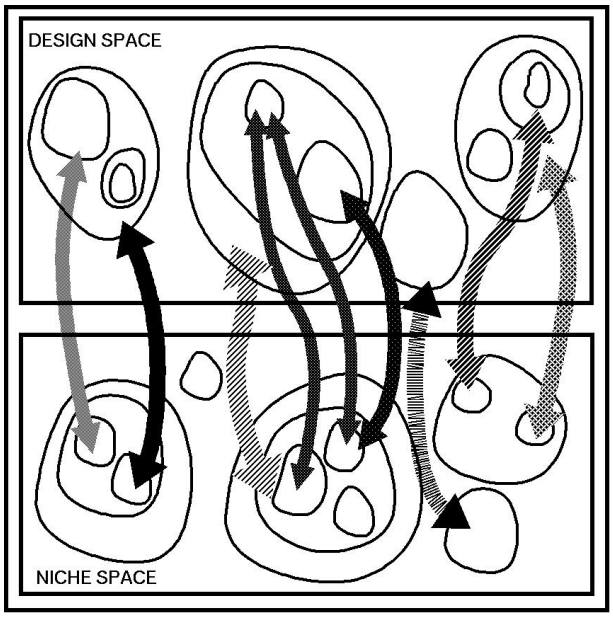

Figure 1: *There are many discontinuities in both design space and niche space. Relations between regions of design space and regions of niche space require structured descriptions, e.g. of what a design can and cannot do in various conditions, not just numerical measures. There are interacting trajectories of various sorts in both spaces, not shown here, including evolutionary, developmental and social trajectories. Biological trajectories are not continuous, but gaps are usually small. Trajectories in engineering labs can include large discontinuities, and unlike biological trajectories can include non-functional cases.*

3. Virtual machines and theories of everything informational

3.1. The potential scope of AI

Some would say the goal of studying those two spaces and their relations is far too ambitious, perhaps a symptom of megalomania. But it is no more so than the grand aim of the physical sciences to find a general way of understanding all the kinds of physical and chemical structures and processes that can occur, from the very smallest to the whole physical universe.

What physics does not include, however, because it lacks the required conceptual tools and theories, is the study of information and information processing mechanisms – which is why AI is needed. Even computer science as normally practised and taught is not general enough: it deals with mechanisms, structures, processes and formalisms with syntax that machines can manipulate. But computer science, except where it overlaps with AI, is not concerned with the semantic understanding required by a machine (or animal) in order to interact purposefully with the environment, as opposed to the semantics required by designers of machines who aim to produce machines that in turn produce the behaviour they (the designers, not the machines) desire.[4]

[4]See http://web.mit.edu/abyrne/www/intentionality.html. As explained there, John Searle distinguished *derived* from *intrinsic* intentionality, a distinction which John Haugeland expressed as *derivative/original*. Computer science could be described as concerned with giving machines derivative/derived semantic competences, not intrinsic/original semantic competences. This view is qualified in [44].

The relevant concept of information here is not the misleadingly named numerical "information" measure of Shannon and Weaver, related to channel capacities and signal statistics, but a notion of semantic content, that supports notions of truth, falsity, inference, consistency, and contradiction. Those notions are not parts of physics, though the work of physicists exemplifies them.

Some physicists attempt to bridge the gap between "meaningful" information processing and physical mechanisms. A well-known example is Henry Stapp, who believes that the only way to make sense of the equations of quantum physics is to allow some sort of consciousness to be involved in selecting between alternative possibilities. In an interview reported in 2006 he claimed *"... the core idea that the events in our streams of consciousness are two-way causally linked to events in the physical world lies at the intuitive heart of our daily dealings with reality"* [70]. What I'll say below about virtual machines is consistent with that, but Stapp wants to argue that according to classical physics it is impossible for mental phenomena to be causally efficacious, whereas it is possible in quantum physics. He is not making the standard naive assumption that quantum indeterminacy makes room for human free will, but puts forward arguments taking account of the details of the mathematics of quantum theory: *"We are dealing here with the sophisticated way in which mental intention influences quantum processes in the brain. Ideas do not simply push classically conceived particles around!"* and *"Why hang onto one of the most controversial aspects of a materialist worldview, namely the notion that the causal efficacy of our conscious efforts is an illusion, when quantum theory seems to say just the opposite, and even provides the technical means for implementing the causal efficacy of our efforts?" (op. cit.).*[5]

There are several problems with that attempt to unify physical and mental processes. The first is that it assumes that our ordinary obscure, and in some ways muddled, everyday concepts of mentality (e.g. "consciousness", "intention", "feeling") are suitable terms to use in combination with the mathematically precise theories of quantum physicists. Explaining why that is a dubious move would require an excursion into analytical philosophy.[6] A second problem is that giving human consciousness an essential role in determining the dynamics of physical processes leaves unanswered questions about how physical mechanisms operated on earth before the evolution of human-like animals.

The third, and most important, objection (which I have put to Stapp in correspondence on open discussion lists in the past, but which he ignores in his published papers) is that the concept of a virtual machine (VM), originally developed to support development of complex information-processing systems, provides an alternative, and more substantial, explanation of the relationship between mental and physical processes that is neutral between classical and quantum mechanics. I shall try to explain what virtual machines are and why they are so important. That will require making explicit facts that are obvious to many software engineers and often taken for granted, but not often stated, and certainly not usually included in text-books on AI, cognitive science, or philosophy of mind. (An exception is Edelman's [10].)

[5]For technical details, see Stapp's paper.

[6]Of the sort discussed in http://www.cs.bham.ac.uk/research/projects/cogaff/misc/logical-geography.html

3.2. Different concepts of "virtual machine"

We need to distinguish two main notions of a VM, followed by various sub-cases. The first main notion is of an abstract specification of a class of capabilities, as in "The Java VM", or "the Linux VM" which is a sort of mathematical object (actually a whole series of them, as new versions are developed).

The second main notion is of a process actually running, or runnable but currently dormant, in a machine, and doing or capable of doing things. I call these "active virtual machines" (even though they can include temporarily inactive components). The abstract specification does nothing: it is just an abstract structure, a sort of mathematical object, like a scientific theory, or a set of axioms, about which statements can be made and theorems proved. Active VMs, in contrast, do things, allow things to happen, and prevent things happening. An active VM may be an instance of a particular abstract VM. When a Java program runs, it uses an active VM that conforms to the abstract Java VM specification, but the active instance can make things happen while the abstract VM is no more causally active than a number. Different active Java VMs can exist on different computers, doing different things, and changing their states, while the abstract VM they all instantiate remains unchanged. Since I shall mostly only be talking about active VMs I shall not normally use the prefix "active".

3.3. Varieties of active virtual machine

We can distinguish different sorts of active VM. For example, some emulate a possible physical machine, whereas others do not: e.g. a virtual machine containing interacting 5-dimensional objects. Some active VMs, such as operating systems, provide a general platform within which other more specific VMs can run and interact. These are sometimes referred to as "system" or "system-level" VMs. Often a programming language is associated with a type of active system VM that includes a set of mechanisms that are available to support programs written in the language, for instance mechanisms for doing arithmetic, manipulating data-structures, constructing new procedures, and controlling procedure activations, or concurrent threads. Are there naturally occurring VMs analogous to that, e.g. VMs in brains to support particular forms of representation and operations on them?

A special subset of VMs could be thought of as "virtual reality" (VR) systems. They are simulations of something that could happen, though not necessarily something physical. These are useful in planning, predicting, reasoning, and explaining, and the notion of "imagining" covers many cases of this sort. Craik proposed in 1943 that evolution had produced such abilities in animals [7], with the advantage that they could be used to work out consequences of actions. There are many examples in Minsky's [30].

There are more fundamental kinds of VM that are not necessarily simulators. Examples include parsing VMs, image analysing VMs, calculating VMs, and VMs that perform logical inferences. We could call these "primary" VMs in contrast with simulation VMs which are in a sense secondary to what they simulate. Of course, being a primary VM does not rule out being used for simulation, since one primary VM can simulate another of the same type.

Some VMs allow more varied forms of process than others. One sort of variety comes from a fixed set of mechanisms whose activations can be combined in different ways, another from a VM that can grow new mechanisms, thereby acquiring new capabilities – e.g. building new procedures, plans, formalisms or connections between subsystems. This may be, but need not be, linked to physical growth, e.g. acquiring more memory, or another CPU to increase parallelism, or growing more neurons. There are also much simpler sorts of change in which parameters are modified in an otherwise fixed VM structure.

3.4. Biological virtual machines

I suspect biological evolution "discovered" the need for many types of VM that we have not yet studied, and also produced types of VM whose sophistication we have not yet matched, including varieties of self-constructing and distributed VMs. The variety and the sophistication result from the number and variety of design problems (i.e. sets of requirements, or niches) confronted in biological evolution, and the enormous versatility of the biological engines supporting evolution.

In order to achieve a better understanding of the full range of VMs found in living and artificial systems and the variety of VM components that can be assembled to produce that range, we probably need a new language for describing the components and ways of combining them to produce a systematic ontology of VMs. This could provide consistent and widely accepted terminology. Unfortunately at present AI researchers describe the VMs they design in an ad hoc way, often inventing their own diagrammatic conventions and labels. In descriptions of architectures, even widely used labels, like "reactive" and "deliberative" have no agreed definitions.[7]

We also need a systematic language (with an ontology) for describing sets of requirements (niches), including naturally occurring and artificial ones. This will require us to identify the problems evolution actually solved, instead of just assuming that the problems are obvious and easily describable, as often happens. McCarthy implicitly criticises an assumption made by many researchers when he writes *"Evolution solved a different problem than that of starting a baby with no a priori assumptions"*.

The importance of active, interacting, VMs is not widely understood, even by those who are well aware of virtual machines and use them every day. I shall try to explain why they are important for future developments in AI, as they seem to have been for biological evolution.

4. Why virtual machines are useful in animals and machines

Nowadays we routinely use physical computers to run virtual machines, whose properties and behaviours are very different from the properties of the physical machines in which they are implemented. This aspect of computation has largely been ignored by many who discuss the nature and significance of information-processing, including many, though not all, philosophers, psychologists and neuroscientists trying to understand the implications of information-processing concepts and mechanisms for their own fields.

4.1. Active, interacting VMs can reduce complexity for designers

An important reason for making use of VMs on computers is that they usually are much simpler (have fewer components and fewer possible states and state-transitions) and therefore easier to design, extend, understand, test, and debug than the physical processes that occur when they run.

When a word-processor or chess program starts up on a computer, that causes many changes in the memory of the computer and alters the sequence of machine instructions executed by the CPU (or CPUs), though many previously running programs will continue running. Exactly which sequence of instructions occurs will depend not only on which other processes were running, but which stages they have reached. Thinking about all the possible

[7] [59] suggests uses for new labels "proto-deliberative" and "fully deliberative".

combinations of low-level processes would be humanly impossible. Instead, use of virtual machines allows the designer to think about characters, words, fonts, paragraphs, layout of text, pages, etc., or about chess pieces, moves, board locations, threats, defences, captures, winning and losing in the second case.

If such a system needs to be modified it is usually the design at the VM level of operation that a programmer needs to change. This is possible because the existence of compilers, interpreters, general purpose subroutines, garbage collectors, operating systems, various communication protocols and high level interface specifications makes it possible for the system designer to leave the mappings between virtual machines and physical machines as someone else's problem. Describing a virtual machine requires use of an ontology related to its functions and domain of application, whereas designing and controlling the physical machine, or the digital circuit, requires a totally different ontology to be used. A subset of designers, e.g. those concerned with compilers and device-drivers need to work with two different levels of ontology.[8]

4.2. VMs reduce combinatorial complexity for system designers

If human programmers designing complex word-processors were required to think through all the processes at the level of digital electronics (as was required for the earliest mechanical and electronic computers running much simpler programs than current operating systems and application packages) they would have to consider all the different ways in which huge bit-arrays change as a result of paging operations, garbage collections, and context-switching in multi-processing systems, an explosively complex task because of the vast amount of variability in what happens.

Moreover, even if such a bit-level or physics level design worked, it would have to be changed if new software had to be installed to run at the same time, or if more memory were added to the computer, or a new kind of hard drive for swap space. Even bigger changes would be required if the programs had to run on a new kind of cpu, or using memory with a different size of addressable unit.

4.3. Benefits of problem-decomposition

By separating the problems of producing a program that uses a certain VM layer, i.e. specifying what structures and processes are to be created in that layer, from the problems of ensuring that a VM layer of that type is properly implemented on a particular physical machine, designers turn an intractable task into two tractable tasks.

In summary, formulating designs at a VM level makes it much easier for engineers to relate designs to the functionality required, that work across widely varying contexts and combinations of processing, and to take advantage of new technology to implement improved (faster, cheaper, more compact, more reliable) physical systems running previously designed systems. This depends on the use of at least two levels of design:

(a) designing and implementing a VM layer that supports certain classes of programs as long as mechanisms are provided for mapping virtual objects, events and processes into physical mechanisms and their states and processes;

[8] Alan Turing in effect demonstrated the possibility of virtual machines when he showed, in the 1930s, that some Turing machines can emulate others, though the ontological differences there are small. The practical importance of virtual machines did not emerge until some time later. I think Ada Lovelace understood the general principle a century earlier.

(a) designing working systems that can be implemented on that VM layer.

Task (a) includes designing formalisms, along with interpreters or compilers for the formalisms, collections of generally useful subroutines, operating system interfaces, device drivers, and physical interfaces to several kinds of device (e.g. keyboard, screen, hard drive, microphone, video camera, robot arm, temperature sensor, etc.), network protocols, transmission mechanisms, memory management mechanisms, and especially context-switching and saving mechanisms, that allow multiple processes to be time-shared on one processor. Often this kind of support for a VM layer is provided on several different physical machines, making the VM layer portable. However, portability may be partial – if not all features are available on all physical implementations, e.g. if one implementation of a VM layer takes advantage of physical features available only on some processors.

Not every new VM layer needs to be directly implemented in low level physical mechanisms: some are implemented using a pre-existing VM, such as the VM specified as a pentium processor or a sparc processor or a combination of processor and operating system. So VM layers can be stacked and combined in various ways, often using several layers of virtual machinery. This is unlike combining VMs that provide different functionality on the same layer, e.g. a parser and a planner.

The separation of design tasks means that different VMs (e.g. word-processors, mail-handlers, chess-programs, simulation packages, games) can be relatively easily implemented on a single VM layer (e.g. an operating system with a set of compilers), and they will continue to function if that VM layer is re-implemented in different hardware (provided the hardware has adequate speed and storage capacity for the task).

Often a VM layer is associated with a programming language (e.g. a Java, Lisp, C++ or Prolog virtual machine layer), though higher level VMs implemented in a programmable VM need not themselves be programmable, or if they are programmable they may use an application specific programming language.

We could call the use of VM layers *vertical modularity*. This is different from *horizontal modularity*, i.e. the separation into distinct modules that may or may not coexist. Both vertical and horizontal modularity support "separation of concerns" for designers, maintainers, and so on.

4.4. Layered biological virtual machinery

It seems that long ago biological evolution started making use of separation of improvements in the *design and implementation* of new VM layers and improvements in *use* of existing VMs. Obvious examples are genomes producing individuals that can learn many things, and the evolution of species in which different cultures can develop. There may be more subtle and unobvious examples in varieties of individual learning. The VM layers produced in organisms are implemented in very different physical mechanisms from VM layers that run on computers. Not much is known about how most of the abstract VMs are implemented, e.g. the ability to be puzzled, the ability to form explanatory theories, the ability too take in many abstract features of complex visual scenes, and many more. Neither is it known whether computing machines are capable of replicating all brain functions in great detail.

Virtual machinery provides a particular kind of benefit not for designers, but for the working systems themselves, as some engineers are beginning to appreciate, e.g. [6] (mentioned below). This will now be explained.

4.5. Benefits for individual machines, or animals, of using VMs

It is rarely noticed that "vertical modularity" or "vertical separation of concerns" is not just an issue for the engineering process of designing and debugging complex systems: it is also important for any system that needs to monitor and control *itself* at run time, as machines using schedulers, memory-management systems, file-access control systems, and many others have to do, and as sophisticated AI systems will increasingly need to do. So, not only is the use of VMs useful for engineers who design and build information-processing systems: it can also be important for sub-systems *within* the machine, that monitor and control other systems – for example a scheduler that ensures fair allocation of resources to different processes, or a file-system manager that monitors and constrains reading and writing of files, or a self monitoring program that can find and fix bugs in a running system.

These are examples of systems that need self-awareness and self-control. Self-awareness requires mechanisms that can observe and describe what is going on within the system, and possibly form hypotheses about patterns or regularities in the internal processes. Self-control requires mechanisms that can change what is going on, possibly using such patterns to predict consequences of changes. The forms of representation used by such mechanisms in computers (e.g. logic, diagrams, flow-charts, etc.) are not discussed here: that they represent structures, events, processes and generalisations about what is going on in the machine is all I am concerned with. Compare McCarthy in [23] and Kennedy in [20]: although neither says so, they both assume that the contents of self-observation are not the physical contents of the machine, but contents of virtual machines implemented in the physical machinery.

The point is made explicitly by Clark *et al.* in [6]: the authors propose research on *"a sort of network that can assemble itself given high level instructions, reassemble itself as requirements change, automatically discover when something goes wrong, and automatically fix a detected problem or explain why it cannot do so"*. They propose a new construct, *"the Knowledge Plane, a pervasive system within the network that builds and maintains high-level models of what the network is supposed to do, in order to provide services and advice to other elements of the network. The knowledge plane is novel in its reliance on the tools of AI and cognitive systems. We argue that cognitive techniques, rather than traditional algorithmic approaches, are best suited to meeting the uncertainties and complexity of our objective."*

For intelligent systems, that not only observe and modulate existing VMs running within themselves, but also have to extend VMs, or even add new layers of virtual machinery, the processes will be far more manageable if what is being built is a new VM using an old one, than if what has to be built is new digital or neural hardware, or a new way of connecting up existing hardware. If the mechanisms involved in self-monitoring, self-control, self-modulation, self-extension had to be designed to make physical changes instead of using VM levels, they would have to be re-designed whenever the hardware technology changed, and also as the set of programs running on the computer changed.[9]

[9]There are subtle issues about the differences in causal powers of VMs based on *compiled* and *interpreted* code, which I don't have space to discuss here. There are also subtle differences between the kind of monitoring and control that are possible when one part of a neural network continuously monitors and modulates another, using "hard-wired" connections for the purpose, and when one piece of active software monitors and modulates another, which usually requires the monitored subsystem to leave 'traces' of what it is doing, if it is a compiled program. An interpreter is less dependent on cooperation from the programs it runs. Specialised hardware is required to enforce monitoring of compiled programs that do not cooperate by leaving traces, memory management hardware being an example.

Finally, not only the development of individuals, but also the evolution of new biological species to provide new functionality, can be much easier if separate VM layers exist, than if the design of each species is a non-modular tangle of mechanisms. Something like this idea was proposed by Popper in [36], when he suggested that evolution might sometimes proceed by giving members of a species new desires that can be activated when appropriate, that make them want to do things that such animals previously did not do (e.g. wanting to get food from higher locations), which in turn could make it beneficial to evolve new behaviours to achieve those goals, which might then make it beneficial to evolve new physiology to support those behaviours (e.g. a longer neck, or more powerful muscles for jumping). Starting by providing new desires implies a level of structure at which desires can be added or modified. The same goes for adding new behaviours or modifying old ones.

5. Concurrent virtual machine processes in a serial computer

We have seen that separation of a complex design into different VM levels and different coexisting modules using those VM levels can have benefits in both engineering applications and in biological systems, and that this is particularly important in intelligent systems that need to understand what is going on and take decisions at run time, especially decisions that include self-monitoring, self-modulation and self-extension. However, the existence of coexisting interacting running VMs has additional implications.

5.1. Concurrent causal influences on a sequential machine

Even if there is a single time-shared CPU, so that instructions are executed one at a time (ignoring pipelines for now), the data-structures in memory that implement most of each active process continue to endure in parallel, and a process that is not in control of the CPU may still be influencing other processes because its data-structures are interrogated by running programs.

Moreover, there are often far more coexisting potential causes than actual ones, i.e. far more coexisting *dispositions* whose existence is important even when they are not activated. E.g. some risky behaviours may be attempted if a rapidly activated recovery mechanism is known to be *available* even if it is not doing anything when the decision is taken. So even in single-processor computers there is far more parallelism than most people realise, including coexistence of multiple dispositions and competences ready to be deployed if required. Interfaces to external devices or networks and additional processors can extend that parallelism.

We have seen that when several virtual machines run concurrently, the set of physical processes that can occur at the implementation level, is enormously complex and enormously varied, since what exactly goes on in all the central registers, in the buses transferring bit patterns around the system, and in the various hardware interfaces, depends on which *collection* of virtual machines happens to be running on the physical machine, on how they are phased in relation to one another, and also on the operation of schedulers and memory management systems. By abstracting away from most of that detail and representing only the important VM phenomena, we can understand and reason about behaviour of VMs whose invariant features and causal relationships allow huge variations in physical event sequences that implement them.

5.2. Evolution seems to have got there first

I conjecture that those advantages of virtual machines, and possibly others, were "discovered" and exploited by biological evolution long before human engineers thought of using them, and long before humans existed on earth, though details of virtual machines and how they are implemented are very different in biological systems. One example, apparently unique to humans, is the ability of brains using different human languages (e.g. English and Chinese) to use those languages to learn and use the same mathematical, physical, and geographical facts. The physical, neural processes involved in using different languages, will be different even when thought contents, calculations, inferences, or decisions, are the same. The brain processes implementing a thought about night falling or about the solution to an equation will be different in speakers of English and Chinese, yet the same VM event can occur in both.[10] I suspect that future research will show the importance of such separation of concerns, based on VMs, in many aspects of brain function, metabolism, individual development, transformations of a genome across generations and use of social or cultural virtual machines.

5.3. Why the physical sciences have explanatory gaps

Mechanisms based in a VM layer are all *implemented* in physical systems, but their description and analysis requires a science of information processing systems, including virtual machines that operate on virtual information structures (not *physical* symbols as claimed by Newell and Simon – e.g. see [31]). That science needs many specialisms concerned with special classes of structures, processes and mechanisms, for instance chess-playing VMs, parsing VMs, planning VMs, spelling-correcting VMs, equation-solving VMs, movement-controlling VMs, VMs interpreting sensory data, and many others, including VMs composed of collections of interacting VMs.

Describing the specialised VMs requires use of appropriate domain-specific ontologies, with their distinctive concepts (e.g. pinning a chess piece, a win, a draw, an illegal move, an inconsistency, an improved strategy, a spelling error, a syntactic error, etc.) which are not definable in terms of concepts of physics. (That indefinability claim will have to be defended in another paper.) Neither can the methods of the physical sciences produce measuring devices to detect occurrences of those concepts. In general, their occurrence can only be detected by interacting VMs, which also include events that can only be detected by interacting VMs.[11]

Dennett has suggested, e.g. in [9], that it is useful to take "the intentional stance" when dealing with other individuals who have beliefs, desires, etc. Some of what he writes seems to imply that attributions of mental states and processes are a matter of convenience rather than truth, and that they presuppose that the individual concerned is rational. (This is similar to Newell's notion of "the knowledge level" [32].) What I am suggesting is that far from merely being useful, it is a prerequisite of proper functioning of components of a complex information processing system that the various components take something analogous to the intentional

[10] Fodor's suggestion in [11] that every thought gets compiled into an innate "Language of thought" (LOT), presumably common to all humans, seems to imply the wildly implausible claim that evolution provided new-born human babies with an internal language that can express quantum mechanics, or even advanced scientific theories that have not yet been invented. I am not proposing that there is such an innate language: there are forms of development that extend semantic competence, not discussed here. See http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#models

[11] In [66] it was suggested that we can deal with philosophical puzzles about qualia by regarding the contents of such internal detections in certain sorts of machines as qualia.

stance to other parts of the system – that is, components of a complex virtual machine are designed to treat other components as sources of information or as users of information, a different matter from designing systems as producers and consumers of fuel, forces, energy, etc. A more detailed discussion would need to distinguish information users and senders from receivers and senders of signals, i.e. information-bearers. Many devices store, manipulate or transmit items such as bit-patterns, without having any need, or any ability, to treat them as expressing information, e.g. things that may be true or false. However, that competence is required for computers with conditional instructions [44].

5.4. Some philosophical implications

AI, broadly construed as the science of possible requirements for behaving systems and possible designs that meet those requirements, is a major part of the science of information-processing systems, a science that complements the physical sciences. I have argued elsewhere that this can revolutionise philosophy. For example, the relationships between *virtual machines* (which are real, and can have real effects, despite their label) and the underlying physical machines echo many of the philosophically puzzling features of the relationship between minds and brains: a point most philosophers have ignored, or failed to understand, largely because they are ignorant of the philosophically interesting features of the virtual machines they use every day. A notable exception is Pollock [35]. Dennett often refers to virtual machines, giving useful explanations for the benefit of philosophers and others, but denies that they really exist: he thinks talk about them is just a useful metaphorical fiction ([8], footnote 10). Metaphorical fictions cannot cause airliners to crash, banks to be robbed or even dot-patterns to move on screens.

5.5. The objective existence of virtual machine processes

The fact that some VMs are not definable or detectable using the concepts or tools of physics does not make the concepts subjective as some have supposed: it is not a matter of personal preference, or mere convenience, whether a portion of the world is or is not accurately definable as containing a certain sort of virtual machine.[12] The key point is that the concepts used are *relational* in a way that is hard to explain simply. What makes the VM subsystems in a complex VM what they are is the structure of the network of interacting causal connections, some internal, and some possibly external connections mediated by sensors and motors. (It is possible to read Gilbert Ryle as attempting to say this sixty years ago in [38], but lacking the concepts. Perhaps he should be credited as one of the inventors of the concept of a virtual machine.)

It is possible, in principle, though difficult in practice, to specify such a network of interacting subsystems by a large theory consisting of a collection of axioms with undefined symbols referring to various parts and aspects of the system. There will then be a difference between things that are and things that are not models of the collection of axioms, even though checking whether something is or is not a model may be difficult, e.g. if some of its parts are invisible virtual machines. Where the system is one that has been designed by human engineers who know how it works we can have more confidence in our descriptions of what is going on inside it, based in part on having access to the programs that it runs.

[12] Wilkes writes in [73]: "...are 'virtual machines' real machines with 'real' states? This sort of question deserves much more attention than it has received so far."

However it is notoriously difficult to determine that a system does exactly what its designers claim, and often "bugs" do not show up until a system has been running for a long time. However, the fact that bugs can be detected and fixed is evidence that the virtual machines running in the system do largely conform to the theory provided by designers, though it is not common for designers to express their theory axiomatically in a logical formalism. (It may become more common in future.)

The undefined symbols in the theory, referring to internal objects, relations, states, or processes in the active VM will be *implicitly* defined (or partially defined if the theory is extendable) by their roles in the theory. At present, our ontology for specifying roles is somewhat ad-hoc and ill-defined, especially roles thought to be referred to by terms of ordinary language, such as "desire", "belief", "intention", and "emotion". Some steps towards a more precisely defined ontology are taken in [67], including technical notions of "desire-like" and "belief-like" states. The theory may also refer to some causal links between the system and other (e.g. physical) entities, events and processes, for example in specifying input and output devices and processes of perception and action. This is partly analogous to the "meaning postulates", or "bridging rules" that relate theoretical concepts in the physical sciences to possible observations and experiments, but without thereby defining those terms, e.g. Carnap [3].

5.6. Formalisms for describing VMs

Which formalisms are adequate for expressing such theories is a research question, though various subsets have been explored by theoretical computer scientists. Any description of the physical details that leaves out the causal connections between events in the VMs will fail to describe important features of the system, including some of the true generalisations and counterfactual conditional statements about what would have happened if something specific had or had not happened: e.g. if a certain move in chess had not been made then the opponent would not have been forced to sacrifice the queen.

There is no finite statement in terms of behaviours of atoms or electronic circuits that is equivalent to such a statement. E.g. it might have been a game of postal chess using paper, or email, or some as yet undiscovered interplanetary mode of communication.

5.7. VMs with continuously varying components

It may turn out that some kinds of virtual machine cannot be implemented with sufficient accuracy in digital computers without adding analog components. Animals are made of interacting information-processing mechanisms of many sorts, including chemical (molecular), electrical and fluid-dynamical systems, whose functions scientists are still unravelling. This raises the question whether such mechanisms are capable of supporting types of VM that computer scientists have not yet thought of, perhaps including virtual machines for which conventional computers are inadequate.[13]

If a network of interacting virtual machines ("abstract dynamical systems") includes some parts that vary continuously, a formalism describing the network may need to include either components that vary continuously or suitable representations of continuous variation (e.g. differential equations, or logical sentences describing the changes).

[13]The aims of the UK Computing Research Committee's Grand Challenge 7 – "Journeys in Non-Classical Computation", address this: http://www.cs.york.ac.uk/nature/gc7/

5.8. Loose coupling or non-coupling with the environment

Behaviourist psychology assumed, in effect, that statements about what can happen when a VM runs refer to externally observable or physically measurable behaviour.

However, it is possible for some parts of a complex multi-component active VM machine to be disconnected from external sensors and motors, or to go through state changes that are too fast to be reflected in the input-output devices because of their limited bandwidth. In such cases we can talk about what would become externally visible if there were more components in the system, including higher bandwidth output devices.

There may be some VMs in a complex system that have to share output devices, so that some of the time one VM is controlling output, while others are doing things, but not producing any external effects. This implies that internal communication channels are not always switched "on". For example, sighting a dangerous predator could suppress the normal consequences of an internally detected need for food, e.g. moving to the location where the predator is and food also is. If all the connections were always turned on, external behaviours could be a mishmash of consequences of multiple internal processes with undesirable effects – e.g. the machine attempting to move in two directions at the same time. Clearly in a situation like that, a vector sum of influences can be far worse than a selection between influences (e.g. using a "winner-takes-all" competitive mechanism). So the ability to "disconnect" output channels of some subsystems in order to give unfettered control to high priority tasks is a useful design feature.

Likewise, some perceptual subsystems may do a lot of processing, whose results in certain circumstances have no influence beyond the relatively low-level processing subsystems, because doing anything else with them within the system in those circumstances would divert resources from more important and urgent tasks. So both "inward" and "outward" information-flows within the machine may be enabled or disabled according to context, without stopping the processing that occurs before the suppressed link. Similar remarks about changing connectivity apply to other connections between VMs within a larger system. The example of long term grief, which continues to exist while some of its influence is temporarily suppressed was analysed in some detail in [74].

All of this implies that reliance on behavioural tests for what an information processing system is up to can be misleading. The truth of counterfactual conditional statements linking internal states and processes to consequences they would have *if* conditions were appropriate may be difficult to assess experimentally if not all the relevant conditions (including internal conditions) can be manipulated in a laboratory. This not only implies that finding out how animals work can be difficult, but also that testing *complete* artificial systems can be difficult for the same reasons. In many cases, however, the components of such systems can be tested in isolation, before they are used in larger systems. In the case of biological systems, some of the components may have been first tested, and selected, in simpler evolutionary precursors, and then retained because their cost is relatively low and they can occasionally be extremely useful.

5.9. Virtual machines, not quantum machines

Not one of the design options discussed in this paper depends on the virtual machines being implemented in machines with quantum indeterminacy, as opposed to totally deterministic switching circuits. Arguments presented by Stapp and others (see Section 3.1) that suggest that only a quantum-mechanical machine allows mental processes to have real effects,

all depend on ignorance of the ways in which virtual machines (including running operating systems, schedulers, word processors, virus detectors, spam filters, etc. used by such people) can have real physical effects even though their powers, as described above, are of the mental sort, including making inferences, and other manipulations of structures with semantic content. All of this depends on the remarkable fact that a network of causal relationships between abstract virtual machine states and processes, can play a role in controlling physical processes, and in some cases can even resist external attempts to alter those processes – though such resistance has not yet matched the intransigence of HAL in the 2001 movie. That network of causal relationships is related to the truth of a network of conditional statements.

6. Counterfactual conditionals and virtual machine behaviours

6.1. Potentialities inherent in VMs

Earlier it was pointed out that whether a subsystem has effects or not can depend on what else is going on. Similarly when a subsystem does have effects, what the effects are can depend in complex ways on the context. A planning subsystem in a robot may, during construction of a plan, get information from a visual subsystem about whether a certain door is open, which influences a planning decision. If different information had been provided a different planning decision would have been taken. This is one among many different true conditionals describing internal processes in virtual machines.

At some point it may turn out that two goals that the whole system is pursuing are in conflict. Detection of the conflict by a monitoring VM can trigger a process in a conflict-resolution subsystem. At that point various other subsystems could influence that decision, including a subsystem that proposes a compromise, such as a modification of either goal or both, so as to remove the conflict, or a motivational subsystem that uses ethical or aesthetic or practical considerations to select one goal rather than the other.

It is also possible for a motivational or conflict-resolution subsystem to be undecided, causing the whole system to ask another individual for help, or to toss a coin – possibly an "internal coin". Collections of interacting VMs performing such tasks are what Minsky was (mostly) describing in [29, 30]. An early partially similar account can be found in [41, Chapter 6] (now online).

This account of what is involved in the existence of VM events and processes would seem to be fishily circular, were it not for the fact that systems are being built and used all the time that work on these principles. My colleagues, students and I have, for example, developed a toolkit used by students and researchers building such multi-component virtual machines[14] – which are nothing like the simple finite state virtual machines often used to explain philosophical functionalism, e.g. Block, in [2], and there are many other such toolkits (e.g. see [71]). Despite any appearance of fishiness due to untestability, the situation is not very different from the status of physical theories that postulate particles and processes whose behaviours may be undetectable by any currently available devices. Moreover, the existence and operations of genes in organisms is another example of virtual machinery referred to by scientific theories, where the virtual machines are assumed to be implemented in physical and chemical mechanisms even though many of the details are not yet understood.

[14] http://www.cs.bham.ac.uk/research/projects/poplog/packages/simagent.html

6.2. On feeling and being free to choose

The existence of myriad true counterfactual conditionals, concerning how things might have gone differently, is part of the basis for a self-monitoring machine to conclude that its choices are not compelled, so that it has a feeling of freedom to choose otherwise than it actually chose. In fact what happens may be determined by the totality of what is in the system at the time, and what information has been obtained about the environment, and that is as it should be: an intelligent agent's decisions should depend in systematic ways on its percepts, desires, preferences, hopes, fears, knowledge, expertise, and so on, except where they do not determine which of a set of equally acceptable or equally unacceptable options is available, in which case any mechanism, including a random generator, or possibly some socially influenced, arbitration mechanism that the individual may be unaware of, can make the selection.

Even when the total system is determined, there is no part of the system that *necessarily* has to do whatever it actually does, since other parts that influenced it work in such a way that they could have been in different states. (A more detailed exposition of this point is in [48].) In all those cases, what happens in one part of the system, or in the whole system depends in part on what happens in other subsystems, or how some part of the environment is perceived, and if the other subsystems had been in different states the results could have been different.

These really are causal interactions between events and processes in different subsystems, and things that interact causally are certainly things that exist – not just metaphorical ways of speaking, just as the poverty that can lead to crime is part of a social virtual machine that has causal powers – and poverty certainly exists, as a social phenomenon, even if it, like human mental processes, is ultimately implemented in a very large, indescribably complex, collection of physical mechanisms on our planet. Neither poverty nor virtual machine states and processes in computers, are merely convenient metaphors for complex physical interactions.

6.3. Machines that refer to the internals of other machines

It should be obvious that many of the problems of specifying precise meanings for both the language we use in talking about virtual machines and the language used for explaining how the physical world works in terms of unobservable entities and processes are examples of general problems regarding advanced scientific theories. The use of such "theoretical terms" with only loose and indirect connections to observations was noticed by philosophers of science long ago (e.g. [3]).

There are similar problems for young children learning about the environment and for future intelligent machines. As McCarthy notes, a young child needs to learn that objects in the environment can be made of different kinds of stuff, that react differently to various kinds of influences. Simple examples are different sorts of liquids, more or less runny, more or less sticky, etc., and different sorts of solids, more or less rigid, with different kinds of non-rigidity including plasticity, elasticity, fragility, etc. Often the child's immediately available means of perception do not suffice to distinguish such cases and experiments on objects are then required to identify their properties, e.g. how much weight a beam can support before it breaks, or whether a bent object will return to its original shape when the bending force is removed. McCarthy mentions the requirement for a child both to be able to learn about the atomic theory of matter, and that animate objects are to be understood in terms of their desires and actions. He could have mentioned many more examples.

Our ability to think about, and form theories about, virtual machines in systems we have not built ourselves is a special case of our ability to refer to unobservable features of the environment. It is hard to tell how much of this other animals can do, since learnt or evolved, reactions to complex objects in the environment need not be based on any theory of why those reactions are appropriate – though in that case the expertise will have definite limits, whereas someone with an explanatory theory can often derive new consequences about what will happen, or what can be done, in novel situations.

If future robots or other machines are to have human-like intelligence then they, like us, will need to be able to make use of concepts that cannot be defined in terms of what they can sense in the environment, but which have a role in theories that can be used in understanding and acting on the environment in creative ways, e.g. finding new ways to achieve or prevent physical states of affairs, or new ways to deceive or to persuade other individuals.

This use of theoretical terms referring to unobservable entities is incompatible with the philosophical theory of concept empiricism, rejected by Immanuel Kant over two centuries ago, and more firmly demolished by 20th century philosophers of science ([3] and others) because of the need to use theoretical concepts in physics, chemistry, biology, etc. However concept empiricism was recently resurrected in the form of "Symbol Grounding" theory [15] and has confused many AI researchers who lacked a philosophical education.[15] Designing a robot so that all its symbols are "grounded" as required by the theory is a way of making sure that it has serious limitations as an intelligent system.

6.4. Substantive scientific questions about VMs

All this leaves open the question: what sorts of virtual machinery are useful for various kinds of behaving system interacting productively with various environments? Despite the fact that AI has been going for over half a century, the problems that define the various sorts of niche that need to be dealt with by human-like robots have not yet been fully identified. The situation is not that we know what problems have to be solved and simply haven't solved them yet: rather the process of specifying the problems is still on-going. Many of McCarthy's discussions, e.g. of the well-designed child [24] and self-conscious machines [23] are contributions to analysis of requirements, interspersed with suggested partial solutions. Identifying the requirements is the more important task.

Theories like Stapp's, and most theories of consciousness proposed by philosophers and physicists, ignore the *detailed* requirements for scientific explanations of how animals, including humans, work. This is common among researchers who have no experience of attempting to design working systems yet propose explanations of how humans or animals work. Unfortunately, researchers who do have such experience make different mistakes because they also do not analyse the requirements in sufficient detail.

7. What sort of architecture is required?

One of the substantial questions is: what sorts of architecture are required for animals and machines that have to meet different requirements? That requires us to have a good way of thinking about both how requirements can vary and how architectures can vary. At

[15] A discussion of limitations of concept empiricism is available online in http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#models "Why symbol-grounding is both impossible and unnecessary, and why theory-tethering is more powerful anyway".

present many architectures in AI are assembled in what appears to be an ad hoc fashion, on the basis of hunches about what the problems are and how they can be solved, without any systematic overview of what the options are from which selections are made, and how variations in requirements should affect variations in designs. Instead, many papers simply present architectures without a deep analysis of the space of options: that may sometimes be acceptable for getting applications to work (if they work) but not for advancing our scientific understanding.

There are various ways of thinking about the space of designs and it is not yet clear how best to do this. One idea, inspired by chapter 25 of Nilsson's [33], combines two divisions of functionality among the information-processing components in an animal or machine, namely in terms of "towers" and "layers", as described in Section 10 of [63]. This is the basis of the CogAff schema proposed for describing architectures.[16]

The three (overlapping) "towers of functionality" are perception, action, and more central processing. There are different ways of thinking about "layers of functionality", and different researchers consider different numbers of layers. The CogAff schema uses layers corresponding roughly to evolutionary age in animals and computational abstraction in animals and machines: reactive layers evolved first, and mainly perform functions of responding to sensed (external or internal) situations and events by changing something (externally or internally). Deliberative layers, of varying sophistication, can consider and evaluate *possible* structures, events and processes including spatially or temporally composed possibilities [59]. Finally, layers of functionality which some have labelled "reflective", or "meta-management" layers, which I prefer now to label "meta-semantic", include the ability to represent, monitor, reason about, learn about and evaluate and act on other layers.[17] There are several ways in which the division into layers and towers is over-simple and does not allow for nearly enough sub-divisions. Analysing the tradeoffs properly will depend on looking closely at many of the detailed evolutionary and developmental transitions found in organisms as well as many types of engineering problem.

7.1. Layered dynamical systems

There are different ways of investigating requirements. One approach is to reflect on and analyse facts that are widely known but whose implications have not been noticed. E.g. what cognitive functions are involved in building a nest from twigs, as opposed to mud? Much of McCarthy's work involves such analysis. Another approach is to construct experiments that reveal features of human (or animal) competence that need to be explained, especially features related to common ways of interacting with the environment. Ambiguous pictures that flip between two different interpretations without anything changing in the image help to identify some requirements for visual perception: the differences between the two interpretations help to identify what needs to be represented in a visual system: it cannot be a matter of describing or classifying image contents, for example. A different sort of clue comes from considering how what is seen changes as the perceiver moves. When perceivers move there are changes in which surfaces are visible, aspect ratios, and other geometric and topological changes in the

[16] Progress report: http://www.cs.bham.ac.uk/research/projects/cogaff/03.html#200307

[17] One of the hard problems of meta-semantic competence is how to handle referential opacity – very briefly defined in http://en.wikipedia.org/wiki/Opaque_context. McCarthy [25] favours a solution expressed in a logical form, whereas I suggest specific architectural designs are needed. But that is a topic for another occasion.

information available to the perceiver, as well as subtle changes in highlights and reflected illumination. Sometimes what changes is that new possibilities and new impossibilities are perceived [61, 63].

Gibson's [13] has rightly been highly influential in both psychology and AI. He pointed out that the contents of what is seen are not just physical and geometrical facts about the environment, but also include *affordances* relevant to actions that could be performed and goals the perceiver could have. However, that is just a special subset of something more general, which includes perception of possible motions, and constraints on motions, that need not involve the perceiver's actions or preferences [52]. We could call that perceiving "proto-affordances", since proto-affordances can become affordances if they have features relevant to the perceiver's desires, preferences, needs and capabilities.

There is also perception of what information is or is not available, i.e. perception of "epistemic affordances". Changes in epistemic affordances can also be perceived, e.g. perceiving that something is being progressively hidden by something else. An important aspect of learning is discovering relations between action affordances and epistemic affordances: e.g. doing some actions may increase or decrease epistemic affordances, by changing the information available about some task. Michael Brenner has pointed out in discussion that some actions, or events occurring in the environment, can be seen to provide information relevant to an unfinished planning task, or information suggesting that an existing plan needs to be revised: these could be called "deliberative affordances".

7.2. Speed of information processing in humans.

Those changes of perceptual content, at various levels of abstraction, can happen very quickly in ways that no current computer vision system can match, and no known neuropsychological theories can explain.

An example demonstration is an experiment available online as a pdf file[18]. Viewers are shown a sequence of unrelated pictures at the rate of about one per second, and then asked questions at various levels of abstraction about what they have seen. This is intended to probe requirements for human-like vision that seem to be involved in going round a large opaque obstacle or looking out of a window for the first time, or moving something that blocks one's line of sight, and many other transitions involving suddenly perceiving a new complex scene that may require rapid and complex reactions (in contrast with simply ducking or moving sideways to avoid a rapidly approaching object, for example).

Thinking about possible ways of explaining how the demonstrated visual processing speed might be achieved, in contexts where details of what is seen may be unclear, noisy, or partially obstructed by dirty windows, rain, poor light, etc., led to the hypothesised multi-level network of dynamical systems (virtual machines that can rapidly change their state, subject to constraints of many kinds with many sources) depicted crudely in Figure 2. For more on this see [61, 63].

One function of the concurrency in such a system is to allow mixed top-down, bottom-up and middle-out processing. Another is to allow some VMs to have the role of monitoring and modulating others. I conjecture that a perceptual architecture of this general kind is

[18]See http://www.cs.bham.ac.uk/research/projects/cogaff/misc/multipic-challeng.pdf There are questions of varying degrees of abstraction after the pictures, probing the forms of processing that may have taken place when the pictures are viewed fairly quickly.

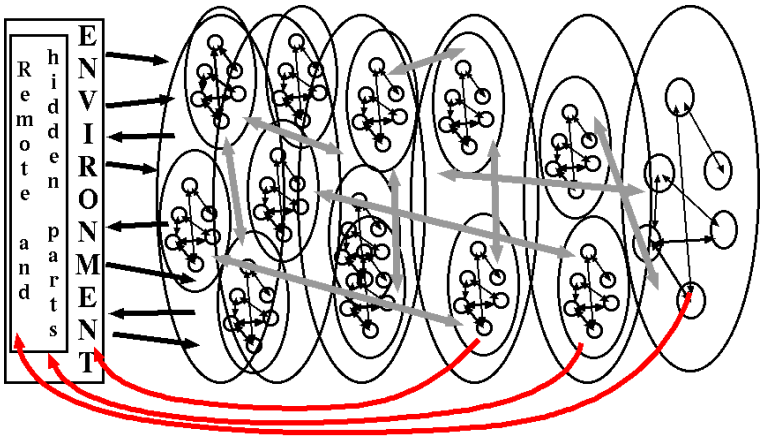

Figure 2: *A hybrid virtual/physical machine including multiple concurrently active dynamical systems (VMs) linked in a constraint network. Some of the components are closely linked to sensory and motor transducers. Some are more and some less remote from sensorimotor interfaces, with more remote subsystems able to refer to un-sensed aspects of reality. Some involve continuous dynamics, others discrete changes. At any time many components may be dormant, while capable of being rapidly activated (top-down, bottom-up or sideways) by the constraint network. The various perceptual (e.g. visual) subsystems are connected with more central processing subsystems and in some cases also with action subsystems, e.g. for reflexes, or active sensing mechanisms.*

connected with the development of mathematical competences alluded to in the title of this paper, as will now be explained below.

The rest of this paper focuses on a particular problem that not only still needs to be solved, but has not been generally recognised to exist. Although AI researchers have been working on how to design machines that can solve various classes of mathematical problems, they appear not to have noticed what it is about our environment that caused biological evolution to produce a species that developed certain sorts of mathematical competences as a result of the problems and opportunities for action afforded by that environment.

8. Animal intelligence and human mathematics

8.1. AI and philosophy of mathematics

My own original motivation for getting involved in AI, nearly 40 years ago, was not based on the grand vision described above: I was merely trying to understand myself. In particular, after a degree in mathematics and physics, I was seduced by philosophy, and worked on a DPhil thesis [39] defending Kant's view of the nature of mathematical discoveries in his *Critique of Pure Reason* [19], at least as I understood him. I tried to show that Kant was correct in arguing, against an opinion attributed to Hume, favoured by many analytical philosophers when I was a student, namely that mathematics was inherently trivial: a collection of tautologies in elaborate disguises. A variant of this was Bertrand Russell's assertion in [37]: "Mathematics may be defined as the subject in which we never know what we are talking about, nor whether what we are saying is true." I tried to defend Kant by showing that mathematical discoveries expand our knowledge, for example about spatial structures and about counting procedures, but not in the same way as empirical discoveries, such as the discovery that many substances expand on being heated.

About six years later I started learning about AI, as a result of getting to know Max Clowes [42], who introduced me to work done on machine vision in the 1960s (e.g. [18]) and especially the writings of Marvin Minsky (e.g. [26, 27] and other papers in [28]) and John McCarthy [22, 25]. Those papers should still be compulsory reading for students studying AI, cognitive science, theoretical psychology and philosophy, but alas fashions change and they have been forgotten or ignored by teachers in those fields. Some of Chomsky's ideas in [5], e.g. about generative power, the performance/competence distinction, and varieties of adequacy (extended by McCarthy and Hayes) were also important.

Although I did not agree with everything they wrote, the AI theorists transformed my way of thinking about philosophy, mainly by introducing me to the idea of an information-processing mechanism that could operate on many kinds of more or less abstract information structures, albeit implemented concretely in a physical machine – as explained in preceding sections.

However, around that time much of AI was focused on what went on *inside* intelligent machines, with the exception of a few robotic projects, such as Shakey at Stanford (http://en.wikipedia.org/wiki/Shakey_the_Robot) and Freddy in Edinburgh (http://www.aiai.ed.ac.uk/project/freddy/) [1], whereas I was sure that my ability to do mathematics, including logic and set theory, was connected with my ability to see things in the environment, or at least the mechanisms that support that ability and which could also function when my eyes were shut.[19]

In the thesis I had defended Kant merely by presenting and analysing examples of mathematical concepts and discoveries. As a result of learning about AI, I became convinced that if we could produce a machine that could make mathematical discoveries in something like the way I (and many mathematicians) did, that would help to show why Kant was right and his opponents wrong. However, unlike the majority of AI researchers at that time, e.g. McCarthy & Hayes in [22], I did not think that the robot would do all its mathematical reasoning using logic. On the contrary, I knew I could reason using diagrams (whether on paper or in my mind made no difference [58]) and in my first AI paper, at IJCAI 1971 [40], I argued that, for some purposes, spatial, diagrammatic reasoning might be more useful than logical reasoning, and just as rigorous, though in a different way. This got me an invitation to spend a year at Edinburgh University (1972-3) learning more about AI and especially learning to write programs (in POP-2 and Lisp). I later learnt that several other researchers had also thought spatial/diagrammatic reasoning important for AI, e.g. see [14].

8.2. Towards a mathematical robot

Several things became clear that were written up in a book a few years later [41] (now online). First, the kind of visual reasoning system I had in mind had to be embedded in a rich, multi-functional architecture containing many different sorts of mechanisms interacting with one another [41, Chapter 6], whereas most of the work in AI at that time seemed to be concerned with developing an algorithm to do one thing at a time, e.g. find a plan, find a proof, parse a sentence, recognise a pattern, answer a question, etc. Second, examining many examples of uses of vision showed that producing visual systems was a far more complex task than I had realised at first and would itself require visual sub-processes operating concurrently

[19] There have been outstanding blind mathematicians, of course. But their brains, like mine, were products of evolution in a species with sophisticated visual capabilities.

at different levels of abstraction, as explained in [41, Chapters 6 & 9]. However at that stage I did not appreciate the importance of concurrent perception of *processes* at different levels of abstraction, mentioned above in Section 7.1, and discussed further in [61, 63].

Third, as a result of watching a child both learning to count and discovering features of counting processes (such as that one way to answer "What number comes before N?" is to count up to N, and then use your memory of the last number uttered), I realised that many mathematical discoveries were discoveries about properties of *processes*, especially processes involving two discrete sub-processes performed in parallel and in synchrony, like pointing and reciting number names. Many applications of number concepts required the ability to generate such synchronised processes, but with different stopping conditions – e.g. stop when one of the processes cannot continue (no more marbles) or stop when some stage in one of the processes has been reached (fetching six marbles), as described in [41, Chapter 8].[20]

Fourth, it became clear that some discoveries made by a learner could start off empirical (e.g. discovering that counting a row of objects from left to right gave the same result as counting from right to left) but as the learner acquired a deeper understanding of what was going on such discoveries could be seen not to be empirical, but examples of necessary truths.

8.3. Transformation of epistemic status

This ability to make an empirical discovery then later realise that it is not empirical is, of course, not restricted to childhood. Many readers of this journal will have had experience of trying to solve puzzles in a certain way, then discovering after a while that their failures, initially discovered empirically, were provably *necessary* consequences of the relationship between the methods used and the structure of the problem. For example, if you have a rectangular slab of chocolate made of e.g. 6 by 8 squares and you try to break it into the individual squares by breaking one piece at a time, using a linear break, you can try different sequences of breaking actions but will always need 47 of them. After thinking about it you will see that that is an *inevitable* consequence of the structure of the problem and, moreover, for any array of squares the number of breaks is one less than the number of squares.[21]

Another example: try drawing a circle and a triangle on a plane and counting the number of contact points, where a contact point is either an intersection between an edge of the triangle and the circumference of the circle, or a vertex of the triangle lying on the circumference or an edge forming a tangent to the circle. You will find that there can be any number of contact points between 0 and 6, but no more, and that limit is independent of the size of the circle, and the size and shape of the triangle and its location in relation to the circle. Moreover, if the diameter of the circle is much less than the length of the shortest side of the triangle, it will not be possible to produce six contact points, and likewise if the diameter is much longer than the length of the longest side.

Programmers discover many more such relationships between structures and processes, some of them discovered empirically at first then later understood to be exceptionless, two familiar and useful examples being (a) the correspondence between a process of depth-first search in a tree or graph structure and a loop using a stack of options (last in, first out) and (b) the correspondence between breadth-first search and the use of a queue of options (first in, first out).

[20] http://www.cs.bham.ac.uk/research/projects/cogaff/crp/chap8.html

[21] Some qualifications and special cases are discussed in this presentation http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#math-robot

The claim that some of the discoveries turn out not to be empirical does not imply that we are infallible, or that future robots will have to be. In [21], Lakatos showed how mathematicians can produce fallacious proofs then later discover the bugs in the proofs and try fixing them in various ways. This will also be true of young children, intelligent robots and other humans: we can all make mistakes even when doing mathematics. So a good explanation of the capabilities under discussion will have to show why the methods used are fallible, how mistakes can be detected, and how, in some cases the mistakes can be corrected. Sussman in [72] presented an architecture for a self-extending self-debugging planning system, though with no understanding of the empirical/non-empirical distinction. In [34] Alison Pease attempted to model some simple examples, not connected with any ability to perceive and act on objects in a 3-D environment. There is still a long way to go.

8.4. Problems of being a baby, or toddler

The process of making empirical discoveries that can later be found to have a non-empirical, mathematical, basis is not restricted to artificial puzzles and activities of programmers and mathematicians. This seems to be a deep feature of the process of learning to interact with a structured spatial environment, though not all animals that do such learning seem to have the ability to make the transition from empirical to non-empirical understanding. Humans cannot do it from birth, though I don't yet know when the change starts, nor exactly what makes it possible, except that it seems to require an architecture that supports monitoring of processes while they are performed in order to discover features of those processes, and then construct an ontology and a theory that allows those features to be explained in a general way. Some conjectures that are relevant to this are in [4].

For example discovering that some things are rigid and impenetrable, allows many different generalisations to be derived. E.g. if rod made of a rigid and impenetrable material is in a corridor with parallel walls made of rigid impenetrable material and the length of the rod exceeds the distance between the walls, then the direction of the rod cannot be reversed by rotating it in a horizontal plane. It can be reversed by rotating vertically if the ceiling is high enough, or there is no ceiling. Otherwise it can be rotated by moving the rod into a big adjoining room, rotating it there and moving it back into the corridor. One important thing about this is that if a child understands the problem then she can tell that the colour of the rod is irrelevant, that various aspects of the shape of the rod, the texture of the walls, which hand is used to hold the rod, etc. are all irrelevant to what prevents the rotation. A purely statistical (e.g. Bayesian) learner would have to collect evidence for each new case.

A closely related result is that if two co-planar gear wheels made of rigid, impenetrable material are meshed and pivoted centrally, then if one is rotated the other must also rotate, in the opposite direction. What would happen if a third gear wheel was added, meshed with both of them?

In [12] many examples of learning about affordances in young children are described. It seems to be assumed by the authors, and by many psychologists that there are only two ways for a child to learn what can and cannot occur, namely either by trial and error, or by learning from someone else, either by imitation or being instructed. However, there is another possibility: working things out. Having an appropriate theory about what is going on allows a child to *work out* what must happen even in some situations that have never previously been encountered, e.g. where new shapes are concerned, whereas if the learning were purely empirical every new situation would have to be tested before predictions could be relied on. Compare Figure 3.

Figure 3: *Despite the low resolution, poor lighting, and noise in this image people can easily perceive a collection of objects with definite spatial relationships, even though what is perceived (including shapes, orientations, curvature, relative thickness, etc.) is not perceived with great precision. You can probably visualise various ways in which using your right hand, your left hand, both hands, your teeth, you could get the saucer onto the table, the cup on the saucer and the spoon in the cup. A challenge is to devise forms of representation that (a) are derivable from images despite poor image quality, and (b) have sufficient definiteness to allow actions to be planned and executed reliably. Initially a baby could not do this but a normal child will eventually develop both the ability to do it and to describe how to do it.*

8.5. Logical and non-logical forms of representation

In principle it would be possible to formulate theories about what is going on using logic and to derive all the discovered non-contingent truths from the theory, if the theory had enough axioms and inference rules. But it seems that humans, and presumably some other animals, can do such reasoning by *visualising* processes occurring subject to constraints. This seems to involve running a spatial modelling visual machine, although in complex cases it may have to be supplemented with physical diagrams and models. However, this does not imply that precise models of the scenes under discussion need to be created. In fact in some cases forms of representation seem to be used that express inconsistent contents, as shown by the Penrose triangle and Escher's drawings, which a model could not do.

Subject to those caveats, we need to produce visualisation processes that allow predictions or planning processes to be based fairly directly on what is seen, instead of having to first translate contents of perception to logic, make deductions, then translate back in order to know what to expect. I tried to make points like this 37 years ago in [40] but there still does not seem to be any AI system that convincingly illustrates these capabilities, although various partial capabilities have been implemented e.g. [17].

I suspect that the task is far more complex than it seems because new forms of representation are required and also new forms of self-monitoring architecture that can enable a robot to discover that some of its learnt generalisations are not merely empirical. I hope that drawing attention to this phenomenon will inspire some researchers to work on this who might otherwise pursue more popular objectives.

9. Challenges

A challenge for AI is to work out in more detail the requirements and develop working designs. For AI as science there needs to be systematic analysis of tradeoffs between various sets of requirements and possible designs.

A challenge for developmental psychology is to understand the role of the processes described above, in the development of young minds.

A challenge for biology is to explain how such abilities are related to biological advantages, and how the human genome (and perhaps other genomes) can encode the abilities to learn and develop as described here.

A challenge for educationalists is to develop teaching strategies, especially in mathematics, that relate more effectively to the mechanisms in young learners, and what those mechanisms can and cannot do at various stages of development.

A challenge for philosophers is to absorb tbe implications of all this for philosophy of mathematics. I think it provides at least a partial vindication of Kant [19], and in particular

> *There can be no doubt that all our knowledge begins with experience. For how should our faculty of knowledge be awakened into action did not objects affecting our senses partly of themselves produce representations, partly arouse the activity of our understanding to compare these representations, and, by combining or separating them, work up the raw material of the sensible impressions into that knowledge of objects which is entitled experience? In the order of time, therefore, we have no knowledge antecedent to experience, and with experience all our knowledge begins.*

But what starts empirical need not remain empirical, as we have seen.

9.1. Confusions about embodiment

A challenge for the interdisciplinary research community concerned with embodiment is to undo some of the damage done to research and education in AI and cognitive science by over-emphasising the role of embodiment in intelligence. What seems to be most important about embodiment (e.g. what drove the most significant evolutionary developments in mammal and bird cognition) is not the precise morphology of humans and other animals but rather the need to be able to perceive and interact with 3-D structures and processes (including manipulating, assembling and disassembling 3-D structures) and the need to be able to think about spatially located events, processes and entities in the past, remote spatial regions, and the future.

As noted in [60] the development of bodies with independently movable manipulators that could be used to assemble and disassemble 3-D structures of varying types and complexity added significant information-processing, representational, and ontological demands.[22] In contrast, much of the work on embodied cognition in robots has focused on the terribly narrow problem of learning about sensorimotor relationships, often in the context of a robot that is little more than a mobile point with a viewing direction.

Acknowledgements

This paper owes a considerable debt to the late Max Clowes for pointing me at AI, and to the writings of John McCarthy and Marvin Minsky, as well as other AI researchers. The work of Immanuel Kant first started me thinking along these lines over 50 years ago. More recently I have benefited from discussions with various colleagues including Margaret Boden, Michael Brenner, Alan Bundy, Jackie Chappell, Ron Chrisley, Nick Hawes, Mateja Jamnik, Manfred Kerber, Brian Logan, Dean Petters, Matthias Schlemmer, Susannah Thorpe, Jeremy Wyatt, Michael Zillich and collaborators in the EU-funded CoSy robotics project. Mary Leng got

[22] More examples are available in this web site http://www.cs.bham.ac.uk/research/projects/cosy/papers/#dp0601 Orthogonal Recombinable Competences Acquired by Altricial Species (Blankets, string, and plywood)

me thinking about mathematical knowledge again, after a gap of some years, by inviting me to give a talk in January 2008. I do not claim that anything in this paper is new or original, though I have not seen all the points brought together previously. There are probably many things I have read, learnt from, and forgotten.

References

[1] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone. A Versatile Computer-Controlled Assembly System. In *Proc. Third Int. Joint Conf. on AI*, pages 298–307, Stanford, California, 1973.

[2] N. Block. What is functionalism?, 1996. http://www.nyu.edu/gsas/dept/philo/faculty/block/papers/functionalism.html, (Originally in The Encyclopedia of Philosophy Supplement, Macmillan, 1996).

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

[4] J. Chappell and A. Sloman. Natural and artificial meta-configured altricial information-processing systems. *International Journal of Unconventional Computing*, 3(3):211–239, 2007. http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0609.

[5] N. Chomsky. *Aspects of the theory of syntax*. MIT Press,, Cambridge, MA, 1965.

[6] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski. A knowledge plane for the Internet. *Proceedings ACM SIGCOMM'03*, pages 3–10, August 2003. http://www.cs.utah.edu/classes/cs6935/papers/knowledge-plane.pdf.

[7] K. Craik. *The Nature of Explanation*. Cambridge University Press, London, New York, 1943.

[8] D. C. Dennett. Heterophenomenology reconsidered. *Phenomenology and the Cognitive Sciences*, 6(1-2):247–270, 2007. DOI 10.1007/s11097-006-9044-9.

[9] D.C. Dennett. *The Intentional Stance*. MIT Press, Cambridge, MA, 1987.

[10] Shimon Edelman. *Computing The Mind*. Oxford University Press, New York, 2008.

[11] J.A. Fodor. *The Language of Thought*. Harvard University Press, Cambridge, 1975.

[12] El. J. Gibson and A. D. Pick. *An Ecological Approach to Perceptual Learning and Development*. Oxford University Press, New York, 2000.

[13] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, MA, 1979.

[14] J. Glasgow, H. Narayanan, and B. Chandrasekaran, editors. *Diagrammatic Reasoning: Computational and Cognitive Perspectives*. MIT Press, Cambridge, MA, 1995.

[15] S. Harnad. The Symbol Grounding Problem. *Physica D*, 42:335–346, 1990.

[16] E. Jablonka and M. J. Lamb. *Evolution in Four Dimensions: Genetic, Epigenetic, Behavioral, and Symbolic Variation in the History of Life*. MIT Press, Cambridge MA, 2005.

[17] M. Jamnik, A. Bundy, and I. Green. On automating diagrammatic proofs of arithmetic arguments. *Journal of Logic, Language and Information*, 8(3):297–321, 1999.

[18] S. Kaneff, editor. *Picture language machines*. Academic Press, New York, 1970.

[19] I. Kant. *Critique of Pure Reason*. Macmillan, London, 1781. Translated (1929) by Norman Kemp Smith.

[20] C. M. Kennedy and A. Sloman. Autonomous recovery from hostile code insertion using distributed reflection. *Journal of Cognitive Systems Research*, 4(2):89–117, 2003.

[21] I. Lakatos. *Proofs and Refutations*. Cambridge University Press, Cambridge, UK, 1976.

[22] J. McCarthy. Programs with Common Sense. In *Proceedings Conference on the Mechanization of Thought Processes*, Teddington, 1958. Also in [28] and online http://www-formal.stanford.edu/jmc/mcc59.html.

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

[24] J. McCarthy. The Well Designed Child, 1996. http://www-formal.stanford.edu/jmc/child1.html.

[25] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of AI. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, Edinburgh, Scotland, 1969. http://www-formal.stanford.edu/jmc/mcchay69/mcchay69.html.

[26] M. L. Minsky. Steps towards artificial intelligence. In E.A. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 406–450. McGraw-Hill, New York, 1963.

[27] M. L. Minsky. Matter Mind and Models. In M. L. Minsky, editor, *Semantic Information Processing*. MIT Press, Cambridge, MA,, 1968.

[28] M. L. Minsky, editor. *Semantic Information Processing*. MIT Press, Cambridge, MA,, 1968.

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

[30] M. L. Minsky. *The Emotion Machine*. Pantheon, New York, 2006.

[31] A. Newell. Physical symbol systems. *Cognitive Science*, 4:135–183, 1980.

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

[33] N.J. Nilsson. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann, San Francisco, 1998.

[34] Alison Pease. *A Computational Model of Lakatos-style Reasoning*. PhD thesis, University of Edinburgh, Edinburgh, 2007. http://hdl.handle.net/1842/2113.

[35] J. L. Pollock. What Am I? Virtual machines and the mind/body problem. *Philosophy and Phenomenological Research.*, 76(2):237–309, 2008. http://philsci-archive.pitt.edu/archive/00003341.

[36] K.R. Popper. *Unended Quest*. Fontana/Collins, Glasgow, 1976.

[37] B. Russell. *Mysticism and Logic and Other Essays*. Allen & Unwin, London, 1917.

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

[39] A. Sloman. *Knowing and Understanding: Relations between meaning and truth, meaning and necessary truth, meaning and synthetic necessary truth*. PhD thesis, Oxford University, 1962. http://www.cs.bham.ac.uk/research/projects/cogaff/07.html#706.

[40] A. Sloman. Interactions between philosophy and AI: The role of intuition and non-logical reasoning in intelligence. In *Proc 2nd IJCAI*, pages 209–226, London, 1971. William Kaufmann. http://www.cs.bham.ac.uk/research/cogaff/04.html#200407.

[41] A. Sloman. *The Computer Revolution in Philosophy*. Harvester Press (and Humanities Press), Hassocks, Sussex, 1978. http://www.cs.bham.ac.uk/research/cogaff/crp.

[42] A. Sloman. Experiencing Computation: A Tribute to Max Clowes. In Masoud Yazdani, editor, *New horizons in educational computing*, pages 207 – 219. Ellis Horwood Series In Artificial Intelligence, Chichester, 1984. http://www.cs.bham.ac.uk/research/projects/cogaff/00-02.html#71.

[43] A. Sloman. The structure of the space of possible minds. In S. Torrance, editor, *The Mind and the Machine: philosophical aspects of Artificial Intelligence*. Ellis Horwood, Chichester, 1984. http://www.cs.bham.ac.uk/research/projects/cogaff/07.html#704.

[44] A. Sloman. What enables a machine to understand? In *Proc 9th IJCAI*, pages 995–1001, Los Angeles, 1985. http://www.cs.bham.ac.uk/research/projects/cogaff/81-95.html#4.

[45] A. Sloman. Why we need many knowledge representation formalisms. In M. Bramer, editor, *Research and Development in Expert Systems*, pages 163–183. Cambridge University Press, 1985. http://www.cs.bham.ac.uk/research/projects/cogaff/04.html#200406.

[46] 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. http://www.cs.bham.ac.uk/research/projects/cogaff/81-95.html#7.

[47] A. Sloman. The emperor's real mind. *Artificial Intelligence*, 56:355–396, 1992. Review of Roger Penrose's *The Emperor's new Mind: Concerning Computers Minds and the Laws of Physics*.

[48] A. Sloman. How to Dispose of the Free-Will Issue. *AISB Quarterly*, 82,:31–32, 1992. http://www.cs.bham.ac.uk/research/projects/cogaff/81-95.html#8.

[49] 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, Cambridge, UK, 1993. http://www.cs.bham.ac.uk/research/projects/cogaff/81-95.html#18.

[50] A. Sloman. Explorations in design space. In A.G. Cohn, editor, *Proceedings 11th European Conference on AI, Amsterdam, August 1994*, pages 578–582, Chichester, 1994. John Wiley.

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

[52] A. Sloman. Actual possibilities. In L.C. Aiello and S.C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR '96)*, pages 627–638, Boston, MA, 1996. Morgan Kaufmann Publishers.

[53] A. Sloman. The "Semantics" of Evolution: Trajectories and Trade-offs in Design Space and Niche Space, (Invited talk). In H. Coelho, editor, *Progress in Artificial Intelligence, Proceedings 6th Iberoamerican Conference on AI (IBERAMIA)*, pages 27–38, Lisbon, October 1998. Springer, lecture notes in Artificial Intelligence,.

[54] A. Sloman. Architectural requirements for human-like agents both natural and artificial. (what sorts of machines can love?). In K. Dautenhahn, editor, *Human Cognition And Social Agent Technology*, Advances in Consciousness Research, pages 163–195. John Benjamins, Amsterdam, 2000.

[55] A. Sloman. Interacting trajectories in design space and niche space: A philosopher speculates about evolution. In *et al.* M.Schoenauer, editor, *Parallel Problem Solving from Nature – PPSN VI*, Lecture Notes in Computer Science, No 1917, pages 3–16, Berlin, 2000. Springer-Verlag.

[56] A. Sloman. Evolvable biologically plausible visual architectures. In T. Cootes and C. Taylor, editors, *Proceedings of British Machine Vision Conference*, pages 313–322, Manchester, 2001. BMVA.

[57] A. Sloman. Architecture-based conceptions of mind. In *In the Scope of Logic, Methodology, and Philosophy of Science (Vol II)*, Synthese Library Vol. 316, pages 403–427. Kluwer, Dordrecht, 2002. http://www.cs.bham.ac.uk/research/projects/cogaff/00-02.html#57.

[58] A. Sloman. Diagrams in the mind. In M. Anderson, B. Meyer, and P. Olivier, editors, *Diagrammatic Representation and Reasoning*. Springer-Verlag, Berlin, 2002.

[59] A. Sloman. Requirements for a Fully Deliberative Architecture (Or component of an architecture). Research Note COSY-DP-0604, School of Computer Science, University of Birmingham, Birmingham, UK, May 2006. http://www.cs.bham.ac.uk/research/projects/cosy/papers/#dp0604.

[60] A. Sloman. Diversity of Developmental Trajectories in Natural and Artificial Intelligence. In C. T. Morrison and T. Tim Oates, editors, *Computational Approaches to Representation Change during Learning and Development. AAAI Fall Symposium 2007, Technical Report FS-07-03*, pages 70–79, Menlo Park, CA, 2007. AAAI Press. http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0704.

[61] A. Sloman. Architectural and representational requirements for seeing processes and affordances. In *Computational Modelling in Behavioural Neuroscience: Closing the gap between neurophysiology and behaviour.* Psychology Press, London., 2008. http://www.cs.bham.ac.uk/research/projects/cosy/papers#tr0801.

[62] A. Sloman. Putting the Pieces Together Again. In Ron Sun, editor, *Cambridge Handbook on Computational Psychology*, chapter 26, pages 684–709. Cambridge University Press, New York, 2008. http://www.cs.bham.ac.uk/research/projects/cogaff/07.html#710.

[63] A. Sloman. Some Requirements for Human-like Robots: Why the recent over-emphasis on embodiment has held up progress. In B. Sendhoff, E. Koerner, O. Sporns, H. Ritter, and K. Doya, editors, *Creating Brain-like Intelligence*. Springer-Verlag, Berlin, 2009. http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0804.

[64] A. Sloman and J. Chappell. The Altricial-Precocial Spectrum for Robots. In *Proceedings IJCAI'05*, pages 1187–1192, Edinburgh, 2005. IJCAI. http://www.cs.bham.ac.uk/research/cogaff/05.html#200502.

[65] A. Sloman and J. Chappell. Computational Cognitive Epigenetics (Commentary on [16]). *Behavioral and Brain Sciences*, 30(4):375–6, 2007. http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0703.

[66] A. Sloman and R.L. Chrisley. Virtual machines and consciousness. *Journal of Consciousness Studies*, 10(4-5):113–172, 2003.

[67] A. Sloman, R.L. Chrisley, and M. Scheutz. The architectural basis of affective states and processes. In M. Arbib and J-M. Fellous, editors, *Who Needs Emotions?: The Brain Meets the Robot*, pages 203–244. Oxford University Press, New York, 2005. http://www.cs.bham.ac.uk/research/cogaff/03.html#200305.

[68] A. Sloman and B.S. Logan. Building cognitively rich agents using the Sim_agent toolkit. *Communications of the Association for Computing Machinery*, 42(3):71–77, March 1999. http://www.cs.bham.ac.uk/research/projects/cogaff/96-99.html#49.

[69] A. Sloman and M. Scheutz. A Framework for Comparing Agent Architectures. In *Proceedings UKCI'02, UK Workshop on Computational Intelligence*, Birmingham, UK., September 2002. http://www.cs.bham.ac.uk/research/projects/cogaff/00-02.html#89.

[70] H.P. Stapp and H. Atmanspacher. Clarifications and Specifications: A Conversation with Henry Stapp. *Journal of Consciousness Studies*, 13(9):67–85, 2006. http://www.igpp.de/english/tda/pdf/stapp.pdf.

[71] Ron Sun, editor. *Cambridge Handbook on Computational Psychology.* Cambridge University Press, New York, 2008.

[72] G.J. Sussman. *A computational model of skill acquisition*. American Elsevier, 1975.

[73] K. V. Wilkes. Analysing Freud: Review of Mind, Psychoanalysis and Science, Eds. P. Clark, C. Wright. *The Philosophical Quarterly*, 40(159):241–254, April 1990.

[74] I.P. Wright, A. Sloman, and L.P. Beaudoin. Towards a design-based analysis of emotional episodes. *Philosophy Psychiatry and Psychology*, 3(2):101–126, 1996. http://www.cs.bham.ac.uk/research/projects/cogaff/96-99.html#2.