Seminar, University of Birmingham 16 Oct 2008
The Great debate Newcastle 21 Oct 2008
Mind as Machine Weekend Course, Oxford 1-2 Nov 2008

(This is work in progress. Comments and criticisms welcome.)

Why virtual machines really matter – for several disciplines

Aaron Sloman

http://www.cs.bham.ac.uk/~axs

School of Computer Science

The University of Birmingham

The latest version of these slides will be available online at
http://www.cs.bham.ac.uk/research/cogaff/talks/#virt

These slides on virtual machines and implementation are closely related:
http://www.cs.bham.ac.uk/research/cogaff/talks/#super
http://www.cs.bham.ac.uk/research/cogaff/talks/#inf

Last revised November 12, 2008

Presentations based on versions of these slides

Thur 16 Oct 2008: School of Computer Science Seminar, Birmingham

Why virtual machines really matter – for several disciplines
http://www.cs.bham.ac.uk/events/seminars/seminar_details.html?seminar_id=560

Tues 21 Oct 2008: The Great Debate, Newcastle

What can biologists, roboticists and philosophers learn from one another? (Unnoticed connections)
http://thegreatdebate.org.uk/UnnoticedConnections.html

Sat-Sun 1-2 Nov 2008: Weekend course Mind as Machine, Oxford

Why philosophers need to be robot designers
http://www.conted.ox.ac.uk/courses/details.php?id=O08P107PHR
http://oxfordphilsoc.org/

10-12 November 2008: Workshop on Philosophy and Engineering
Royal Academy of Engineering, London

Extended abstract: Virtual Machines in Philosophy, Engineering & Biology
http://www.cs.bham.ac.uk/research/projects/cogaff/08.html#803
My presentation will use only a small subset of these slides (20minutes)

ACKNOWLEDGEMENTS

Thanks especially to
Matthias Scheutz, Ron Chrisley and Jackie Chappell
And users of our SimAgent toolkit, from whom I have learnt much.

Many thanks to Linux/Unix developers:

I constantly use excellent virtual machines
that they have designed.

I am interacting with one now
and also several others running on it

Apologies for clutter: read only what I point at.
The slides are meant to be readable without me talking.

(Slides marked |X| are to be skipped during presentation.)

Abstract for talk on 16 Oct 2008, CS, Birmingham

One of the most important ideas (for engineering, biology, neuroscience, psychology, social sciences and philosophy) to emerge from the development of computing has gone largely unnoticed, even by many computer scientists, namely the idea of a running virtual machine (VM) that acquires, manipulates, stores and uses information to make things happen.

The idea of a VM as a mathematical abstraction is widely discussed, e.g. a Turing machine, the Java virtual machine, the Pentium virtual machine, the von Neumann virtual machine. These are abstract specifications whose relationships can be discussed in terms of mappings between them. E.g. a von Neumann VM can be implemented on a Universal Turing Machine. An abstract VM can be analysed and talked about, but, like a mathematical proof, or a large number, it does not **do** anything. The processes discussed in relation to abstract VMs do not occur in time: they are mathematical descriptions of processes that can be mapped onto descriptions of other processes. In contrast a physical machine can consume, transform, transmit, and apply energy, and can produce changes in matter. It can make things happen. Physical machines (PMs) also have abstract mathematical specifications that can be analysed, discussed, and used to make predictions, but which, like all mathematical objects cannot do anything.

But just as instances of designs for PMs can do things (e.g. the engine in your car does things), so can instances of designs for VMs do things: several interacting VM instances do things when you read or send email, browse the internet, type text into a word processor, use a spreadsheet, etc. But those running VMs, the active instances of abstract VMs, cannot be observed by opening up and peering into or measuring the physical mechanisms in your computer.

My claim is that long before humans discovered the importance of active virtual machines (AVMs), long before humans even existed, biological evolution produced many types of AVM, and thereby solved many hard design problems, and that understanding this is important (a) for understanding how many biological organisms work and how they develop and evolve, (b) for understanding relationships between mind and brain, (c) for understanding the sources and solutions of several old philosophical problems, (d) for major advances in neuroscience, (e) for a full understanding of the variety of social, political and economic phenomena, and (e) for the design of intelligent machines of the future. In particular, we need to understand that the word “virtual” does not imply that AVMs are unreal or that they lack causal powers, as some philosophers have assumed. Poverty, religious intolerance and economic recessions can occur in socio-economic virtual machines and can clearly cause things to happen, good and bad. The virtual machines running on brains, computers and computer networks also have causal powers. Some virtual machines even have desires, preferences, values, plans and intentions, that result in behaviours. Some of them get philosophically confused when trying to understand themselves, for reasons that will be explained. Most attempts to get intelligence into machines ignore these issues.

|X| Abstract for Newcastle

There are deep connections between ideas developed in computer science, biology and philosophy that have not been widely understood. A central feature common to biological organisms is the acquisition, manipulation, and use of information. Since the development of electronic computers, computer science has made major advances in the study of forms of information-processing. However we have still understood only a small subset of the information processing problems and solutions produced by biological evolution. Despite major advances in tools and techniques for investigating biological systems, we still lack good theories about what they are doing, how they do it, and whether it is possible to replicate or model those chemical information-processing functions in digital electronic computing systems.

One of the major advances in computer science and software engineering has been the separation of virtual machines from physical implementation, allowing many different kinds of functionality to share the same physical basis. It is very likely that evolution also ”discovered” the importance of that separation. Understanding what organisms do and how they do it may require us to shift the main focus of research on biological information-processing away from physical/chemical details towards investigation of the virtual machines used. That will require new ways of thinking about brains and other biological mechanisms.

Acknowledging the importance of virtual machines that process information and perform control functions has profound implications for philosophical investigations of the nature of causality, for it implies that events in virtual machines, can cause physical effects. Moreover, if engineers often find it useful to design and analyse complex systems in terms of the virtual machines involved rather than the specific physical mechanisms implementing them, the same could be true of biological and artificial systems that need to understand their own operations. From this viewpoint biological self-aware systems could to be construed as self-monitoring, self-modifying virtual machines that run on, but are different from, physical information-processing substrates. This has profound implications for several branches of philosophy, including, philosophy of mind, epistemology, philosophy of science and mathematics, and philosophical studies of free will. Questions about the nature of free-will are transformed in the context of virtual machines that are able to grow themselves ...

Summary

1. **What is a machine?**
2. Different concepts of “virtual machine”
   including active or running VMs, in contrast with abstract VMs (mathematical entities).
3. Oversimplified notions of VM used by many philosophers versus richer notions
4. VM functionalism, supervenience and causation
5. Virtual machines are everywhere
6. Why VMs are important in engineering
7. Concurrent, interacting VMs sharing a substrate
8. Why the same considerations are relevant to biology
9. “Separation of concerns” (modularity) can be horizontal or vertical
10. Importance for self-monitoring, control, management
11. Biological evolution probably “discovered” all this (and more) first
12. Why the physical sciences have explanatory gaps
13. We don’t need quantum mechanics to explain the phenomena
14. Counterfactual conditionals and free will
15. Virtual machines that build themselves
16. Problems for biology, psychology, neuroscience, philosophy
17. Further reading

What is a machine?

A machine is a complex enduring entity with parts (possibly a changing set of parts) that interact causally with one another as they change their properties and relationships.

Most machines are also embedded in a complex environment with which they interact.

The internal and external interactions may be discrete or continuous, sequential or concurrent.

Different parts of the machine, e.g. different sensors and effectors, may interact with different parts of the environment concurrently.

The machine may treat parts of itself as parts of the environment (during self-monitoring), and parts of the environment as parts of itself (e.g. tools, external memory aids). (See Sloman 1978, chapter 6)

The machine may be fully describable using concepts of the physical sciences (and mathematics), in which case it is a physical machine (PM).

Examples include levers, assemblages of gears, clocks, clouds, tornadoes, and myriad molecular machines in living organisms.

Some machines have states, processes and interactions whose descriptions use concepts that cannot be defined in terms of those of the physical sciences

e.g. “checking spelling”, “playing chess”, “winning”, “strategy”, “desire”. “belief”, “poverty”, “crime”, ...

(For now I’ll take that indefinability as obvious: it would take too long to explain. See `http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#models`)

Those include information-processing machines.

Mostly they are virtual machines (explained below).

Including socio-economic machines, ecosystems and many biological control systems.

What is a virtual machine?

A virtual machine (VM) is a machine whose interacting components and their states cannot be measured or detected using the techniques of the physical sciences (e.g. physics, chemistry), though in order to exist and work, it needs to be **implemented** in a physical machine.

An example is a running computer program doing things like checking spelling, playing chess, sorting email, computing statistics, etc.

"Incorrect spelling" cannot be defined in terms of concepts of physics, and instances of correct and incorrect spelling cannot be distinguished by physical measuring devices.

A socio-economic system is a more abstract and complex form of virtual machine: "economic inflation" and "recession" cannot be defined in terms of concepts of physics

Mental states and processes in humans and other animals can be regarded as states and processes in virtual machines, implemented in brains.

Much has been written about virtual machines by philosophers and others, but they are often mistaken, e.g.

Most ignore the complexity of the relations between VMs and PMs (e.g. 2-way causation)

Some claim that virtual machines, or their components, do not really exist or do not have causal powers, or that talking about them is merely a metaphorical way of talking about physical machines.

The virtual machines that allow criminals to transfer funds that belong to others do really exist, and do really produce effects.

Are virtual machines really just physical machines viewed as non-physical?

Different concepts of "virtual machine"

Some people object to claims

- that causal interactions can occur within a virtual machine,

and

- that events in a virtual machine can be caused by or can cause physical events,

because they ignore the difference between:

- a VM which is an abstract mathematical object
  (e.g. the Prolog VM, the Java VM, the Unix VM)
- a VM that is a running instance of such a mathematical object,
  controlling events in a physical machine (among other things it may be doing).
  (E.g. the instance of the linux operating system running my computer now.)

The difference between these two is very important.

The mathematical object does not do anything (as numbers don't).

Running instances of virtual machines can do many things e.g.

- landing a plane
- controlling a chemical plant
- monitoring patients in intensive care

Anyone who claims that a virtual machine is just a formal entity that cannot cause anything has not understood, or has forgotten, these points.

As several critics pointed out, this seems to be one of the mistakes in John Searle's paper attacking "strong AI" in 1980: "Minds Brains and Programs" *The Behavioral and Brain Sciences*,

There are two notions of virtual machine

We can contrast the notion of a PHYSICAL machine with:

- a VM which is **an abstract mathematical object** (e.g. the Prolog VM, the Java VM)
- a VM that is **a running instance of such a mathematical object**, controlling events in a physical machine, e.g. a running Prolog or Java VM.

Physical processes:	**Running virtual machines:**	**Mathematical models:**
currents	calculations	numbers
voltages	games	sets
state-changes	formatting	grammars
transducer events	proving	proofs
cpu events	parsing	Turing machines
memory events	planning	TM executions

VMs as mathematical objects are much studied in meta-mathematics and theoretical computer science. They are no more causally efficacious than numbers.

The main theorems of computer science, e.g. about computability, complexity, etc. are primarily about mathematical entities

They are applicable to non-mathematical entities with the same structure – but no non-mathematical entity can be proved mathematically to have any particular mathematical properties.

There's more on varieties of virtual machines in later slides.

SHOW SOME DEMOS OF VIRTUAL MACHINES

Some are artificial virtual machines, some natural virtual machines.

- Virtual 'marchers'
- Toy Emotional Agents in a virtual machine in a computer.
- Purely Reactive Sheepdog in a virtual machine.
- Hybrid Reactive Deliberative Sheepdog in an extended version of that VM.

  In both, the virtual dog can interact with virtual trees, virtual sheep, and also with the "mouse pointer" moved by a human.

  Changes in the active virtual machines produce changes on the screen as well as in the computer's memory, in the CPU, in interfaces, etc., and also changes in the mind of a person looking at the display, or trying to interact with it using a mouse.
- Betty the New Caledonian crow (if there's time)

  See Betty making a hook out of wire and using it, apparently working out in advance exactly what to do: `http://users.ox.ac.uk/~kgroup/tools/tools_main.html`

  Alas Betty died in 2005.
- A human child (if there's time)

  A pre-verbal child can see what someone else is trying to do, adopt the goal of helping, work out how to help, and then perform the actions. Before the actions start, a virtual machine is at work.

Some of the demos and videos are available online here

`http://www.cs.bham.ac.uk/research/poplog/figs/simagent`

`http://www.cs.bham.ac.uk/research/projects/cosy/conferences/mofm-paris-07/sloman/vid`

Summary

1. What is a machine?
2. Different concepts of “virtual machine”
   including active or running VMs, in contrast with abstract VMs (mathematical entities).
3. **Oversimplified notions of VM used by many philosophers versus richer notions**
4. VM functionalism, supervenience and causation
5. Virtual machines are everywhere
6. Why VMs are important in engineering
7. Concurrent, interacting VMs sharing a substrate
8. Why the same considerations are relevant to biology
9. “Separation of concerns” (modularity) can be horizontal or vertical
10. Importance for self-monitoring, control, management
11. Biological evolution probably “discovered” all this (and more) first
12. Why the physical sciences have explanatory gaps
13. We don’t need quantum mechanics to explain the phenomena
14. Counterfactual conditionals and free will
15. Virtual machines that build themselves
16. Problems for biology, psychology, neuroscience, philosophy
17. Further reading

Oversimplified notions of VM used by many philosophers

Some philosophers who know about Finite State Machines (FSMs), use a simple kind of "functionalism" (atomic state functionalism) as the basis for the notion of virtual machine, defined in terms of a set of possible states and transitions between them.

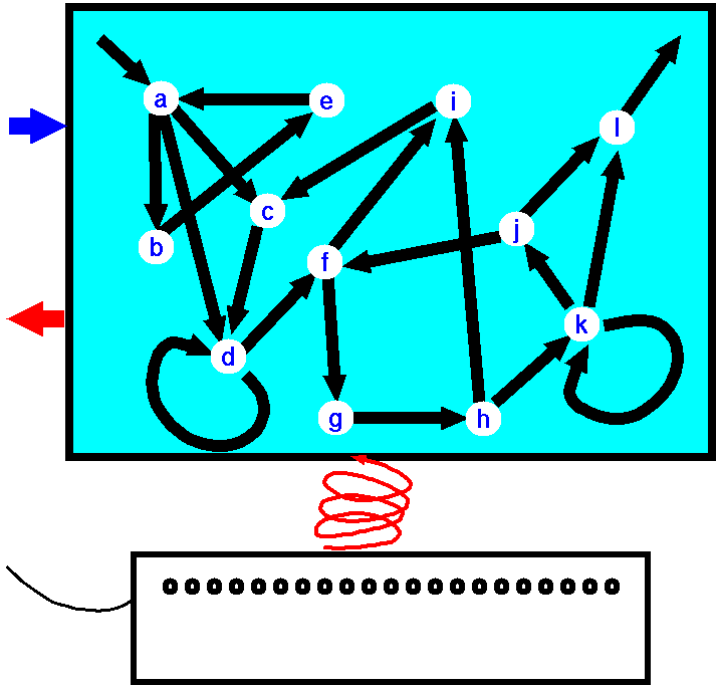
E.g. Ned Block, "What is functionalism?", 1996. (I think he has changed his views since then.)

On this model, a virtual machine that runs on a physical machine has a finite set of possible states (a, b, c, etc.) and it can switch between them depending on what inputs it gets. At each switch it may also produce some output. (The idea of a Turing machine combines this notion with the notion of an infinite tape.)

Finite, Discrete, State Virtual Machine:

Each possible state (e.g. a, b, c, ...) is defined by how inputs to that state determine next state and the outputs produced when that happens.

The machine can be defined by a set of rules specifying the state-transitions.



Implementation relation:

Physical computer:

As demonstrated by Alan Turing and others

This is a surprisingly powerful model of computation: but it is not general enough.

That kind of Functionalism is too simple

Instead of a single (atomic) state which switches when some input is received, a virtual machine can include many sub-systems with their own states and state transitions going on concurrently, some of them providing inputs to others.

- The different states may change on different time scales: some change very rapidly others very slowly, if at all.
- They can vary in their granularity: some sub-systems may be able to be only in one of a few states, whereas others can switch between vast numbers of possible states (like a computer's virtual memory).
- Some may change continuously, others only in discrete steps.
- The changes need not be synchronised – not even the discrete changes: some will be controlled by changes in the environment. Relative speeds of components can change over time.
- Some physical inputs may be "carriers" for virtual inputs,

  e.g. multiplexed signals, internet packets with different abstraction layers, hearing a symphony, speech inputs with different levels of meaning

Some sub-processes may be directly connected to sensors and effectors, whereas others have no direct connections to inputs and outputs and may only be affected very indirectly by sensors or affect motors only very indirectly (if at all!).

A richer model: Multiple interacting FSMs

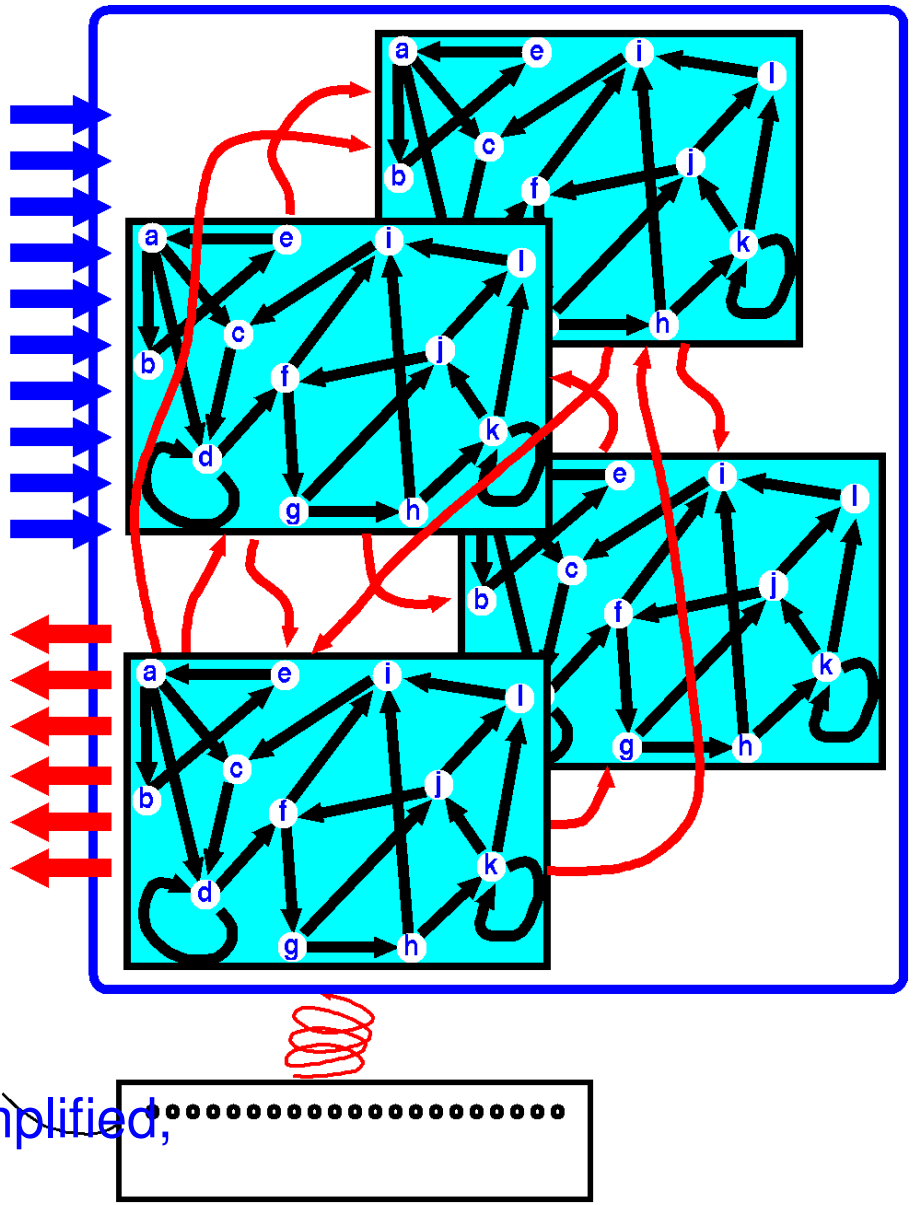
This is a more realistic picture of what goes on in current computers:

There are multiple input and output channels, and multiple interacting finite state machines, only some of which interact directly with the environment.

You will not see the virtual machine components if you open up the computer, only the hardware components.

The existence and properties of the FSMs (e.g. playing chess) cannot be detected by physical measuring devices.

But even that specification is over-simplified, as we'll see.



A possible objection: only one CPU?

Some will object that when we think multiple processes run in parallel on a single-CPU computer, interacting with one another while they run, we are mistaken because only one process can run on the CPU at a time, so there is always only one process running.

This ignores the important role of memory mechanisms in computers.

- Different software processes have different regions of memory allocated to them, which endure in parallel. So the processes implemented in them endure in parallel, and a passive process can affect an active one that reads some of its memory.

Moreover

- It is possible to implement an operating system on a multi-cpu machine, so that instead of its processes sharing only one CPU they share two or more.
- In the limiting case there could be as many CPUs as processes that are running.
- The differences between these different implementations imply that
  how many CPUs share the burden of running the processes is a contingent feature of the implementation of the collection of processes and does not alter the fact that there can be multiple processes running in a single-cpu machine.

A technical point: software interrupt handlers connected to physical devices that are constantly switched on, e.g. keyboard and mouse interfaces, video cameras, etc., mean that some processes are constantly "watching" the environment even when they don't have control of the CPU.

In virtual memory systems, and systems using "garbage collection" things are more complex than suggested here: the mappings between VM memory and PM memory keep changing.

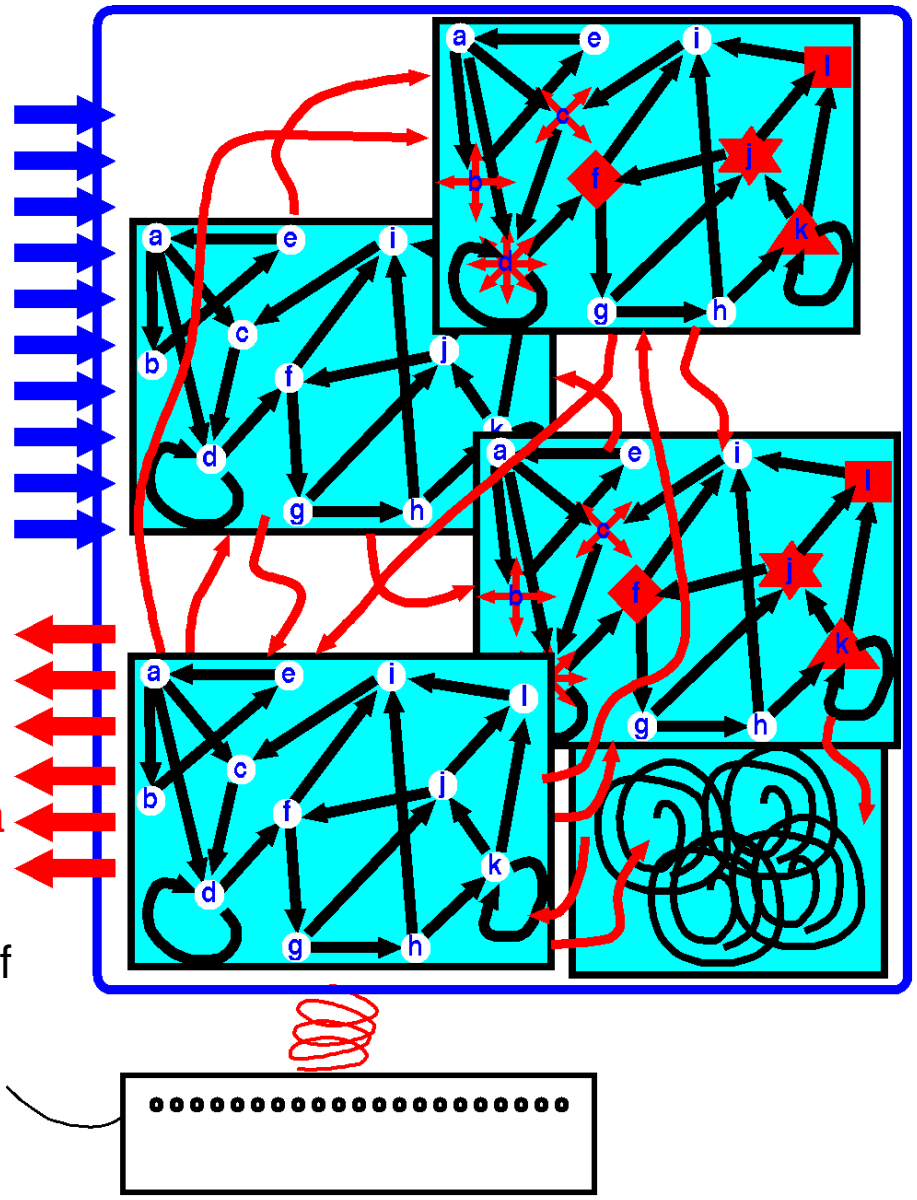
An even more general model

Instead of having a fixed set of sub-processes, many computing systems allow new VMs to be constructed dynamically,

- of varying complexity
- some of them running for a while then stopping,
- others going on indefinitely.
- some spawning new sub-processes...

The red polygons and stars might be subsystems where new, short term or long term, sub-processes (e.g. a new planning or parsing process) can be constructed within a supporting framework of virtual machines.

As indicated in the box with smooth curves, if analog devices are used, there can be VM processes that change continuously, instead of only discrete virtual machines. Some VMs simulate continuous change.



VMs can have temporarily or partly 'decoupled' components

- "Decoupled" subsystems may exist and process information, even though they have no connection with sensors or motors.
- For instance, a machine playing games of chess with itself, or investigating mathematical theorems, e.g. in number theory.
- It is also possible for internal VM processes to have a richness that cannot be expressed externally using the available bandwidth for motors.
- Likewise sensor data may merely introduce minor perturbations in what is a rich and complex ongoing internal process.

This transforms the requirements for rational discussion of some old philosophical problems about the relationship between mind and body:

E.g. some mental processes need have no behavioural manifestations, though they might, in principle, be detected using 'decompiling' techniques with non-invasive internal physical monitoring.
(This may be impossible in practice.)

Can evolution produce de-coupled VM sub-systems?

It is sometimes argued that sub-systems that do not have externally observable effects on behaviour would never be produced by evolution, because they provide no biological advantage.

This assumes an over-simplified view of evolution:

e.g. ignoring the fact that many neutral or harmless mutations can survive because they don't make sufficient difference to the survival chances of individuals. This could be because the environment is not sufficiently harsh or because more able individuals help less able ones or for other reasons.

A consequence is that a succession of changes that do not directly produce any great benefits (or disadvantages) may eventually combine to produce something very beneficial.

In some cases the benefits are insignificant until there's a major change in the environment requiring some new capability.

E.g. a succession of changes producing a mechanism for "thinking ahead" may be of no real benefit to members of a species until the environment changes so that food is not plentiful and actions to find food have to begin before the food is needed.

Likewise in individual development: virtual machines may change in (partly genetically programmed) ways that have no immediate benefit and show no behavioural consequences, but later on link up with other sub-systems and give the individual considerable advantages, e.g. mathematical thinking capabilities, perhaps.

Decoupling rules out behaviourism

The possibility of internal systems that are not connected, or not always connected with sensor or motors implies that descriptions of states and processes of virtual machines are not equivalent to any descriptions of input-output relations of the whole system.

This means that in some cases it may be impossible to detect what is going on inside some virtual machines by doing experiments like those in a traditional psychology lab, investigating responses to various kinds of stimuli.

But there are other means of investigation.

(Don't expect too much of brain imaging techniques in the near future.)

Finding out how to test a deep psychological theory can take a long time, and may depend on many other advances being achieved first.

Summary

1. What is a machine?
2. Different concepts of “virtual machine”
   including active or running VMs, in contrast with abstract VMs (mathematical entities).
3. Oversimplified notions of VM used by many philosophers versus richer notions
4. **VM functionalism, supervenience and causation**
5. Virtual machines are everywhere
6. Why VMs are important in engineering
7. Concurrent, interacting VMs sharing a substrate
8. Why the same considerations are relevant to biology
9. “Separation of concerns” (modularity) can be horizontal or vertical
10. Importance for self-monitoring, control, management
11. Biological evolution probably “discovered” all this (and more) first
12. Why the physical sciences have explanatory gaps
13. We don’t need quantum mechanics to explain the phenomena
14. Counterfactual conditionals and free will
15. Virtual machines that build themselves
16. Problems for biology, psychology, neuroscience, philosophy
17. Further reading

VM functionalism, supervenience and causation

We can contrast the widely used model of functionalism "Atomic State Functionalism" (ASF) with "Virtual Machine Functionalism" (VMF), where the latter allows far more complexity, including more complex mappings between virtual and physical phenomena.

There are deep implications for philosophy, psychology, biology, etc.

Instead of a single state that changes in discrete steps, with one input and one output channel, we can have many parts in different states, interacting with other sub-processes.

That is much closer to our ordinary understanding of a machine – e.g. a car engine with many concurrently active parts, or even a clock.

Abandon any idea that the total state is made up of a fixed number of discretely varying sub-states, since the complexity of a VM process can change dynamically:

We need to allow systems that can grow virtual structures and processes whose complexity varies over time, as crudely indicated in previous pictures.

Including trees, networks, algorithms, plans, thoughts, expectations, imaginings, desires, etc.

The machine may also include sub-systems that can change their state continuously, such as many physicists and control engineers have studied for many years

e.g. for controlling movements.

The label 'dynamical system' should be applicable to all these types of sub-system and to complex systems composed of them.

|X| Wrong models of computer programs

Many non-programmers, and some programmers have wrong models of what computer programs are.

E.g. they assume a program is a sequence of instructions to be obeyed, possibly with some loops or conditional instructions based on simple binary conditions.

Because of impoverished computing education they don't know about

- pattern-invoked modules (e.g. Prolog and Rule-based systems)
- non-deterministic invocation
- non-deterministic allocation/selection of values to variables or structures.
- communication through unification
- various kinds of context-sensitive instructions (e.g. the difference between an overt conditional and implicit conditionals based on variable binding)
- event-based module invocation, as opposed to explicit invocation
- use of memory management and garbage collection techniques that constantly alter the mapping from VM to physical machines
- programs that modify themselves or extend themselves while running

Several such non-conventional forms of programming were developed in our SimAgent toolkit, for the Cognition and Affect project:

`http://www.cs.bham.ac.uk/research/projects/poplog/packages/simagent.html`

See also the UKCRC Grand Challenge 7: "Journeys in Nonclassical Computation"

`http://www.bcs.org/server.php?show=ConWebDoc.4720`

Explaining what's going on in such cases requires a new deep analysis of the notion of **causation**

The relationship between objects, states, events and processes in virtual machines and in underlying implementation machines is a tangled network of causal interactions.

Software engineers have an intuitive understanding of it, but are not good at philosophical analysis.

Philosophers mostly ignore the variety of complex mappings between VMs and PMs when discussing causation and when discussing supervenience,

even though most of them now use multi-process VMs daily for their work.

Explaining how virtual machines and physical machines are related requires a deep analysis of causation that shows how the same thing can be caused in two very different ways, by causes operating at different levels of abstraction.

Explaining what 'cause' means is one of the hardest problems in philosophy.

For a summary explanation of two kinds of causation (Humean and Kantian) and the relevance of both kinds to understanding cognition in humans and other animals see:

`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#wonac`

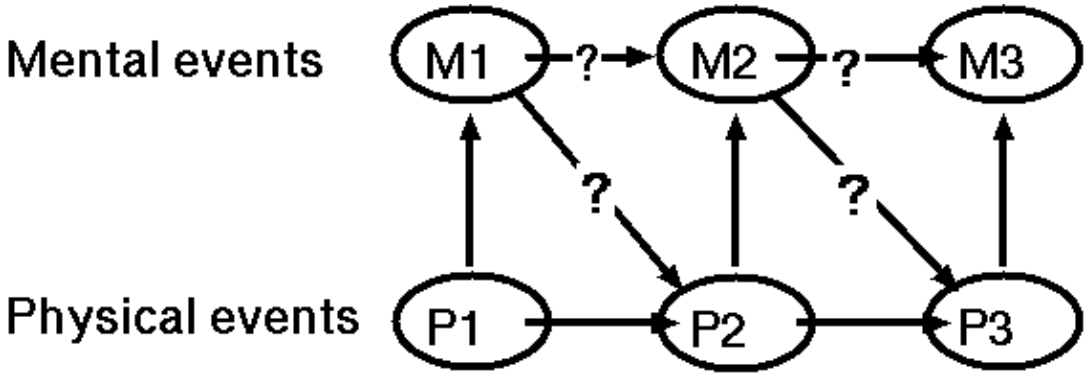
Supervenience and the mind-body relation

Some philosophers have tried to explain the relation between mind and body in terms of a notion of 'supervenience':

Mental states and processes are said to supervene on physical ones.

But there are many problems about that relationship:

Can mental process cause physical processes? (Sometimes called "downward causation".)
How could something happening in a mind produce a change in a physical brain?



(Think of time going from left to right)

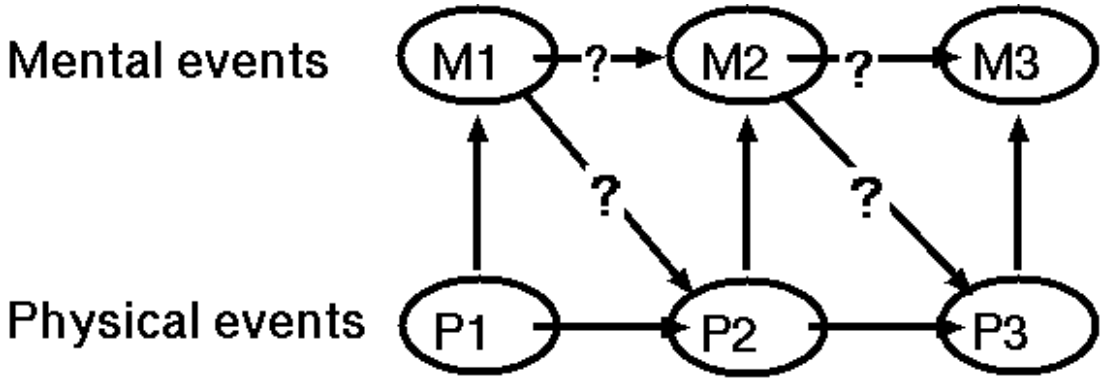
If previous physical states and processes suffice to explain physical states and processes that exist at any time, how can mental ones have any effect?

How could your decision to come here make you come here – don't physical causes (in your brain and in your environment) suffice to make you come?
If they suffice, how could anything else play a role?

Must non-physical events be epiphenomenal?

Traditionally, epiphenomenalism states that mental events and processes cannot cause anything to happen.
A modern version might say the same about VM events and processes.

Consider a sequence of virtual machine events or states M1, M2, etc. implemented in a physical system with events or states P1, P2, ...



If P2 is caused by its physical precursor, P1, that seems to imply that P2 cannot be caused by M1, and likewise M2 cannot cause P3.

Moreover, if P2 suffices for M2 then M2 is also caused by P1, and cannot be caused by M1. Likewise neither P3 nor M3 can be caused by M2.

So the VM events cannot cause either their physical or their non-physical successors.

This would rule out all the causal relationships represented by arrows with question marks, leaving the M events as epiphenomenal.

The flaw in the reasoning?

THIS IS HOW THE ARGUMENT GOES:

IF

(1) physical events are physically determined

E.g. everything that happens in an electronic circuit, if it can be explained at all by causes, can be fully explained according to the laws of physics: no non-physical mechanisms are needed, though some events may be inexplicable, according to quantum physics.

AND

(2) physical determinism implies that physics is 'causally closed' backwards

see "The Completeness of the Physical" in Stanford Encyclopedia of philosophy.
`http://plato.stanford.edu/entries/mental-causation/#ComPhy`

I.e. if all caused events have physical causes, then nothing else can cause them: any other causes will be *redundant*. (E.g. your decisions cannot cause your hands to move.)

THEN

no non-physical events (e.g VM events) can cause physical events

E.g. our thoughts, desires, emotions, etc. cannot cause our actions.
And similarly poverty cannot cause crime, national pride cannot cause wars, and computational events cannot cause a plane to crash, a picture to be displayed, etc.

ONE OF THE TWO CONJUNCTS, (1) or (2), IS INCORRECT. WHICH?

It's the second conjunct

Some people think the flaw is in the first conjunct:

i.e. they assume that there are some physical events that have no *physical* causes but have some other kind of cause that operates independently of physics, e.g. a spiritual or mental event that has no physical causes.

The theoretical physicist Henry Stapp argues that the equations of quantum theory require consciousness to influence physical events. `http://www.igpp.de/english/tda/pdf/stapp.pdf`

The real flaw is in the second conjunct:

i.e. the assumption that physical determinism

(i.e. every physical event or process is completely determined by previous states and processes)

implies that physics is 'causally closed' backwards.

Examples given previously from computing and also human psychology, social, economic and biological phenomena, show that many of our common-sense ways of thinking and reasoning contradict that assumption.

Explaining exactly what is wrong with it requires unravelling the complex relationships between statements about causation and counterfactual conditional statements.

A sketch of a partial (incomplete) explanation can be found in the last part of this tutorial:
`http://www.cs.bham.ac.uk/~axs/ijcai01`

Some philosophers think this requires thinking about connections between possible worlds. I think a better route is to understand how **this world** works.

We often assume multiple varieties of causation

A person drives home one night after drinking with friends in a pub.

As he goes round a bend in the road he skids sideways into an oncoming car and and the driver in the other car dies.

In court, the following facts emerge.

- The driver had exceeded the recommended alcohol limit for driving, but had often had the extra glass and then driven home on that route without anything going wrong.
- There had earlier been some rain followed by a sharp drop in temperature, as a result of which the road was unusually icy.
- The car was due for its MOT test, and he had been given two dates for the test, one before the accident and one after. He chose the later date. Had he taken the earlier date worn tires would have been detected and replaced with tires that could have gripped ice better.
- There had been complaints that the camber on the road was not steep enough for a curve so sharp, though in normal weather it was acceptable.
- The driver was going slightly faster than normal because he had been called home to help a neighbour who had had a bad fall.
- A few minutes after the accident the temperature rose in a warm breeze and the ice on the road melted.

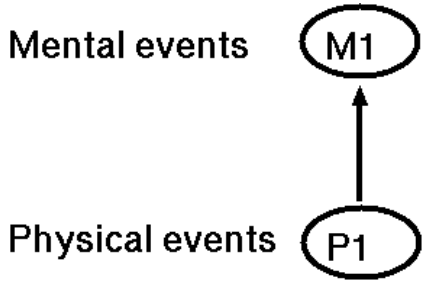
What caused the death of the other driver?

How VM events and processes can be causes

We need an explanation of how VM causes and PM causes can co-exist and both be causes of other VM and PM events and processes.

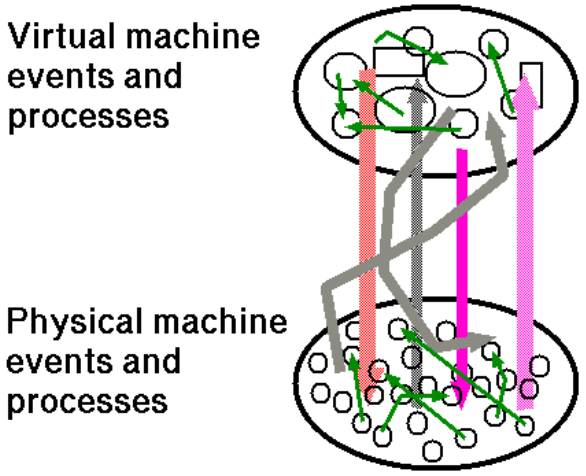
The crucial point is that the existence of causal links is equivalent to the existence of whatever makes certain sets of conditional statements (including counterfactual conditionals) true or false.

Our previous diagrams implicitly supported a prejudice by showing a single upward pointing arrow from each physical state to the mental, or virtual machine state, above it.



This implied a simple one-way dependency relationship, where complex two-way relationships actually exist.

The software and hardware engineering that has to be done to make virtual machines work as they do on computers requires a much more complex set of relationships, with two-way causal interactions between VM events and physical events: programs must make things happen.



Hardware and software system engineers have learnt over half a century how to **ensure** that there are many different sorts of true counterfactual conditionals both about how VM events would have been different if PM events had been different **and** about how PM events would have been different if VM events had been different.

They are not perfect, so sometimes there are bugs, but many programs do work!

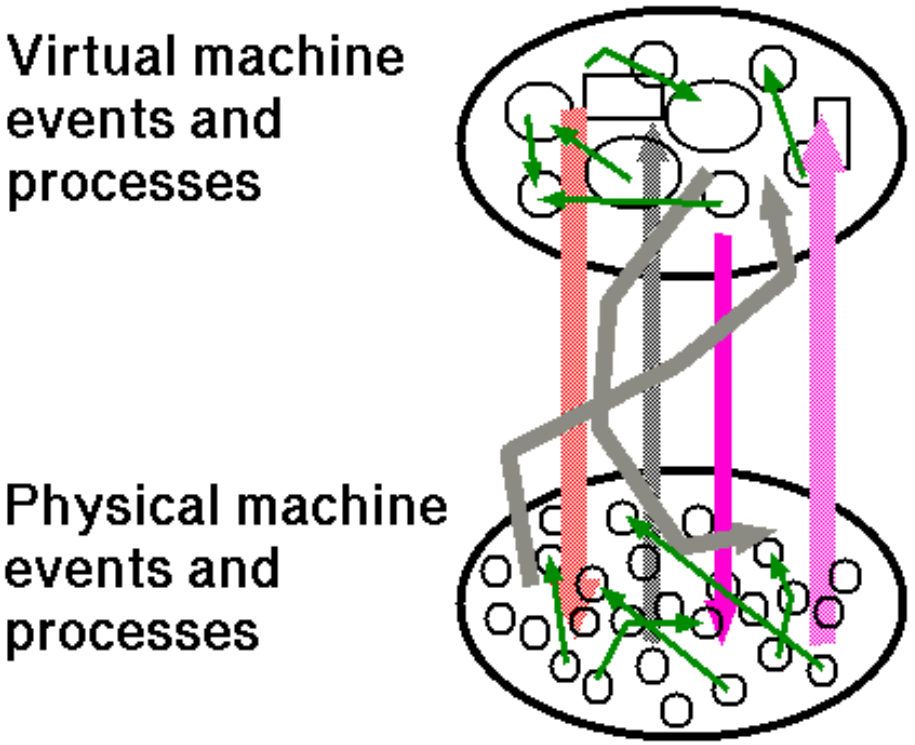
Generalised Supervenience Depicted

The two machines (PM and VM) need not be isomorphic: they can have very different structures.

There need not be any part of the PM that is isomorphic with the VM.

Not only static parts and relations but also processes and causal relations can supervene on physical phenomena.

The structure of the VM can change significantly (parts added and removed, and links between parts being added and removed) without structural changes occurring at the physical level –



though the physical states of millions of switches may change as the (much simpler, more abstract) VM changes and causal interactions occur.

NB: the mappings between PM components and VM components may be complex, subtle in kind, and changing.

A very large “sparse array” in the VM may contain many more locations than there are switches in the PM (as long as not all locations are actually occupied).

Distinct objects in the VM can have implementations that share parts of the PM.

|X| Notions of Supervenience

We can distinguish at least the following varieties

- property supervenience

  (e.g. having a certain temperature supervenes on having molecules with a certain kinetic energy.)

- pattern supervenience

  (e.g., supervenience of various horizontal, vertical and diagonal rows of dots on a rectangular array of dots, or the supervenience of a rotating square on the pixel matrix of a computer screen.)

- mereological, or agglomeration, supervenience

  (e.g., possession of some feature by a whole as the result of a summation of features of parts, e.g. the supervenience of the mass of a stone on the masses of its atoms, or the supervenience of the centre of mass on the masses and locations of its parts, each with its own mass)

- mechanism supervenience

  (supervenience of one machine on another: a collection of interacting objects, states, events and processes supervenes on some lower level, often more complex, reality, e.g., the supervenience of a running operating system on the computer hardware – this type is required for intelligent control systems, as probably discovered by evolution millions of years ago?)

We are talking about mechanism supervenience.
The other kinds are less closely related to implementation of VMs on PMs.

Mechanism supervenience, far from being concerned with how one property relates to others, is concerned with how a complex ontology (collection of diverse types of entity, types of events, types of process, types of state, with many properties, relationships and causal interactions) relates to another ontology.

This could be called “ontology supervenience”. Perhaps “ontology instance supervenience” would be better.

A more general notion of supervenience

Supervenience is often described as a relation between properties: e.g. a person's mental properties supervene on his physical properties (or "respects").

> '[...] supervenience might be taken to mean that there cannot be two events alike in all physical respects but differing in some mental respects, or that an object cannot alter in some mental respect without altering in some physical respect.'
>
> D. Davidson (1970), 'Mental Events', reprinted in: *Essays on Action and Events* (OUP, 1980).

In contrast we are concerned with a relation between *ontologies* or complex, interacting parts of ontologies, not just properties.

> The cases we discuss involve not just one object with some (complex) property, but large numbers of VM components enduring over time, changing their properties and relations, and interacting with one another: e.g. data-structures in a VM, or several interacting VMs, or thoughts, desires, intentions, emotions, or social and political processes, all interacting causally – the whole **system** supervenes.

A single object with a property that supervenes on some other property is just a very simple special case. We can generalise Davidson's idea:

> A functioning/working ontology supervenes on another if there cannot be a change in the first without a change in the second.

NOTE: the idea of "supervenience" goes back to G.E.Moore's work on ethics. A useful introduction to some of the philosophical ideas is: Jaegwon Kim, *Supervenience and Mind: Selected philosophical essays*, 1993, CUP.

Multiple layers of virtual machinery

The discussion so far suggests that there are two layers

- Physical machinery
- Virtual machinery

However, just as some physical machines (e.g. modern computers) have a kind of generality that enables them to support many different virtual machines

(e.g. the same computer may be able to run different operating systems – Windows, or Linux, or MacOS, or ....)

so are there some virtual machines that have a kind of generality that enables them to support many different "higher level" virtual machines running on them

(e.g. the same operating system VM may be able to run many different applications, that do very different things, – window managers, word processors, mail systems, spelling correctors, spreadsheets, compilers, games, internet browsers, CAD packages, virtual worlds, chat software, etc. ....)

It is also possible for one multi-purpose VM to support another multi-purpose VM, which supports additional VMs.

So VMs may be layered:

VM1 supports VM2 supports VM3 supports VM4, etc.

The layers can branch, and also be circular, e.g. if VM1 includes a component that invokes a component in a higher level VMk, which is implemented in VM1.

|X| Computer Engineering and Science

A complete explanation of how VMs interact with PMs in computers would require descriptions (tutorials) explaining how the following work:

- Physical components used in computers
- Digital electronic circuits and their mechanisms
- Various kinds of interfaces/transducers linking computers to other devices (hard drives, displays, keyboards, mice, networks)
- Operating systems
- Device drivers
- File systems
- Memory management systems
- Compilers
- Interpreters
- Interrupt handlers
- Caches
- Programmable firmware stores

and other things I've forgotten to mention.

Note: my own understanding of many of those is incomplete.

Problems with epiphenomenalism

Problems with the 'monistic', 'reductionist', physicalist thesis that non-physical events are epiphenomenal:

- It presupposes a layered view of reality with a well-defined ontological bottom level, where "real" causation is assumed to exist.
  IS THERE ANY SUCH BOTTOM LEVEL?
- There are deep unsolved problems about how any level could be the "real" physical level, and how we could tell whether something is the bottom level.
- The thesis renders inaccurate or misleading much of our indispensable ordinary and scientific discourse, e.g.
  - Anger can make people do things they later regret.
  - Was it the government's policies that caused the depression or would it have happened no matter which party was in power?
  - Your despair made me frightened.
  - Changes in a biological niche can cause changes in the spread of genes in a species.
  - Information about Diana's death spread rapidly round the globe, causing many changes in TV schedules and news broadcasts.
  - Addiction to junk information can cause intellectual obesity.

  However, many of our normal ways of speaking may be confused or erroneous (like old ideas about diseases).

Some scientists and philosophers argue that science is not concerned with causation: laws express fixed relationships not causal connections.

Is a VM identical with a PM that implements it?

One philosophical response is the "identity theory", which states:

The VM **just is** whatever physical mechanism implements it,
so causation by VM events is nothing more than causation by PM events.

(There are different variants of this sort of response.)

However, saying that the non-physical phenomena are **identical** with physical ones is problematic:

(a) It does not explain anything (e.g. how different sorts of VMs work).

(b) It contradicts the asymmetry in the implementation/realisation relation:

- If A is identical with B then B is identical with A.
- But if A is implemented/realised in B it does not follow that the reverse is true.
- So implementation/realisation cannot be an species of identity.
- Nor is supervenience a kind of identity, if that's the same is implementation/realisation.

(Attempts to escape these problems include treating identity as a relation between instances/individuals and implementation as a relation between properties or types. But the **instance** of Linux running on my computer is implemented in the physical machinery of the computer.)

(c) It ignores all the differences between the causation in VMs
that software engineers think about when developing, debugging and extending their programs,
and the causation in electronic circuits and other PMs
that electronic engineers think about when designing computer components.
and the complex and changing relations between VMs and PMs.

|X| Philosophical note

- The argument that VM events cannot have causal powers is usually based on ignorance of how actual implementations of virtual machines work, and the ways in which they produce the causal powers, on which so much of our life and work increasingly depend.

  More and more control systems depend on virtual machines that process information and take decisions.

- The ideas presented here provide only an incomplete draft attempt to clarify these topics.
- There is much more that is intuitively understood by engineers, but has not yet been clearly articulated and analysed. (As far as I know.)

Strictly, many of the people who use this kind of practical understanding should be called craftsmen/women rather than engineers.

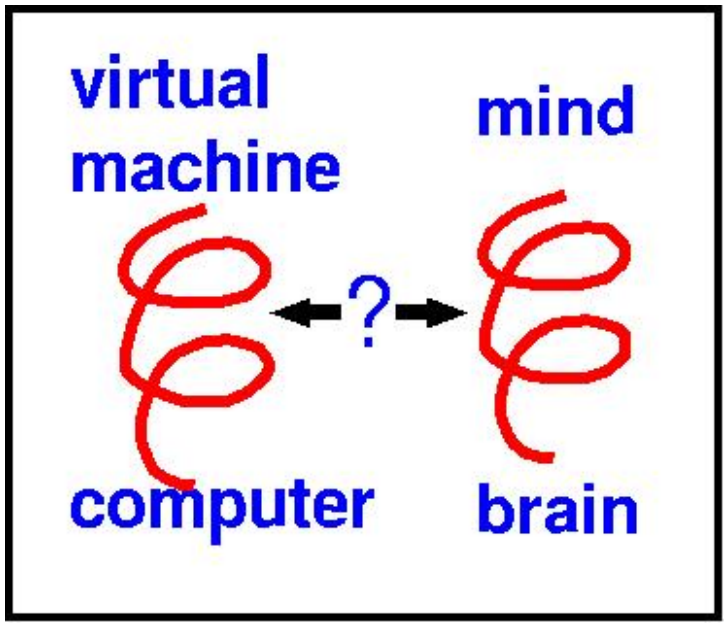
This is a special case of the general fact that craft precedes science and engineering.

(Engineering = Craft+Science)

Philosophers, psychologists and neuroscientists need to learn some of the craft, and the underlying science, in order to avoid confusions and false assumptions about virtual machines and possible varieties of emergence.

‘Emergence’ need not be a bad word

People who have noticed the need for pluralist ontologies often talk about ‘emergent’ phenomena.
But the word has a bad reputation, associated with mysticism, vitalist theories, sloppy thinking, wishful thinking, etc.



If we look closely at the kinds of ‘emergence’ found in virtual machines in computers, where we know a lot about how they work (because we designed them and can debug them, etc), then we’ll be better able to go on to try to understand the more complex and obscure cases, e.g. mind/brain relations.

Virtual machine emergence adds to our ontology: the new entities are not definable simply as patterns or agglomerations in physical objects (they are not like ocean waves).

My claim is that engineers discussing implementation of VMs in computers and philosophers discussing supervenience of minds on brains are talking about the same ‘emergence’ relationship – involving VMs implemented (ultimately) in physical machines.

NB. It is not just a metaphor: both are examples of the same type.

(NEW) Am I a reductionist?

Two kinds of reductionism.

- Definitional/deductive reductionism of M to P

  This claims that phenomena of type M can be fully described using concepts that are definable using the concepts of type P.

  A consequence is that for any set Sm of true statements about phenomena of type M there will be some set Sp (maybe more than one set) of true statements about phenomena of type P, such that

  SM CAN BE DERIVED FROM SP USING VALID LOGICAL INFERENCES

- Implementational reductionism of M to P

  This claims that no set Sm of phenomena of type M, or set of things done by or caused by phenomena Sm of type M can be exist unless there exist phenomena Sp of type P that provide a sufficient causal basis for Sm, I.e.

  SM MUST BE **IMPLEMENTED IN** OR **REALISED IN** SOME SET SP

  This is compatible with the possibility of Sm being implemented in a quite different set Sp'.

(NEW) How can truth of Sm be established?

How can the existence of an implemented Sm be established?

Sometimes the designer of Sm can indicate how Sp was created in order to implement Sm, and may be able to specify an unending set of tests that the system will pass.

For naturally occuring examples of Sm the claim that it exists and has particular components doing particular things will be a **theory** subject to ongoing tests and revisions in the light of new information.

If there are rival theories attempting to explain the same phenomena this will be no different from rival theories in other sciences.

A theory that specifies that a Vm is running in some physical system making some set Sm of statements true, may be able to specify an indefinitely large and varied set of tests for which Sm will predict the outcomes.

If two distinct theories Sm and Sm' explain the same results then it may be necessary to derive more consequences to see if there is some unnoticed difference, or else to open up the physical system to new forms of inspection.

At present brain scanning mechanisms are mostly much too crude for this purpose.

That may change one day.

Summary

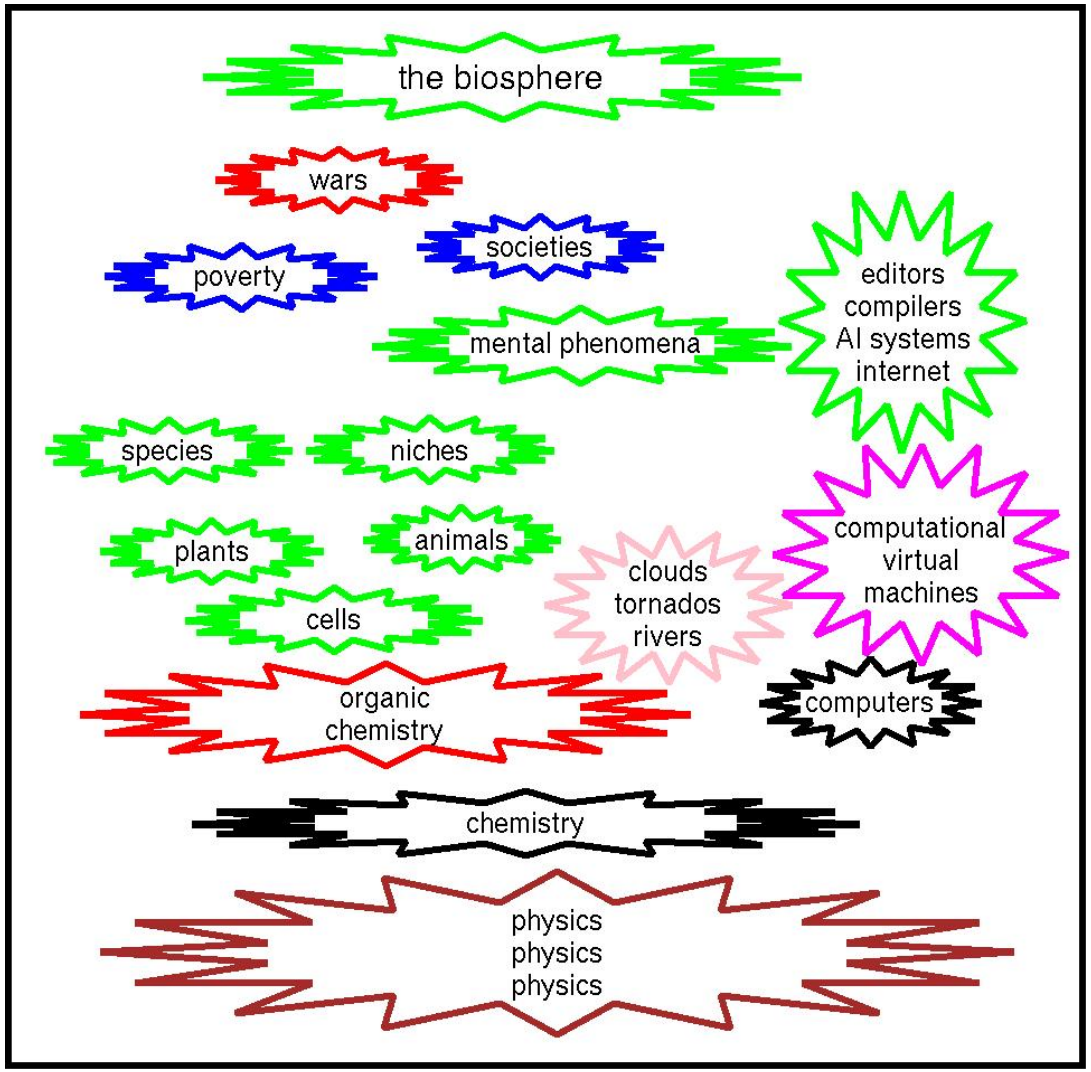
1. What is a machine?
2. Different concepts of “virtual machine”
   including active or running VMs, in contrast with abstract VMs (mathematical entities).
3. Oversimplified notions of VM used by many philosophers versus richer notions
4. VM functionalism, supervenience and causation
5. **Virtual machines are everywhere**
6. Why VMs are important in engineering
7. Concurrent, interacting VMs sharing a substrate
8. Why the same considerations are relevant to biology
9. “Separation of concerns” (modularity) can be horizontal or vertical
10. Importance for self-monitoring, control, management
11. Biological evolution probably “discovered” all this (and more) first
12. Why the physical sciences have explanatory gaps
13. We don’t need quantum mechanics to explain the phenomena
14. Counterfactual conditionals and free will
15. Virtual machines that build themselves
16. Problems for biology, psychology, neuroscience, philosophy
17. Further reading

Virtual machines are everywhere

At all levels there are objects, properties, relations, structures, mechanisms, states, events, processes and also many CAUSAL INTERACTIONS.

E.g. poverty can cause crime.

- All levels are ultimately realised (implemented) in physical systems.
- Different disciplines use different approaches (not always good ones).
- Nobody knows how many levels of virtual machines physicists will eventually discover. (Uncover?)
- The study of virtual machines in computers is just a special case of more general attempts to describe and explain virtual machines in our world.



See the IJCAI'01 Philosophy of AI tutorial (written with Matthias Scheutz) for more on levels and causation:
`http://www.cs.bham.ac.uk/~axs/ijcai01/`

Physics also deals with different levels of reality

- The "observable" level with which common sense, engineering, and much of physics has been concerned for thousands of years:
  - levers, balls, pulleys, gears, fluids, and many mechanical and hydraulic devices using forces produced by visible objects.
- Unobservable extensions
  - sub-atomic particles and invisible forces and force fields, e.g. gravity, electrical and magnetic forces.
- Quantum mechanical extensions
  - many things which appear to be inconsistent with the previous ontology of physics

Between the first two levels we find the ontology of chemistry, which includes many varieties of chemical compounds, chemical events, processes, transformations, causal interactions.

The chemical entities, states, processes, causal interactions are normally assumed to be "fully implemented" (fully grounded) in physics.

We don't know how many more levels future physicists will discover.

IS THERE A 'BOTTOM' LEVEL?

Rigid Physicalism

What could be called "rigid physicalism" states:

- there is only one level of reality, the 'fundamental' physical level,
- causes can exist only at that level of physics, and nowhere else,
- everything else is just a way of looking at those fundamental physical phenomena.

The possibility of future extensions to physics causes problems for this view.

- Rigid physicalism leaves many questions unanswered: we have no idea what will be regarded as fundamental in physics in a hundred or a thousand years time, or perhaps is already so regarded by more advanced physicists on another planet,
- If the only 'real' causes are those that operate at the fundamental level of physical reality then our talk about one billiard ball *causing another to move* does not describe what is 'really' going on.

Rigid physicalism is an extreme version of the theory that

'only fundamental physical causes are real'

This form of physicalism implies that most of our beliefs about causation, including physical and chemical causation, are illusory.

An alternative view:

Alternatively we can accept that there are different 'levels' at which causes can operate.

Then not only do sub-atomic particles and other recently discovered physical entities interact causally, so also do

- billiard balls,
- clock-springs,
- tidal waves,
- tornadoes,
- planets,

etc.

And also many non-physical things?

For instance

biological entities, social phenomena, economic phenomena, mental phenomena

and structures, processes, and relationships in virtual machines running in computers.

Example: The ontology of biology

Biology introduces several extensions to our ontology, most non-physical: E.g.

- Organism
- Reproduction
- Growth, development and learning
- Perception, reasoning, motivation, planning
- Disease, injury and death
- Species and societies
- Genes and inheritance
- Information (acquired and used by individuals or by genomes)
- Evolution, natural selection, fitness, advantage, competition, etc. ....

These are non-physical in that they have properties that are not physical properties, and are not definable in terms of physical concepts, and are not observable or measurable using physical instruments (scales, calipers, voltmeters, thermometers, etc. etc.)

We normally (apart from vitalists and some theologians) assume that, just as chemical phenomena are implemented/realised in physics, so also are:

Biological objects, events, processes
"fully implemented (realised)" in physics and chemistry,

The implementation/realisation relation

Phenomena of type X (e.g. biological phenomena)

are fully implemented in, or realised in, or grounded in phenomena of type Y

(e.g. physical phenomena)

if and only if:

(a) type X phenomena *cannot exist without* some entities and processes of type Y.

(i.e. it is necessary that something of type Y exist for anything of type X to exist)

(b) certain entities and processes of type Y *are sufficient for* the phenomena of type X to exist – they constitute the implementation.

(The actual implementation may be sufficient but not necessary: there can be alternative implementations.)

Example: if computational virtual machines are fully grounded in physical machines then

(a) computational machines cannot exist without being physically embodied

(b) their physical embodiments suffice for their existence - no extra independent stuff is needed. no computational spirit, soul, etc.

Warning: I have argued that some of what implements a VM (e.g. enables its mental states to be what they are) exists outside the physical body associated with the VM.

P.F. Strawson argued in *Individuals: An essay in descriptive metaphysics* (1959) that what makes it possible for my mental state now to include a thought about Julius Caesar includes a chain of causal connections between things that existed in Caesar's time and place and things now in my mind.

Summary

1. What is a machine?
2. Different concepts of “virtual machine”
   including active or running VMs, in contrast with abstract VMs (mathematical entities).
3. Oversimplified notions of VM used by many philosophers versus richer notions
4. VM functionalism, supervenience and causation
5. Virtual machines are everywhere
6. **Why VMs are important in engineering**
7. Concurrent, interacting VMs sharing a substrate
8. Why the same considerations are relevant to biology
9. “Separation of concerns” (modularity) can be horizontal or vertical
10. Importance for self-monitoring, control, management
11. Biological evolution probably “discovered” all this (and more) first
12. Why the physical sciences have explanatory gaps
13. We don’t need quantum mechanics to explain the phenomena
14. Counterfactual conditionals and free will
15. Virtual machines that build themselves
16. Problems for biology, psychology, neuroscience, philosophy
17. Further reading

Why virtual machines are important in engineering

They provide "vertical" separation of concerns

Part of the table of contents of The Well-Designed Young Mathematician, *AI Journal*, December 2008
http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0807

4 Why virtual machines are useful in animals and machines
  4.1 Active, interacting VMs can reduce complexity for designers
  4.2 VMs reduce combinatorial complexity for system designers
  4.3 Benefits of problem-decomposition
  4.4 Layered biological virtual machinery
  4.5 Benefits for individual machines, or animals, of using VMs
5 Concurrent virtual machine processes in a serial computer
  5.1 Concurrent causal influences on a sequential machine
  5.2 Evolution seems to have got there first
  5.3 Why the physical sciences have explanatory gaps
  5.4 Some philosophical implications
  5.5 The objective existence of virtual machine processes
  5.6 Formalisms for describing VMs
  5.7 VMs with continuously varying components
  5.8 Loose coupling or non-coupling with the environment
  5.9 Virtual machines, not quantum machines
6 Counterfactual conditionals and virtual machine behaviours
  6.1 Potentialities inherent in VMs
  6.2 On feeling and being free to choose
  6.3 Machines that refer to the internals of other machines
  6.4 Substantive scientific questions about VMs

"Separation of concerns" (modularity): horizontal or vertical

When different designers work on different parts of a complex system, having previously agreed on the specifications of **interfaces** between the systems this can enormously reduce workloads.

Each team of designers works on a specific module, or small collection of modules, without having to think about how the other modules are designed, as long as they meet their specifications.

Of course it sometimes turns out that this separation of effort was misguided: the wrong fracture-lines were assumed in the problem.

But that is one of many kinds of learning that is part of the development of science and engineering.

The use of such modularity in design is sometimes described as "separation of concerns".

I call it "horizontal separation of concerns" because the different modules in some sense exist on the same level, and none of them depends for its very existence on the others: it can be, and often is, combined with different modules in different systems.

When some people design and implement a virtual machine and others work on higher level virtual machines that depend on it, then I call that "vertical separation of concerns".

If we study the benefits of vertical separation of concerns, we can understand why they proved useful in biological evolution, including human evolution.

Concurrent, interacting virtual machines sharing a substrate

In a multi-processing computer the complexity would be totally unmanageable if software designers had to think about all the possible sequences of machine instructions.

Instead we use a VM substrate for handling multiple processes, with mechanisms for

- memory management
- context switching
- scheduling
- handling privileges and access rights, etc.
- filestore management
- various device drivers
- networking
- and in some cases use of multiple CPUs

These allow a “vertical separation of concerns”

including handling: dynamically changing situations, porting to new hardware, and changing or adding parts of a machine (memory, file store, devices).

Benefits of using VMs

In some cases problems in the operation of a computer are due to physical faults, requiring some hardware component to be repaired or replaced. The problem is far more often a fault in some software.

When a piece of software runs, what physical events occur in the computer depends on all sorts of details, including the design of the hardware, what else is using it at the same time, and how physical memory has been allocated between processes,

> In a multiprocessing system with different subsystems running, S1, S2, S3..., the detailed sequence of physical events occurring in each period of a few seconds can be unique.

If programmers developing or debugging software had to think about and explicitly specify all those possible physical events, of which many millions can occur each second, the design task would be completely intractable.

> (That's how early computers (e.g. in the 1950s) had to be programmed, but the sequences were much simpler, and occurred far more slowly.)

The existence of a VM platform such as a modern operating system makes it possible for the designer of each of S1,S2, etc. to think only about the VM events required for that subsystem, ignoring both what the others do (except for a few special relationships) and ignoring what the physical hardware does (except for the exceptional system that has to detect and deal with hardware failures, or insertion/removal of hardware).

Without this vertical separation of concerns it would be impossible for designers to create modern computing systems.

Summary

1. What is a machine?
2. Different concepts of “virtual machine”
   including active or running VMs, in contrast with abstract VMs (mathematical entities).
3. Oversimplified notions of VM used by many philosophers versus richer notions
4. VM functionalism, supervenience and causation
5. Virtual machines are everywhere
6. Why VMs are important in engineering
7. Concurrent, interacting VMs sharing a substrate
8. Why the same considerations are relevant to biology
9. “Separation of concerns” (modularity) can be horizontal or vertical
10. **Importance for self-monitoring, control, management**
11. Biological evolution probably “discovered” all this (and more) first
12. Why the physical sciences have explanatory gaps
13. We don’t need quantum mechanics to explain the phenomena
14. Counterfactual conditionals and free will
15. Virtual machines that build themselves
16. Problems for biology, psychology, neuroscience, philosophy
17. Further reading

Self-monitoring and virtual machines

Systems dealing with complex changing circumstances and needs may need to monitor themselves, and use the results of such monitoring in taking high level control decisions.

E.g. which high priority task to select for action.

Using a high level virtual machine as the control interface may make a very complex system much more controllable: only relatively few high level factors are involved in running the system, compared with monitoring and driving every little sub-process, even at the transistor level.

The history of computer science and software engineering since around 1950 shows how human engineers introduced more and more abstract and powerful virtual machines to help them design, implement, test debug, and run very complex systems.

When this happens the human designers of high level systems need to know less and less about the details of what happens when their programs run.

Making sure that high level designs produce appropriate low level processes is a separate task, e.g. for people writing compilers, device drivers, etc. Perhaps evolution produced a similar "division of labour"?

Similarly, biological virtual machines monitoring themselves would be aware of only a tiny subset of what is really going on and would have over-simplified information.

THAT CAN LEAD TO DISASTERS, BUT MOSTLY DOES NOT.

Importance for self-monitoring, self-control, self-management

Just as the use of VMs to provide a level of control helps human designers, to design, develop, debug, extend complex systems without always having to think about low level electronic processes that support all this, so also could the use of a VM help a complex control system to observe, debug, modify, and control itself.

Some computer scientists and AI researchers have appreciated the importance of this idea, and are investigating ways of giving machines more self awareness, in order to make them more intelligent.

John McCarthy, “Making robots conscious of their mental states”.
`http://www-formal.stanford.edu/jmc/consciousness.html`

Dave Clark at MIT ‘The knowledge layer’ in intelligent networks.

This simplification of control functions restricts the machine’s control sub-system to inspecting and changing only the virtual machine sub-systems.

This makes self-monitoring, self-control and self-debugging much simpler

it may also mean that the machine has incorrect or at least incomplete information about what is going on within it, though that may not matter in most contexts.

(Such machines would be just like humans in that respect!)

Robot philosophers

These inevitable over-simplifications in self-monitoring could lead robot-philosophers to produce confused philosophical theories about the mind-body relationship – e.g. theories about "qualia".

Intelligent robots will start thinking about these issues.

As science fiction writers have already pointed out, they may become as muddled as human philosophers.

So to protect our future robots from muddled thinking, we may have to teach them philosophy!

BUT WE HAD BETTER DEVELOP GOOD PHILOSOPHICAL THEORIES FIRST!

The proposal that a virtual machine is used as part of the control system goes further than the suggestion that a robot builds a high level model of itself, e.g. as proposed by Owen Holland in

`http://cswww.essex.ac.uk/staff/owen/adventure.ppt`

For more on robots becoming philosophers of different sorts see

Why Some Machines May Need Qualia and How They Can Have Them:
Including a Demanding New Turing Test for Robot Philosophers

`http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0705`
Paper for AAAI Fall Symposium, Washington, 2007

Some robot philosophers will get confused

Despite the important advantages of virtual machine architectures,

systems of that sort can get into deep muddles when they try to understand **themselves**

that leads to philosophical confusions, including confusions about qualia.

Intelligent machines (robot philosophers) will not be exempt.

Can we design them so that they get less confused than human philosophers about how they work?

Summary

1. What is a machine?
2. Different concepts of “virtual machine”
   including active or running VMs, in contrast with abstract VMs (mathematical entities).
3. Oversimplified notions of VM used by many philosophers versus richer notions
4. VM functionalism, supervenience and causation
5. Virtual machines are everywhere
6. Why VMs are important in engineering
7. Concurrent, interacting VMs sharing a substrate
8. Why the same considerations are relevant to biology
9. “Separation of concerns” (modularity) can be horizontal or vertical
10. Importance for self-monitoring, control, management
11. **Biological evolution probably “discovered” all this (and more) first**
12. Why the physical sciences have explanatory gaps
13. We don’t need quantum mechanics to explain the phenomena
14. Counterfactual conditionals and free will
15. Virtual machines that build themselves
16. Problems for biology, psychology, neuroscience, philosophy
17. Further reading

Biological evolution probably “discovered” all this (and more) first

Even though biological evolution does not need an intelligent designer to be involved, there are strategies that could be useful for evolution for the same reason as they are useful for designers.

That includes the use of virtual machines, for example.

- More precisely – it could turn out that a modification of a design for an organism that gives it a kind of self-understanding its competitors lack, could make it more successful.
- E.g. it may monitor its own reasoning, planning, and learning processes (at a certain level of abstraction) and find ways to improve them.
- If those improved procedures can also be taught, the benefits need not be rediscovered by chance.

Why the same considerations are relevant to biology

Conjecture: biological evolution "discovered" long ago that separating a virtual machine level from the physical level made it possible to use the VM as a platform on which variants could be explored and good ones chosen,

e.g. different behaviours, or different control mechanisms, different mechanisms for choosing goals or planning actions, or different mechanisms for learning things.

- Long before that, the usefulness of "horizontal" modularity had already been discovered, with different neural or other control subsystems coexisting and controlling different body parts, or producing different behaviours, e.g. eating, walking, breathing, circulating blood, repairing damaged tissue.
- But developing new parts with specific functions is different from developing new behaviours for the whole organism.
- If each new behaviour has to be implemented in terms of low level states of muscles and sensors that could be very restrictive, making things hard to change.
- But if a VM layer is available on which different control regimes could be implemented, the different regimes will have much simpler specifications.
- This allows one genome to support multiple possible development trajectories, depending on environment (as in altricial species).

  Conjecture: this allows common functionality to exist following different trajectories (in different individuals with that genome) e.g. doing mathematics or physics in English or Chinese?

Could such virtual machines run on brains?

We know that it can be very hard to control directly all the low level physical processes going on in a complex machine: so it can often be useful to introduce a virtual machine that is much simpler and easier to control.

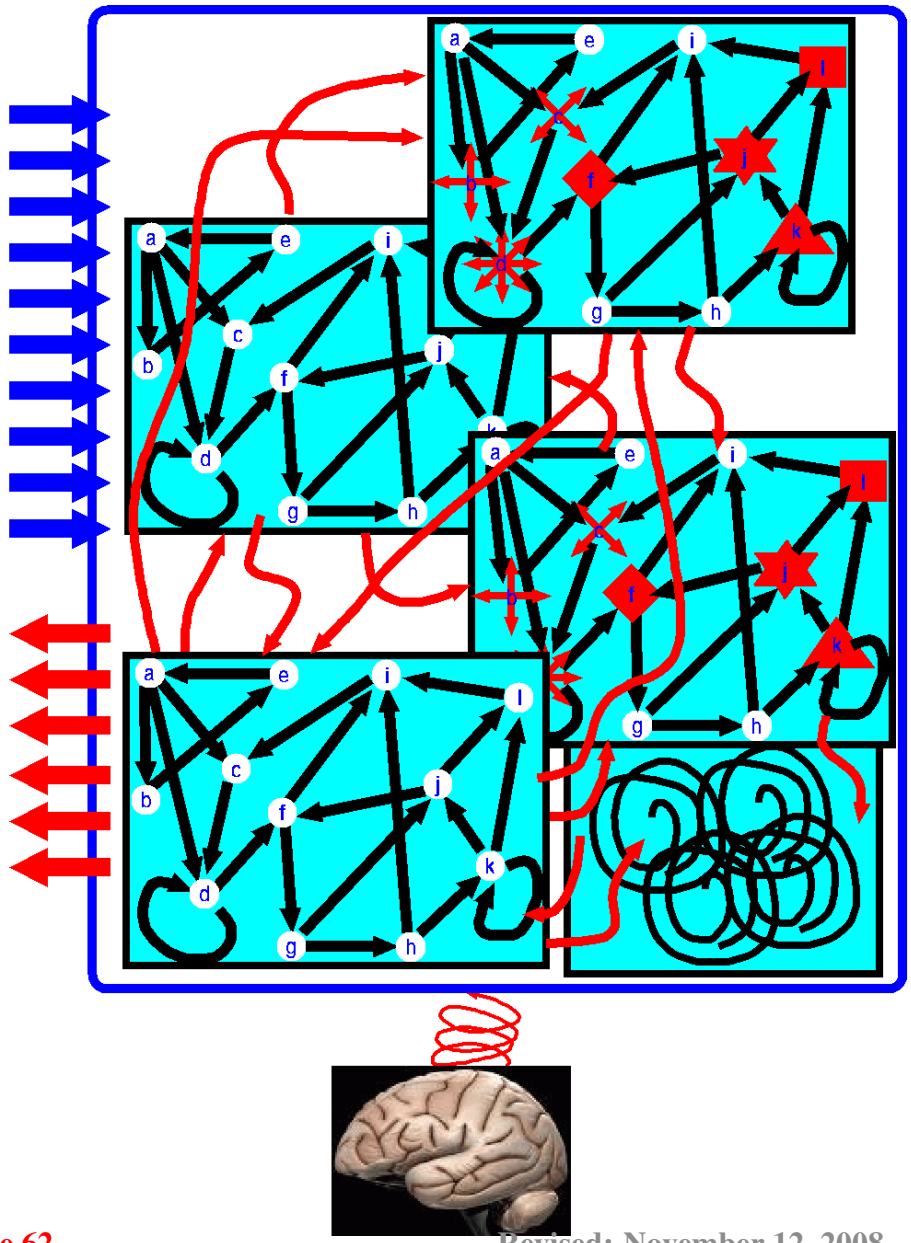
Perhaps evolution discovered the importance of using virtual machines to control very complex systems before we did?

In that case, virtual machines running on brains could provide a high level control interface.

Questions:

How would the genome specify construction of virtual machines?

Could there be things in DNA, or in epigenetic control systems, that we have not yet dreamed of?



Summary

1. What is a machine?
2. Different concepts of “virtual machine”
   including active or running VMs, in contrast with abstract VMs (mathematical entities).
3. Oversimplified notions of VM used by many philosophers versus richer notions
4. VM functionalism, supervenience and causation
5. Virtual machines are everywhere
6. Why VMs are important in engineering
7. Concurrent, interacting VMs sharing a substrate
8. Why the same considerations are relevant to biology
9. “Separation of concerns” (modularity) can be horizontal or vertical
10. Importance for self-monitoring, control, management
11. Biological evolution probably “discovered” all this (and more) first
12. **Why the physical sciences have explanatory gaps**
13. We don’t need quantum mechanics to explain the phenomena
14. Counterfactual conditionals and free will
15. Virtual machines that build themselves
16. Problems for biology, psychology, neuroscience, philosophy
17. Further reading

Why the physical sciences have explanatory gaps

In a complex system, how each of the parts behaves can depend in complex ways on what is going on in the rest of the system.

This can create the impression that what is happening is not determined by what has happened.

In a sense that is true: in any state of affairs, in any component of the whole system, various things **could** happen but what **actually happens** depends on the rest of the system (like the finite state machines described previously).

It is possible to say that of every portion of the system that ever behaved: at the time it **could** have done something different: it was free to do so, as far as its internal state was concerned.

This may lead incorrectly to the conclusion that physics has been proved to have explanatory gaps: the correct conclusion is that it is not the state of individual sub-systems that determines their future behaviour.

But if there are also virtual machines running, then some of the explanations of what is going on may not be expressible in the language of physics:

> e.g. “Three moves before the end, Black had lost because there was no way to prevent the next three moves by White forcing a checkmate”.

That is a real explanatory gap.

But the gap is not mysterious when we know how the chess VM works.

We don't need quantum mechanics to explain the phenomena

Compare the last few slides with what Henry Stapp writes about the alleged possibility, or even necessity, of human consciousness intervening in quantum phenomena.

Nothing I have written or said reports his main conclusion.

Counterfactual conditionals and free will

The above claims provide yet another way of looking at free will and also how easily it can be lost – if things go wrong

E.g. consider addictions.

(Some future robots may need psychotherapy.)

See also

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

Also slides on four concepts of free will, two of them junk, in
`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/`

Summary

1. What is a machine?
2. Different concepts of “virtual machine”
   including active or running VMs, in contrast with abstract VMs (mathematical entities).
3. Oversimplified notions of VM used by many philosophers versus richer notions
4. VM functionalism, supervenience and causation
5. Virtual machines are everywhere
6. Why VMs are important in engineering
7. Concurrent, interacting VMs sharing a substrate
8. Why the same considerations are relevant to biology
9. “Separation of concerns” (modularity) can be horizontal or vertical
10. Importance for self-monitoring, control, management
11. Biological evolution probably “discovered” all this (and more) first
12. Why the physical sciences have explanatory gaps
13. We don’t need quantum mechanics to explain the phenomena
14. Counterfactual conditionals and free will
15. **Virtual machines that build themselves**
16. Problems for biology, psychology, neuroscience, philosophy
17. Further reading

Virtual machines that build themselves

Jackie Chappell and I have been writing about differences between precocial and altricial species, with the suggestion that members of at least some altricial species are able to construct significant portions of their competences as a result of interactions with the environment.

In our IJCAI 2005 paper we argued that instead of treating species as precocial or altricial we should treat competences as precocial or altricial.

Every species will have a mixture of precocial and altricial competences.

Later we proposed a new terminology to avoid confusion, and talked about preconfigured and meta-configured competences.

- Preconfigured: mostly determined by genes plus aspects of the development process that are the same for all members of the species.
- Metaconfigured: determined by the genes plus aspects of the development process that can vary according to the environment and the individual's previous history of development.

See slides on meta-configured development.
`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/wonac/metaconfig.pdf`

A system with the ability to produce layers of meta-configured competences is unlike any monolithic learning system that goes on using the same mechanism (e.g. reinforcement learning, Bayesian learning) to add learnt information in a common memory store (e.g. a neural net of types studied by most neural learning theorists). Contrast "Cascade Correlation Nets".

See Baby stuff slides.

Virtual machines and biological evolution

Biological evolution seems to have "discovered" the need for virtual information-processing machines long before we did and probably has produced far more types than we have as yet seen the need for.

Self-monitoring, self-modifying, self-extending systems need to make use of virtual machines.

Challenges for neuroscience

- layers of virtual machines,
- how are they grown and when
- What sorts of causation to they allow?
  physiology $\rightarrow$ VM or vice versa also?
- How are they implemented?

  in particular what kinds of mappings exist between virtual and neural mechanisms, and how many different layers are used?
- Which aspects of the form or content of the various VMs depend on genetic information, and in what way?

Deep problems for theoretical biology

- how are specifications for virtual machines are represented in the genome?
- including virtual machines that construct new virtual machines?
- how did all this evolve ?

Problems for biology, psychology, neuroscience, philosophy

Previous slides have put forward what amount at present to theoretical conjectures – they are still lacking in precision in many respects and they also need to be related to empirical and theoretical work in other disciplines, including biology, psychology, neuroscience and philosophy.

The ideas also pose problems and challenges for those disciplines.

I hope the challenges can be met by new forms of interdisciplinary collaboration with different disciplines contributing:

- Philosophy: further conceptual clarification and revision of current philosophical theories to deal with the complexity of the VM/PM relationship.
- Psychology: taking seriously some of the conjectures about the contents of the VMs and how they develop and collecting empirical data that could both refine and extend those conjectures, or show how they need to be modified.
- Neuroscience: taking seriously the notion that there may be levels of functionality in brains that cannot be observed by normal physical and physiological investigations of brain structure and function, and looking for clues as to what virtual machines can be supported by known mechanisms (including chemical mechanisms, not just neural nets).
- Biology: collecting evidence from the evolution of genomes for humans and other intelligent species, and exploring developmental mechanisms relevant to the conjectures.

Is this talk really needed?

Various notions I thought were clear turned out not to be clear to everyone.

- Virtual machine
- Information (meaning, semantic content – not Shannon/Weaver)
- Information processing machine
- Information processing virtual machine
- Active (running) virtual machine
- Causation in virtual machines
- Virtual machine architecture
- Functions of components, states, or processes in an architecture
- Organisms as information processors
  (Biological information processing)
- Representation (information-bearer) and form of representation (notation, medium)
- Varieties of information states
  - belief-like states
  - desire-like states
  - perturbant (emotion-like) states
  - other control states

Why discussions are difficult

Discussions of the problems listed are difficult for a number of different reasons.

- We do not have nearly enough empirical knowledge

  about the sorts of things humans (of various ages), and other animals, can and cannot do,
  so, for instance, we think we know what vision is, or what understanding a sentence is, when we don't.

- We do not have agreed concepts
  - for describing different kinds of mental states, e.g. beliefs, desires, emotions, skills, knowledge, understanding.
  - for formulating explanatory theories, e.g about the kinds of mechanisms that explain the behaviours and mental states
    - for describing brain structures and mechanisms (physical and physiological machine architectures)
    - for describing mental structures and mechanisms (virtual machine architectures)

- We do not have good explanatory theories

  Because that would require us to have a good set of concepts and agreement on what they were and what was meant by theories using them.

  At present we don't even have general agreement on what is meant by describing portions of architectures as 'reactive', 'deliberative' or 'reflective', even though these labels are widely used, and this stops us having clear theories regarding architectures.

This presentation is about the need for a good ontology for talking about explanatory mechanisms and architectures.

|X| Elaboration: Spurious debates

The points listed previously are indications of conceptual confusions which produce what I regard as entirely spurious debates between rival factions in AI, cognitive science, neuroscience and philosophy.

- Many debates are spurious because people argue about whether some thesis (e.g. "brains compute", "brains process information", "emotions are necessary for intelligence", "machines cannot be conscious", "a foetus can feel pain") is true or false, without realising that the thesis is so ill-defined that the disputants interpret it differently, and one side argues for one unclear interpretation while the other side argues against another unclear interpretation, rendering the whole debate pointless (or premature).
- Careful conceptual analysis can help to reduce the confusions, by exposing implicit presuppositions and sometimes by revealing options that none of the disputants had considered.
- In particular, developments in computer science and software engineering since the mid 20th century have significantly extended our conceptual tools and the ontology now required for constructing deep explanatory theories – but many people are either completely ignorant of these advances or do not understand them and their implications: e.g. there are people who think that if they know the theory of Turing machines they know all important features of computers – yet they would fail miserably if asked to specify requirements for a modern operating system.
- Some of the confusions (e.g. taking the so-called "hard problem of consciousness" seriously in the context of scientific theorising) are partly of a different kind, based on incorrect philosophical theories, as discussed in other talks here:
  `http://www.cs.bham.ac.uk/research/cogaff/talks/`

For a brief introduction to conceptual analysis and spurious controversies see these slides on varieties of atheism:
`http://www.cs.bham.ac.uk/research/projects/cogaff/misc/varieties-of-atheism.html`

We need to extend our thinking capabilities and our ontologies

Many people are taught to think about

- Matter-manipulating machines
- Energy-manipulating machines

But they do not learn to think about

- Information-manipulating machines.

So they often fail to notice important questions and fail to consider important classes of possible answers: like neuroscientists who study neurons, and psychologists who study behaviour.

We are in the very early stages of learning to think about important age-old products of evolution:

- Virtual machines:
  - with real causal powers
  - e.g. decisions change what happens.
- Much concurrency:
  so that it can be misleading to ask what IT (or she or he) is doing, or can do, or notices, perceives, feels, etc.
  - The answers may be different for different parts of the same system.

Get rid of the idea that a Turing test can be useful

The notion of a “Turing test” as something that can determine what is going on inside a complex system, fails to take account of many of the possibilities for virtual machines described on previous slides

including VMs that include “disconnected” sub-systems.

Implications for testable theories

Virtual Machine Functionalism (VMF) implies that theories about systems using virtual machines can be very hard to test directly.

Instead we have to learn to work like physicists investigating sub-atomic entities, events and processes, where only very indirect testing is possible, and the most one can ever say of any theory is:

"This theory at present is better than any of its rivals"

It is always possible that a new, better, deeper, explanatory theory will turn up than we have discovered at any time, as happened when relativity and quantum mechanics replaced older theories.

This does not make truth relative, only very hard to discover.

Mental states and processes on this view are not mere "attributions" – they are real aspects of virtual machines.

Finding the right ontology for describing what's going on can be very hard: we still have much to learn about this.

Putting it all together

In the hope of reducing the confusion I have assembled these slides by collecting many partial explanations from papers and discussions over the last decade or so and modifying them in the light of what I've heard in recent debates. However:

- The issues are complex because the concepts used are not simple ones that can easily be defined explicitly.
- Moreover there are several different kinds of concepts involved, some relatively non-technical and widely understood, at least intuitively, others relatively technical and not well understood by most people.
- Some of the disputes depend on a view of computers that ignores the history that led up to them. For instance most of the key ideas were understood by Babbage and Lovelace long before the notions of Turing machine and equivalent mathematical notions had been thought of: computers are a recent development in a very old process of producing more and more sophisticated machines for controlling machines.
- Nowadays many of the controllers are virtual machines.
- It is also forgotten that computers were so-named because they were originally intended to take over a task that was previously done by humans, namely computing! (Likewise calculators performed a task previously done by humans.)

More importantly, living organisms have been processing information for millions of years.

What I mean by 'information' is explained partially later in this presentation and here:

`http://www.cs.bham.ac.uk/research/projects/cogaff/misc/whats-information.html`

Towards an ontology of 'mental' (i.e. VM) states

Our vocabulary for talking about virtual machines has two extremes:

- the (very rich and powerful but very hard to analyse) concepts of ordinary language used when we talk about ourselves and other people
- the much more impoverished but much more precise and well understood concepts of virtual machines used in software engineering and AI, which are not yet adequate for characterising biological systems.

We need to move towards something in-between, which is both precise and relevant both to organisms and machines, e.g. states differentiated in

The Architectural Basis of Affective States and Processes

`http://www.cs.bham.ac.uk/research/cogaff/03.html#200305`

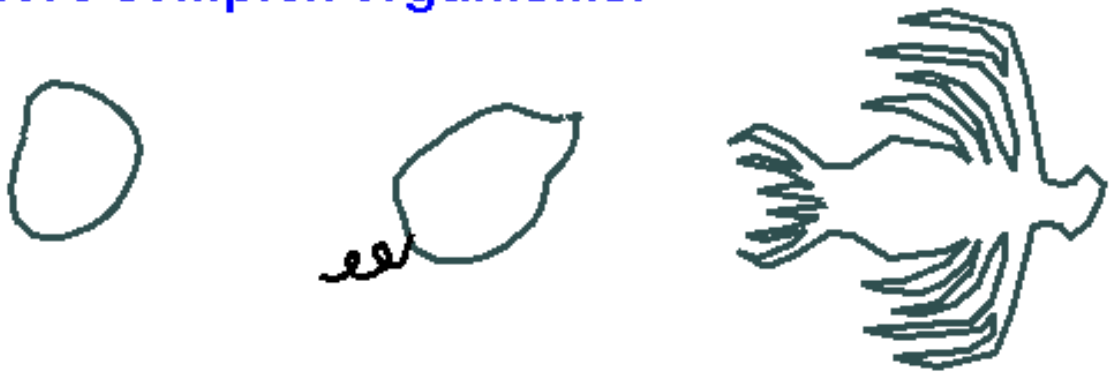
- Belief-like
- Desire-like
- Supposition-like
- Plan-like
- Moods and other varieties of affect
- initiation, termination, modulation, arbitration, evaluation ...
- Emotions as perturbances of one part by another

We can see the required variety of types of VM states by considering diverse biological organisms, from microbes to elephants.

A biological perspective

Once upon a time there were only inorganic things: atoms, molecules, rocks, planets, stars, etc. These merely reacted to *resultants* of all the physical forces acting on them.

Later, there were simple organisms. And then more and more complex organisms.



These organisms had the ability to reproduce. More interesting was their ability to *initiate* action, and to *select* responses, instead of simply being pushed around by physical forces acting on them.

That achievement required the ability to acquire, process, and use *information*.

The ability to act or to select requires information

E.g. organisms can use information about

- density gradients of nutrients in the primaeval soup
- the presence of noxious entities
- where the gap is in a barrier
- precise locations of branches in a tree as you fly through
- how much of your nest you have built so far
- which part should be extended next
- where the nest is, or where a potential mate is
- something that might eat you
- the grass on the other side of the hill
- what another animal is likely to do next
- how to achieve or avoid various states
- how you thought about two problems, one solved the other not
- whether your thinking is making progress ... and much, much more...

All this requires that organisms contain an energy store which can be deployed to meet their requirements, unlike most physical objects whose behaviour is determined only by external forces.

In a bouncing ball, elastic energy is temporarily stored, put there by physical forces, then released in a manner that has nothing to do with a need for survival of the ball. The ball uses no information: it has no needs or purposes — It takes no steps to survive or reproduce.

The notion of need

- Making all that precise requires the notion of a need and a process or mechanism that serves the need.
- The existence of such things amounts to the truth of very complex sets of counterfactual conditional statements
  - About what would or would not happen in various circumstances if the need were not satisfied.
  - About what would or would not happen in various circumstances if the need-serving process or mechanism did not exist or were modified in some way.

The evolution of information-processing

Over time, as organisms became more complex, their use of information became more complex.

- Instead of reacting immediately to sensed states and events, some evolved the ability to take in information and use it later, e.g. going back to a location where food had been perceived.
- Some evolved the ability to make their reactions to particular sensed stimuli depend on internally sensed states of need.
- Some evolved the ability to allow more than one reaction to be triggered simultaneously and to use sensed or stored information influence the choice when the reactions are incompatible.
- Some evolved the ability to react to derived information, e.g. inferring the presence of a predator nearby and reacting to the derived information.
- Some developed the ability to acquire, store and use, possibly much later, generalisations about things in the world.
- Some developed the additional ability to derive and compare two or more predictions or plans, compare them and then select one. This required means of encoding hypotheticals.
- Some developed the ability to acquire and use information about their own information-processing, or information about the information-processing done by other individuals, e.g. predators, prey and neutral individuals

Some qualitative changes

Many assume biological evolution is a continuous process: but it cannot be (a) because DNA cannot change continuously - molecules are discrete structures, and (b) because there are only a finite number of generations between any two states.

- One of the important qualitative changes involved being able to discretise or chunk information: this is necessary to explore branching sets of possibilities, whether for exploring alternative sequences of action in making a plan, or exploring alternative sequences of other kinds in making predictions, or exploring alternative explanations for observed facts.
- That change led to requirements for new processes of perception, new forms of information storage, new kinds of temporary work-spaces, new ways of managing decisions.
- Another kind of qualitative change was development of means of acquiring and using information about the activities of an information user, whether oneself or another individual. This required an extension of the ontology beyond what was adequate for expressing information about physical objects and their interactions in the environment.
- We still do not know enough the requirements for these changes, nor about the possible kinds of mechanisms that can support them, nor which kinds of architectures can combine these and other kinds of information-processing. (But we know much more than we knew a hundred years ago.)

Varieties of biological information-processing

Different animals (microbes, insects, fishes, reptiles, birds, mammals, etc.) clearly differ in their requirements and their capabilities.

It would be helpful to attempt a survey of "dimensions" in which such capabilities can vary, and the kinds of designs that can support the different varieties.

This would be part of a general theory of information – what it is and how it works.

One of the kinds of dimensions would be concerned with the sort of *content* of the information.

- Some information is very localised and simple (here's a dot, there's some motion to the left).
- Other information is far more holistic (e.g. recognising a scene as involving a forest glade).
- Some may be very abstract (the weather looks fine; it looks as if a fight is about to break out in that crowd).
- Some information items contain generally applicable knowledge, e.g. about the geometry and topology of static and moving shapes: e.g. regular hexagons can be packed to fill a convex space.
- Others involve specific facts relevant only in a particular part of the world, e.g. the Eiffel tower is in Paris.
- Some items of information are "categorical" others "hypothetical" or counterfactual, e.g. you would have been killed by that car had you not jumped out of its way.

Other modes of variation are concerned with the medium used and the formal or syntactic properties of the medium.

Resist the urge to ask for a **DEFINITION** of "information"

Compare "energy" – the concept has grown much since the time of Newton. Did he understand what energy is?

Instead of *defining* "information" we need to analyse kinds of processes in which it can be involved, the kinds of effects it can have, and the kinds of mechanisms required, i.e. such things as

- the variety of types of information there are,
- the kinds of forms they can take.
- the variety of means of acquiring information,
- the means of manipulating information,
- the means of storing or transmitting information,
- the means of communicating information,
- the purposes for which information can be used,
- the variety of ways of using information.

  Examples of all of these will be given later

As we learn more about such things, our concept of "information" grows deeper and richer: Like many deep concepts in science (including "energy" and "matter"), the concept of "information" is mostly *implicitly* defined by its role in our theories and our designs for working systems.

Compare "information" and "energy"

It is also hard to define "energy" in a completely general way.

Did Newton understand the concept "energy"?

There are kinds of energy he did not know about:

- chemical energy
- electromagnetic energy, ... etc.

Why were these called "energy"? The theory that energy is *conserved* was crucial.

We can best think of energy in terms of:

- the different forms it can take,
- the ways in which it can be
  - acquired
  - transformed,
  - stored,
  - transmitted,
  - used, etc.
- the kinds of causes and effects that energy transformations have,
- the many different kinds of machines that can manipulate energy
- ....

If we understand all that, then we don't need to *define* "energy" – at least not by specifying its meaning in terms of ways of testing or measuring the presence of energy.

It is a primitive theoretical term – implicitly defined by the processes, relationships and mechanisms that involve it.

How not to define deep theoretical concepts

Newton knew about energy, but did not know anything about the energy in mass:

The possibility of $E = MC^2$ had not been thought of.
(This partially transformed both the concepts "energy" and "mass".)

We should not use currently known forms of energy or current ways of measuring energy to *define* it, since new forms of energy may turn up in future, along with new types of measurement.

(Partial changes to the theory partially change the concepts.)

This is typical of deep scientific concepts: they are to a large extent implicitly defined by the theories in which they are used, and cannot be explicitly defined in terms of pre-theoretical concepts or types of measurements or observations.

Any such definitions ("operational definitions") would omit central features of the concepts, namely their structural and causal connections within the theory.

All this is familiar to philosophers of science, but not always understood by scientists, especially those who think physics and chemistry are merely about laws relating observables.

A related confusion is the wide-spread "concept empiricist" belief that all concepts must somehow be abstracted from experience, sometimes labelled the theory of "symbol grounding". Concept empiricism (and therefore symbol grounding theory) was demolished long ago by Immanuel Kant.

See `http://www.cs.bham.ac.uk/research/cogaff/talks/#models`
Introduction to key ideas of semantic models, implicit definitions and symbol tethering
and `http://www.cs.bham.ac.uk/research/cogaff/talks/#talk14`
"Getting meaning off the ground: symbol grounding vs symbol tethering"

Contrast Shannon's notion of "information"

We are not using Shannon's syntactic notion of "information" which refers to statistical properties of possible collections of symbols.

We are using something closer to the colloquial notion of "information" as

- meaning
- reference
- semantic content

which requires there to be

1. a user or interpreter of the meaning (recipient, in the case of a message)
2. a bearer, or encoding, of the meaning (a picture, sentence, dance, wave pattern, electronic state of a memory chip, etc.)
3. sometimes, but not always, there is a source of the encoding (e.g. sender of a message) (Source, or creator, and recipient or user, are often one thing.)
4. something which is expressed or referred to (the content)
   (Mill, Frege and others distinguished two aspects: sense/connotation/intension and reference/denotation/extension)

Note:
Some "information-bearers" are physical (e.g. marks on paper), but often the bearer is a structure or process in a virtual machine. E.g. a network data-structure in a computational virtual machine could encode, for that machine, information about a network of roads, used by a route-finder.

Differences between energy and information

We are not using a *quantitative* notion of information

One big difference between energy and information (in the sense used here):

It is very useful to *measure* energy e.g. because it is conserved.

Expressing information as a numerical quantity is often of no use.

Numbers describing information (measurements) are *sometimes* useful

(e.g. if one message contains information about three people
and another contains additional information about a fourth person).

But numbers do not capture what is most important about information, for behaving systems:

Numbers don't express where something is (e.g. in a drawer), what it is, how it is related to other things, where it comes from, what it can do, who made it, what the implications of something are, etc.

Further differences:

- If I give you information I may still have it, unlike energy.
- You can derive new information from old, and still have both, unlike energy.
- Information varies primarily not in its *amount*, like energy, but in its structure and content: numeric equations do not represent most information manipulations adequately.
  (Compare chemical equations, parse trees, maps, flow-charts.)
- Energy in a physical object is there independently of whether any machine or organism takes account of it, whereas the information expressed or conveyed by something depends on the information-processing capabilities of the user or perceiver: information (in the sense we are using) is inherently relational.

Being relational does not imply being subjective

- Whether a jacket J is a good fit depends on who the wearer is.
  So being a good fit is a relational property.
- But if X is a particular person, then whether J is a good fit for X is not a relational property.
- Neither is it merely something arbitrarily attributed to J by perceivers.

- Likewise what information a particular information-bearer expresses will depend on who is attending to the information.
- However potential information content for different sorts of perceivers is an objective property: so

  > the statement that an object O can convey information I to agents with certain kinds of information processing capabilities C is not just an arbitrary or subjective attribution:
  > it's a fact about the relationship between features of O and I and C

- Checking its truth may be very difficult however.

Things that can be done with information

Part of an analysis of the notion of "information" is provided by a taxonomy of types of things that can be done with information, by a user or perceiver X:

- X can react immediately (the information can trigger immediate action, external or internal)
- X can do segmenting, clustering, labelling of components within a complex information structure (i.e. do parsing)
- X can try to derive new information from something (e.g. what caused this? what else is there? what might happen next? can I benefit from this?)
- X can store the information for future use (and possibly modify it later).
- X can use the information in considering alternative next events, in making predictions.
- X can use information in considering alternative next actions, in making plans
- If X interprets some information as containing instructions, X can obey them, e.g. carrying out a plan.
- The information can express one or more of X's goals, preferences, ideals, attitudes, etc.
- X can observe itself doing some or all of the above and derive new information from that (self-monitoring, meta-management).
- X can communicate the information to others (or to itself later)
- X can check information for consistency, either internal or external
- X can check information for correctness (truth), precision, relevance, ...

and more ... using different forms of representation for different purposes.

Sentences, lists, arrays, metrical maps, topological maps, pictures, 3-D working models, weights in a neural net, structures of complex molecules, data structures in a computer, gestures, etc.

Diverse mechanisms of varying sophistication

Extracting information from basic sensory data may require very different perceptual mechanisms with varying sophistication.

- Some information can be extracted very simply (using spatial or temporal local change detectors, or mechanisms for constructing histograms of features, such as colour, texture, optic flow).
- Other information may need *relationships* to be discovered between features, e.g. collinearity, lying on a circular arc, parallelism, closure, lying on the intersection of the continuations of two linear segments or two curved segments (where the continuations are also curved).
- Sometimes this requires *searching* for coherent interpretations.
- Some relationships hold only between abstract entities not the image data: e.g. two people seen to be *looking in the same direction*.
- Extracting some of the information requires matching with known models ("That's a triangle, a face, a tree").
- Some learning tasks require noticing new repeated structures within the information structures (e.g. noticing repeated occurrence of polygons with circles at two adjacent corners).

For different kinds of sensory interpretation tasks, different forms of representation are often useful, and different types of processing.

There are different kinds of information

For instance:

- about categories of things (big, small, red, blue, prey, predator)
- about generalisations (big things are harder to pick up)
- about particular things (that thing is heavy)
- about priorities (it is better to X than to Y)
- about what to do (run! fight! freeze! look! attend! decide now!)
- about how to do things (find a tree, jump onto it, climb...)

This categorisation of types of information does not cover all the types found in machines and organisms.
Some of the differences are differences in “pragmatic function” rather than “semantic content”.
We probably still know only about a small subset of types of information, types of encoding, and types of uses of information.
Don’t expect all types to be expressible in languages we can understand – e.g. what a fly sees, or what a bee expresses in a dance!
Or even what a chimp, or a human child sees
We often tend to ask whether an animal can learn that so and so without considering the the implications of the possibility that nothing the animal is capable of learning is expressible in a human language or thinkable in a human mental architecture.

Further aspects of a theory of information.

We need to understand other ways in which information-processing events can vary.

E.g. besides

- Different information contents, and
- the different forms in which they can be expressed,

there are further functional and causal differences:

- the different ways information can be acquired, transformed, stored, searched, transmitted, combined or used,
- the kinds of causes that produce events involving information,
- the kinds of effects information manipulation can have,
- the many different kinds of machines that can manipulate information,

If we understand all that, then we don't need to *define* "information"!

See also

`http://www.cs.bham.ac.uk/research/projects/cogaff/misc/whats-information.html`

Examples of types of processes involving information

- Acquisition
- Filtering/selecting
- Transforming/interpreting/disambiguating
- Compressing/generalising/abstracting
- Deriving (making inferences, but not only using propositions)
- Storage/Retrieval (many forms: exact, pattern-based, fuzzy)
- Training, adaptation (e.g. modifying weights, inducing rules)
- Constructing (e.g. descriptions of new situations or actions)
- Comparing and describing information (meta-information)
- Reorganising (e.g. formation of new ontologies)
- Testing/interrogating (is X in Y, is A above B, what is the P of Q?)
- Copying/replicating
- Syntactic manipulation of information-bearing structures
- Translating between forms, e.g. propositions, diagrams, weights
- Controlling/triggering/modulating behaviour (internal, external)
- Propagating (e.g. in a semantic net, or neural net)
- Transmitting/communicating
- .... (many more)

The differences involve types of content, types of medium used, and the causal and functional relations between the processes and their precursors and successors.

Requirements for Information Processing

Not all the processes listed previously are possible in all architectures.

E.g. constructing and comparing descriptions of possible future actions, needs a "workspace" for items of varying complexity.

Some kinds of neural net require mechanisms supporting continuous variation.

Some kinds of manipulation require an engine able to construct and manipulate "Fregean" structures, with hierarchic function plus arguments decomposition.
(E.g. f(g(a, h(b,c)), h(d,e)))

We must distinguish requirements specified (a) in terms of a virtual machine architecture (b) in terms of physical mechanisms.

A VM SPECIFICATION might mention a strict stack discipline for procedure activations, with local variables and return address in each stack frame.

A PHYSICAL SPECIFICATION might mention fast special purpose registers, etc.

How much the properties of a particular VM can be decoupled from properties of the physical implementation will vary.

How much of a VM is implemented in the "external" environment will vary. (E.g. pheromone trails used by insects.)

An information-processing architecture includes

- forms of representation,
- algorithms,
- concurrently processing sub-systems,
- connections between them

It need not be a rigidly fixed system: some architectures can modify themselves, e.g.

- a unix system that can spawn new processes that can spawn new processes, or
- a child's mind.

We need to understand the space of information processing architectures ("design space") and the states and processes they can support, including:

- The variety of types of perception
- The variety of types of reasoning
- The variety of types of emotions
- The varieties of types consciousness
- ...

Varieties of information-processing architectures in organisms

Not all organisms can do all the things listed previously.

- Everyone knows that organisms can differ in their size, their physiology, their habitats. their behaviours, their social organisation.
- Many researchers do comparative studies, and discuss how these things evolved.
- Differences in their information-processing functions and architectures and how they evolved are not acknowledged to the same extent.
- E.g. the chapter on evolution of memory in S.Rose *The making of memory*, 1993, (excellent book) is mainly about evolution of physiological mechanisms and behaviours.
- Rose, like many others, seems to think that "information processing" refers only to what computers viewed as bit manipulators do, apparently unaware that even in computers there are many varieties of information processing in different sorts of virtual machines.

  Such views obstruct attempts to study natural information processing architectures and their evolutionary and developmental trajectories.

A first draft ontology for architectural components

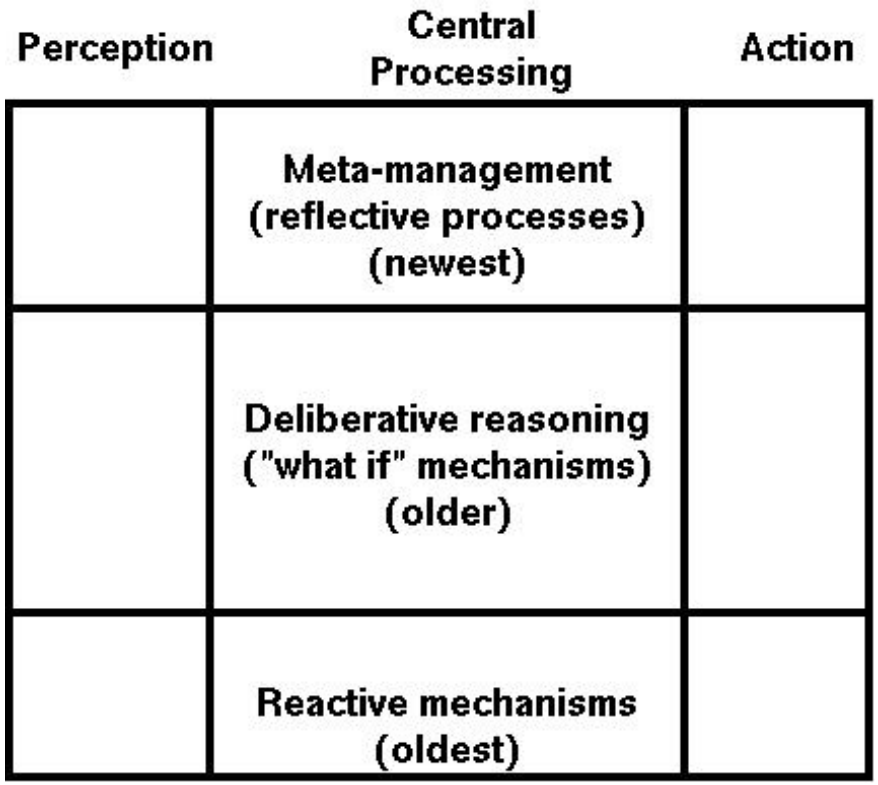
THE COGAFF ARCHITECTURE SCHEMA

For now let's pretend we understand the labels in the diagram.

On that assumption the diagram defines a space of possible information-processing architectures for integrated agents, depending on what is in the various boxes and how the components are connected, and what their functions are.

So if we can agree on what the types of layers are, and on what the divisions between perceptual, central and motor systems are, we have a language for specifying functional subdivisions of a large collection of possible architectures, ...

even if all the divisions are partly blurred or the categories overlap.



Note: Marvin Minsky's draft book *The emotion machine* uses finer-grained horizontal division (six layers). There's largely because he divides some of these cogaff categories into sub-categories, e.g. different sorts of reactive mechanisms, different sorts of reflective mechanisms.

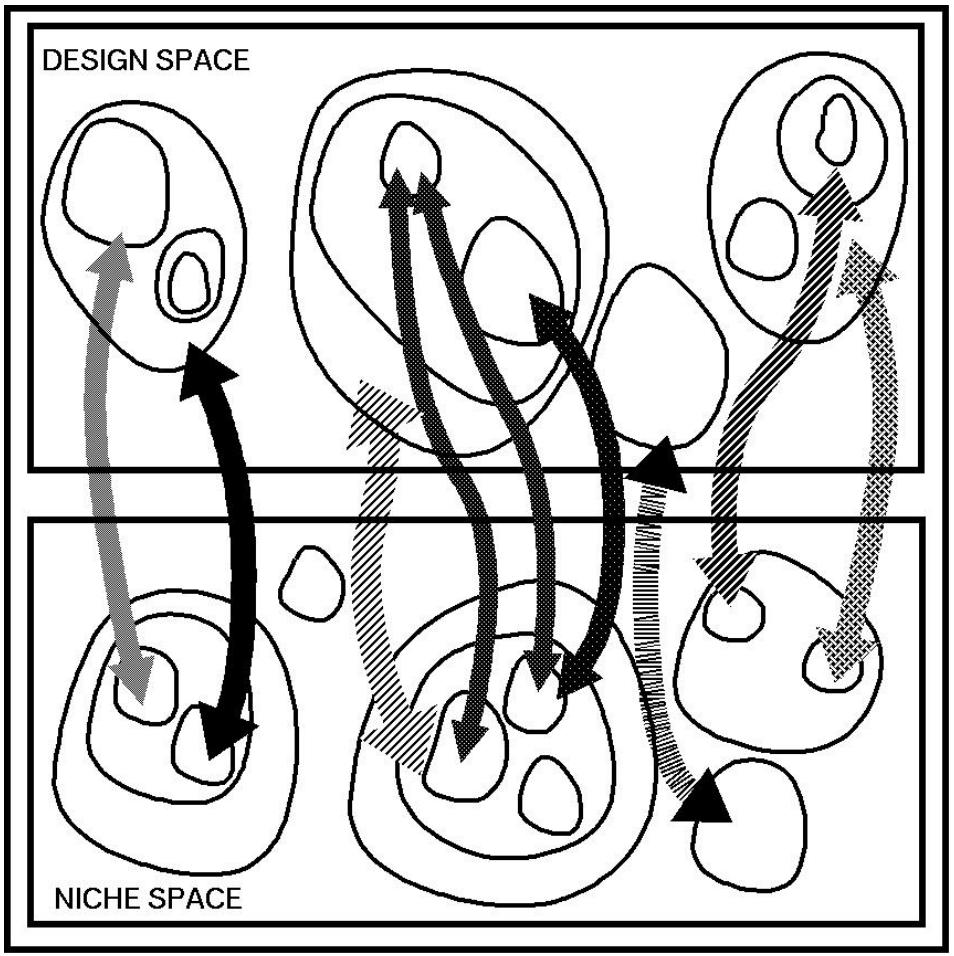
The CogAff Schema is mainly about virtual machine architectures

- Some of the lower level reactive functions could be directly provided by physical devices, e.g. sensory and motor transducers, thermostats, trigger mechanisms, threshold devices that operate relays, etc.
  E.g.: V. Braitenberg, (1984), *Vehicles: Experiments in Synthetic Psychology,* The MIT Press
- However, many of the functions require construction of rapidly changing information structures whose complexity varies over time (e.g. visual percepts, plans, hypotheses) and since neither brains nor computers can constantly and rapidly reorganise their physical structure, the functions in question must be provided by rapidly changing virtual machine structures.
- These virtual machines are ultimately implemented in physical machines whose behaviour is intrinsically reactive: physical processes do not think about what might be done, or what might have happened, or what might be out of sight around the corner.
- Some of the higher level non-reactive virtual machines may be implemented in intermediate level reactive virtual machines, for instance when a planning system is implemented in a symbol-manipulating mechanism which manipulates symbols in a virtual machine.

  Many symbolic AI systems are virtual machines implemented in list-processing virtual machines, implemented in virtual machines like pentiums and sparcs, implemented in digital electronic devices.

This is part of a study of relations between "design-space" and "niche-space"

Instances of designs and niches (sets of requirements) are also interacting virtual machines.

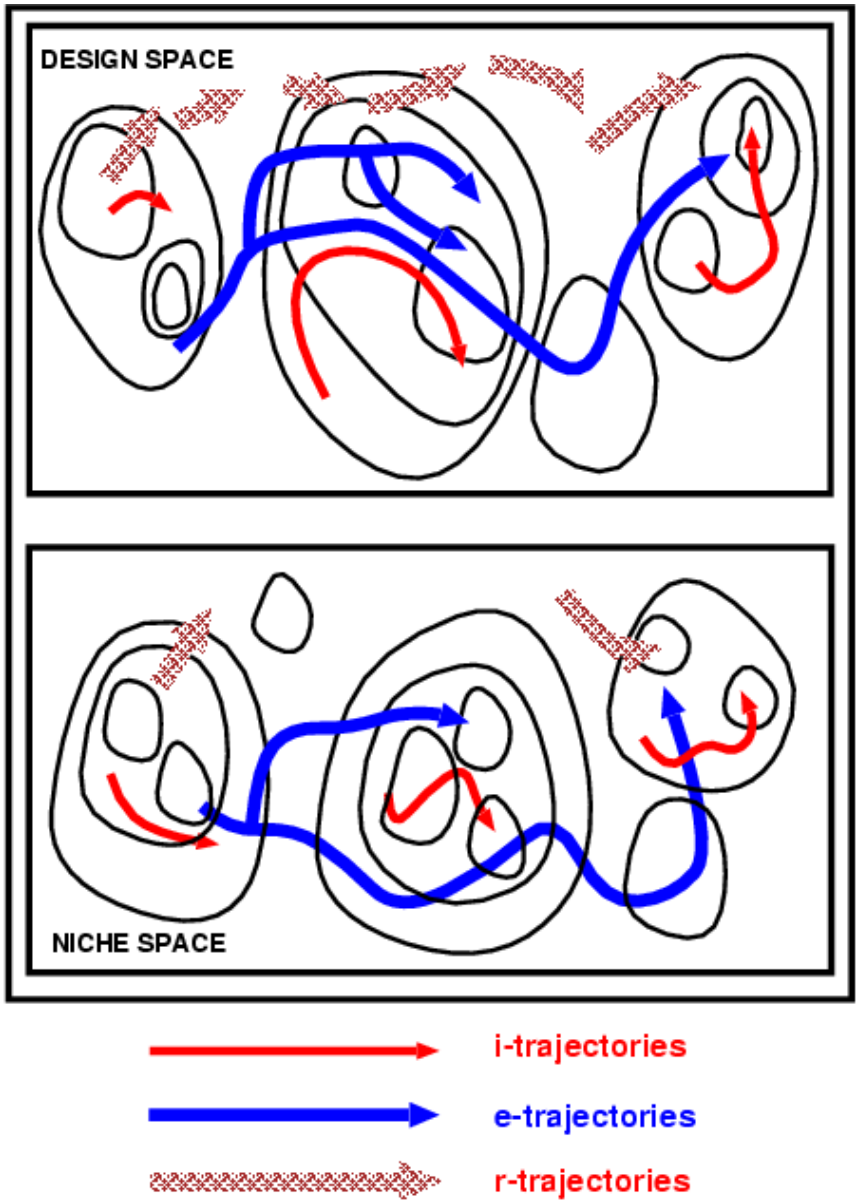


There are (many) fitness relationships — not fitness functions.

And trajectories in both spaces

Various interacting trajectories are possible in design space and niche space: dynamics of biological virtual machines in an ecosystem.

- i-trajectories: individuals develop and learn
- e-trajectories: species evolve across generations
- r-trajectories: a 'repairer' takes things apart and alters them
- s-trajectories: societies and cultures develop (Not shown)
- c-trajectories: e-trajectories where the cognitive mechanisms and processes in the individuals influence the trajectory, as in mate selection, or adults choosing which offspring to foster in times of shortage. (Also not shown)

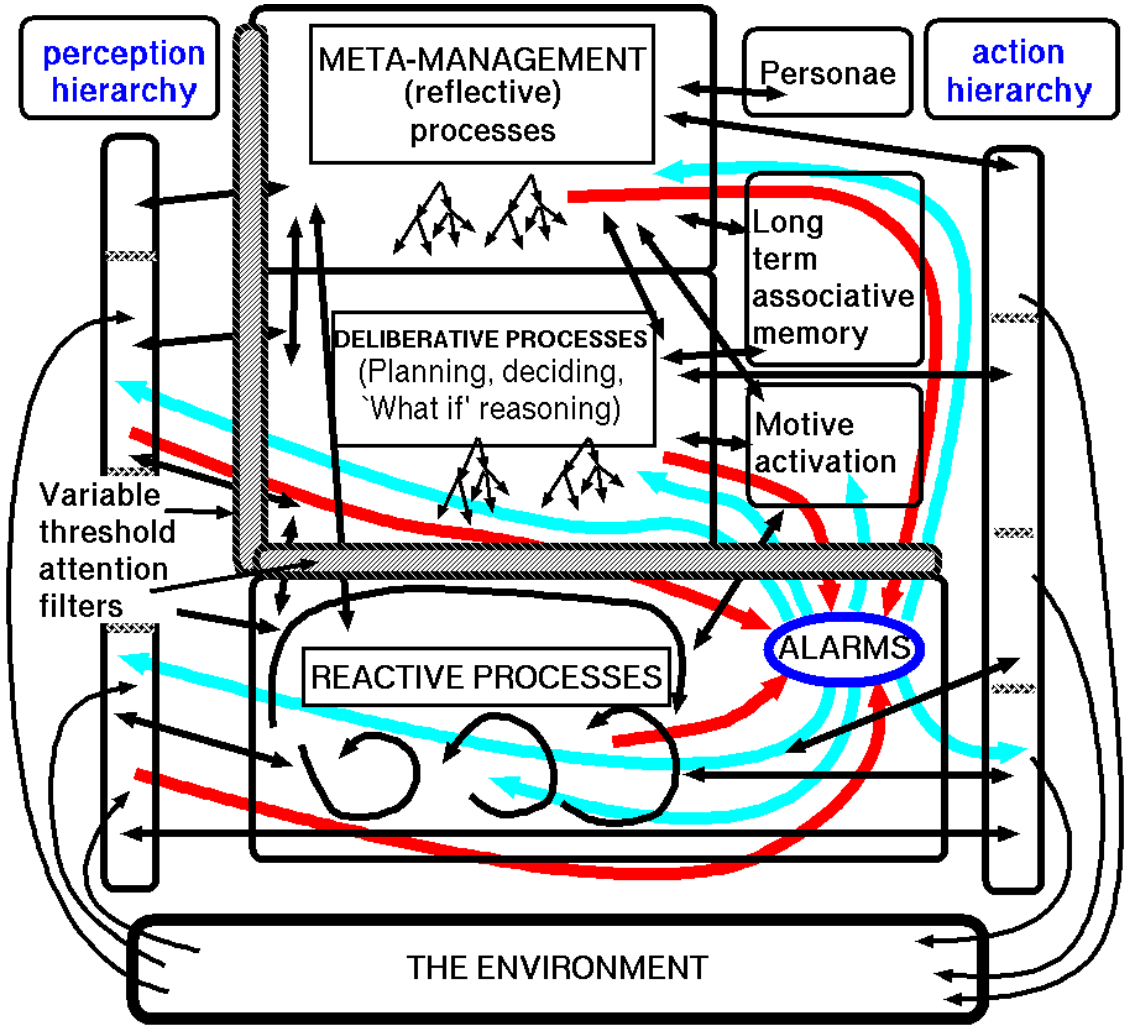


What I am heading towards: H-Cogaff

The H-Cogaff (Human Cogaff) architecture is a (conjectured) special case of the CogAff schema, containing many different sorts of concurrently active mutually interacting components.

The papers and presentations on the Cognition & Affect web site give more information about the functional subdivisions in the proposed (but still very sketchy) H-Cogaff architecture, and show how many different kinds of familiar states (e.g. several varieties of emotions) could arise in such an architecture.

This is shown here merely as an indication of the kind of complexity we can expect to find in some virtual machine architectures for both naturally occurring (e.g. in humans and perhaps some other animals) and artificial (e.g. in intelligent robots).



The conjectured H-Cogaff (Human-Cogaff) architecture

See the web site: `http://www.cs.bham.ac.uk/research/cogaff/`

Further reading: Background to these slides

For many years, like many other scientists, engineers and philosophers, I have been writing and talking about "information-processing" systems, mechanisms, architectures, models and explanations, e.g.:

My 1978 book *The Computer Revolution in Philosophy*, now online here:
`http://www.cs.bham.ac.uk/research/cogaff/crp/` (especially chapters 6 and 10)

A. Sloman, (1993) 'The mind as a control system,' in *Philosophy and the Cognitive Sciences,* Cambridge University Press, Eds. C. Hookway & D. Peterson, pp. 69–110.
Online here: `http://www.cs.bham.ac.uk/research/cogaff/`

Since the word "information" and the phrase "information-processing" are both widely used in the sense in which I was using them, I presumed that I did not need to explain what I meant. Alas I was naively mistaken:

- Not everyone agrees with many things now often taken as obvious, for instance that all organisms process information.
- Some people think that "information-processing" refers to the manipulation of bit patterns in computers.
- Not everyone believes information can cause things to happen.
- Some people think that talk of "information-processing" involves unfounded assumptions about the use of representations.
- There is much confusion about what "computation" means, what its relation to information is, and whether organisms in general or brains in particular do it or need to do it.
- Some of the confusion is caused by conceptual unclarity about virtual machines, and blindness to their ubiquity.

Further Reading

A very stimulating and thought provoking book overlapping with a lot of this presentation is

George B. Dyson *Darwin among the machines: The Evolution Of Global Intelligence* 1997, Addison-Wesley.

Papers and presentations on the Cognition and Affect & CoSy web sites expand on these issues, e.g.

- A. Sloman & R.L. Chrisley, (2003),
  Virtual machines and consciousness, in *Journal of Consciousness Studies*, 10, 4-5, pp. 113–172,
  `http://www.cs.bham.ac.uk/research/cogaff/03.html#200302`
- A. Sloman, R.L. Chrisley & M. Scheutz,
  The Architectural Basis of Affective States and Processes, in *Who Needs Emotions?: The Brain Meets the Robot,* Eds. M. Arbib & J-M. Fellous, Oxford University Press, Oxford, New York, 2005.
  `http://www.cs.bham.ac.uk/research/cogaff/03.html#200305`
- A. Sloman and R. L. Chrisley,
  More things than are dreamt of in your biology: Information-processing in biologically-inspired robots, *Cognitive Systems Research*, 6, 2, pp 145–174, 2005,
  `http://www.cs.bham.ac.uk/research/cogaff/04.html#cogsys`
- A. Sloman
  The well designed young mathematician. In *Artificial Intelligence* (2008 In Press.)
  `http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0807`
- "What's information?"
  `http://www.cs.bham.ac.uk/research/projects/cogaff/misc/whats-information.html`
- Presentations `http://www.cs.bham.ac.uk/research/projects/cogaff/talks/`

M. A. Boden, 2006, *Mind As Machine: A history of Cognitive Science* (2 Vols), Oxford University Press

There are many other books in philosophy of mind and cognitive science.

For more on all this

There is a lot more on all of this in the Cognition and Affect Project papers and talks:

```
http://www.cs.bham.ac.uk/research/cogaff/
http://www.cs.bham.ac.uk/research/cogaff/talks/
```

In particular the Tutorial presentation by Matthias Scheutz and myself on Philosophy of AI at IJCAI'01 discusses objections to the notion that events in virtual machines can be causes.

```
http://www.cs.bham.ac.uk/research/cogaff/talks/#talk5
```