DRAFT

A paper growing out of a set of slides presented at A seminar on Thurs 12th October 2000

What are virtual machines? Are they real?

Aaron Sloman

The University of Birmingham

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

With much help from Matthias Scheutz

Contents

1 Background

It is now fairly common to draw an analogy between two relations, namely the relationship between minds and brains (or more generally the relationship between mental phenomena and physical phenomena), and the relationship between virtual machines in computers and the underlying physical computer and relevant parts of the physical environment.

$$\textsc{Mind} \longrightarrow \textsc{Brain}$$
$$\textsc{Virtual machine} \longrightarrow \textsc{Physical Machine}$$

The first relation is often referred (by philosophers) to as “supervenience”. I.e. minds supervene on brains, mental phenomena supervene on physical phenomena. The second relation, between phenomena in a virtual machine (VM) and the physical phenomena which make the VM possible, is often referred to (by computer scientists and software engineers) as “implementation”, or “realization”, or “support”. Sometimes the first relation is also referred to as “realization”: i.e. brains may be said to realize minds.

This paper investigates the claim that both relations have much in common, and, moreover, both are examples of a much more general relationship between “levels” in reality. A full defence of the claim would be beyond the scope of this paper. Instead the aim is first of all to analyse or explicate a fairly strong version of the claim, by contrasting it with other possible claims about the relationship between mind and matter, and secondly to state, and partially counter, an objection to the claim based on puzzles about how mental, or virtual machine events, can be causes of other events. A full rebuttal of the argument about causation would require a comprehensive analysis of theories of causation, a book-length task. The rebuttal offered here will be only a sketch of an analysis of causation, showing only that there is a plausible view of causation that rebuts the objection. For those who do not find the outline analysis of causation remotely plausible this paper will be at best a formulation of a hypothesis requiring discussion and argument.

A full understanding of the problem, and the requirements for an adequate solution to the problem, depends in part on an understanding of the nature of computation. The concept “computation” has a number of interpretations, one of which refers to a very abstract, essentially *syntactic* notion concerned with structures and relations between structures. That is the notion that is analysed in terms of abstract constructs such as Turing machine, production system, recursive functions. There is another notion of computation that is closer to what designers and users of computers are concerned with, and that involves events, processes, and causal interactions within a machine that is concerned with information processing, usually for some practical purpose, for instance, controlling an external machine, analysing information, production of documents, design of machinery of various kinds. This latter notion of a machine has its origins not in the abstract mathematical notion at the centre of theoretical computer science, but in the history of machines for controlling other machines, e.g. mechanical looms, card sorting machines, mechanical toys, and also numerical calculators.

These machines are concerned with acquiring, storing, transforming, interpreting, and using information. Such processes are not physical processes: they are processes concerned with interactions within and between non-physical entities, e.g. numbers, words, rules, images, procedures, etc. However they are *implemented* in and dependent on physical machines. This relation, of implementation, is the key notion that we need to analyse and understand. We can then take a new look at the philosophers notion of supervenience and modify it to provide an adequate characterisation of the relation between mind and brain.

In short, the key feature of computation on which we focus is the processing of information, often to some practical purpose. The key notion of mind is also the processing of information, in percepts, beliefs, desires, memories, skills, hopes, fears, etc.

2 The implementation relation in computing

The second relation, between virtual machines and computers, which I shall here refer to as “implementation” is well understood intuitively by software engineers (and some computer scientists), since they regularly design, create, use, modify, analyse and debug such systems.

Often the implementation involves several layers: with one VM implemented in another, which is implemented in a physical system. E.g. the Java VM running on a computer is often implemented in a sparc or pentium or alpha, or HPPA, each of which is a virtual machine implemented in a digital circuit which is implemented in physical materials.

Philosophers usually discuss supervenience in complete ignorance of what software engineers know or do. As shown below, some of the criteria proposed as requirements for supervenience are violated by the relations found between VMs and the machines in which they are implemented. If supervenience and implementation are in fact the same relationship, then this provides a counter-argument to the proposed criteria.

However, engineers (like most people who are not trained in philosophy) often cannot articulate what it is that they know and do intuitively. I conclude that philosophers and engineers should help one another with the task of clarifying the relationships.

3 What are the terms of the supervenience relation?

Many philosophers have written on the supervenience relation, whether they used the term or not, including Chalmers (1996), Dennett (1991), and the works of Jaegwon Kim (Kim 1993, 1998).

Some of these philosophers (e.g. Kim) restrict talk of supervenience to relations between PROPERTIES. E.g. it might be said that

the property of being conscious supervenes on brain properties ABC
the property of wanting to drink supervenes on brain properties DEF
the property of seeing a snake supervenes on GHI.

It may be that this usage is common because the philosopher G.E.Moore first introduced "supervene" as a relation between ethical and non-ethical properties (Moore 1903). However, for our purposes, the restriction to properties is unacceptable, since we want to talk about how a whole *working* system, an *ontology*, in one sense of the word, can supervene on another. A working ontology is a portion of reality in which enduring things exist, processes and events occur, and there are causal relations, e.g. between the processes and events. This could be described as "*mechanism supervenience*", since it involves a relationship between two mechanisms, i.e. two complex, systems with causally interacting components. Two ontologies are involved because the two mechanisms include different kinds of objects, properties, relations, states, events and processes.

Obviously brains are mechanisms in this sense. Insofar as a mind includes percepts, desires, beliefs, abilities, attitudes, emotions, moods, skills, etc. and these can interact, for instance when learning produces new skills or new beliefs, or when a new percept causes a new desire to occur, etc., minds too are mechanisms, even though they are not physical mechanisms. They are mechanisms in which abstract entities interact.

Virtual machines in computers are also mechanisms in this sense. For instance, the addition of a new word to a list of words may cause some program to change the way it parses the list of words as a sentence, and analysis of a portion of program text by a compiler may cause the program's parse tree to be optimised prior to generation of machine code. More familiar virtual machine events include arithmetic operations, checking the spelling of a document, inserting a character in a line of text in a word processor, which can cause a the rest of the line, or even the rest of the page to be rearranged because of the resulting overflow.

We naturally think of these virtual machine events as causing not only other virtual machine events but also physical events such as altering the characters appearing on a screen, or changing the contents of a file on a disk. Later we'll discuss an objection to this claimed causal relationship. However, we first describe some alternative notions of supervenience which do not involve virtual machines or other abstract mechanism in which causation occurs.

4 Three uses of the word "ontology"

There are at least three different uses of the word "ontology" worth distinguishing:

1. The name of a field of study, i.e. what ontologists do! This is the oldest use, in philosophy.

2. A set of categories (a descriptive framework), e.g. the ontology used by an agent in perceiving, reasoning, etc. This usage is becoming increasingly important in AI and software engineering. E.g. for a robot to be able to communicate effectively with humans it had better share (some of) our ontology.

3. A collection of things that exist, interact, etc. E.g. we can talk about the ontology in a running computing system.

The main notion of ontology used here is the third one. However, I shall often be somewhat sloppy: the context should disambiguate between these uses.

There are related ambiguities in words like "ecology" and "geology" which sometimes refer to a field of study, and sometimes to a portion of the world that is subject to such study, as in the ecology of the South American equatorial forest, or the geology of Scotland.

A reference to the ontology of a unix system running at a certain time on a particular computer is analogous to a reference to the geology of Scotland: both refer to what exists in a part of the universe.

5 Other types of supervenience

Besides *mechanism supervenience*, where one ontology involving a collection of interacting mechanisms supervenes on another, there are at least three other kinds of supervenience with different features.

- PATTERN SUPERVENIENCE

  This occurs when a configuration of entities is capable of being grouped into various kinds of larger structures or "patterns" which exist because of the physical relationships between the entities. For example a collection of regularly arranged dots may "implement" patterns like vertical or horizontal or diagonal collections of equally spaced lines, spirals, hexagons, etc. Dynamic patterns can also supervene in this sense, e.g. moving lines, spirals, etc. created in a fixed array of lights going on and off in a carefully arranged sequence. Computer screens and TV screens provide many examples.

  Physical structures can support arbitrarily many different static or changing patterns, where the patterns depend on how the components are grouped. I.e. perception of such patterns is a result of a *parsing* process. In this sense a huge rectangular array of dots could include patterns corresponding to all of Shakespeare's sonnets expressed in a "dot matrix" font, though presence of the sonnets is not necessarily visible if dots forming characters are not demarcated by making them a different colour from the "background" dots.

Supervening patterns may have spatial and temporal relations, and mathematical properties, but there are not necessarily *causal* interactions between parts. When you view a football match on television, the patterns on the screen seen as a player's foot will move and so will the patterns seen as the ball, but the former motion does not cause the latter motion: both are physical events on the screen caused by physical events on the football pitch (and many intervening processes).

- PART-WHOLE SUPERVENIENCE
  This could also be called "agglomerative supervenience". It occurs when one or more large collections of entities has causal properties or structural relationships which are a result of the relationships between the component entities. For instance, the total mass and center of gravity of a large object are, in this sense, supervenient on the physical properties of parts. Here too the supervenience may involve dynamically changing properties and relations, e.g. the rotation of a wheel about an axle supervenes on the relative movements of their atoms of which they are composed. These supervenient properties and relations can be involved in causal interactions. An example would be one complex rigid object being completely enclosed by another, which prevents its escape.

  Pattern supervenience and *agglomerative* supervenience, could be grouped together because in both the concepts describing the supervening phenomena are *definable* in terms of the lower level concepts.

  Insofar as the larger scale concepts are definable in terms of the smaller scale ones, the supervening entities are part of the same ontology as those on which they supervene. So this is not the sort of mechanism supervenience we are discussing, which involves a relationship between distinct ontologies, though part-whole supervenience can also include "working" systems.

- MATHEMATICAL SUPERVENIENCE
  The previous two types involve static and changing physical configurations. They are both examples of a more abstract relationship, where one set of structures can be mathematically modelled in another.

  For instance Descartes showed how Euclidean geometry can be modelled in a vector space defined over reals, and vice versa. In the context of computing this sort of relationship is sometimes called "simulation". For example, a universal Turing machine can model a very wide class of virtual machines. If one of them can also model a Turing machine that is an example of *bisimulation*, a symmetric relation between mathematical structures. These mathematical relations hold between abstract entities, not actual working systems with causal interactions.

  However if a physical implementation of a Turing machine, e.g. some electronic mechanism with an extendable electronic memory, is used to implement a Lisp VM, and a lisp program is actually running, then that is an example of the kind of supervenience we are talking about, where a *working* VM is implemented in (and supervenient on) another VM, and both are implemented in and supervenient on a physical system.

  In this case, however, the concepts describing the virtual machines are not definable in terms of those of physics.

- "SHADOW" SUPERVENIENCE
  Like Plato, we know that if Fred's fist hits Frank's head, which moves as a result, then under some lighting conditions this will produce moving shadows on a wall or screen. In the shadow display the shadow of the fist hits (or should we say "hits"?) the shadow of the head, which moves thereafter. Clearly, the motion of the latter is not caused by the motion of the former. Shadows can be described in different ways, using different ontologies. We can talk about the shadow of the head, the shadow of

the fist, and their motion. Or we can talk about shadow blobs and their motion. Or we can talk about illuminated and shadowed portions of a surface. The motion of a moving head shadow or a moving fist shadow supervenes on changing patterns of light hitting the surface. However, although the shadow processes are produced by a mechanism, they do not form a mechanism. There may be some regular correlations between shadow events (e.g. different parts of the shadow move together in a range of circumstances) but the shadow events do not cause other shadow events: rather both are caused by events in a completely separate mechanism which does not require the shadows to exist and which does not necessarily produce the shadow events. For instance if the light source is moved or removed, or replaced by a diffuse light source, the shadows will disappear. Likewise when the fist and face and their owners are long gone, the shadows and their belligerent motions could reappear because someone is projecting a shadow cartoon film onto the floor or wall.

6 Ontologies include more than properties

An ontology (in the sense being discussed here) generally includes not only properties, but entities that endure over different time scales, properties of those entities, relationships between the entities, events, processes, causal connections involving those entities, ...

Working computing systems contain such ontologies.

- E.g. many virtual machines (VMs) running in computers include numbers, strings, lists, arrays, records, procedures, hash-tables, along with events and processes involving these.
- In a running word processor there are fonts, characters, words, lines, paragraphs, pages, headers, footers, diagrams, etc., and relationships between them. E.g. a page contains certain lines of text, each of which contains certain words, etc. An event could include insertion of a word, which causes a new line-break, leading to reformatting on that page and subsequent pages.
- A running compiler uses or produces syntax checks, symbol tables, parse-trees or fragments of parse trees, optimisers, code generators, error handlers, and perhaps a growing collection of sets of machine instructions derived from the source program.

While such a system is running, there is also a working physical computer, with its own ontology, which includes transistors, voltages, currents, atoms, molecules, etc., and causal connections between events in different parts of the computer.

What is the relationship between the two ontologies?

7 We want to understand how abstract ontologies relate to the underlying physical phenomena.

Virtual machine ontologies in computers are just a special case. There are many other cases were abstract virtual machines have persistent, interacting components, e.g. economic systems, social systems. For instance, social inequalities can cause jealousy and crime. Economic inflation can cause pensioner poverty.

In all these cases we are talking about more than the supervenience of *properties*.

Of course, someone may claim that the existence of objects, relationships, events or processes in such abstract virtual machines can all be “reduced” to the existence of properties of the system.

But *starting* from that assumption is inappropriate, since the vast majority of statements made by software engineers and users of computing systems refer to entities, relationships, etc., within the system, and not just to properties of the whole system.

8 Has all of this been solved by computer scientists?

Computer theorists analyse mathematical relations between abstract designs for machines. (E.g. showing how a particular sort of virtual machine could be modelled a Turing machine, or in an N-state automaton, where both are treated as mathematical abstractions).

That’s a start, but does not explain how those machines can run in physical systems.

Usually such mathematical analyses relate one virtual machine to another, but not to physical systems.[1]

Part of the relationship we are interested in is understood by computer engineers who build physical machines that implement digital electronic circuits, e.g. AND gates, NAND gates, electronic memory components, etc.

Such engineers do not usually think about the philosophical issues, though they may intuitively hold certain philosophical positions.

But we can try to learn from what they have built.

9 Virtual machine (VM) ontologies

We are talking about relations between *active* or *working* ontologies, involving actual events and processes, not just mathematical abstractions:

- When prolog is running in a computer, there is a prolog VM *doing* things. The ontology includes numbers, atoms, strings, lists, terms, assertions, rules, variables, etc. and processes like unification, rule-invocation, backtracking, database changes, etc.

  That VM supervenes on the ontology of the lower level VM (Sparc, Alpha, Pentium...), with registers, addresses, bits, bytes, words, mechanisms and processes that involve them, etc.

  And the latter VM supervenes on that of the digital circuitry, and that supervenes on a physical ontology...

- When a chess program runs, the chess VM includes pawns, kings, rows, columns, moves, captures, pinned pieces, etc....

  That might supervene on a Prolog virtual machine, or a Pop-11 virtual machine, or a Java virtual machine...

We need to understand how one (running) ontology can provide the substratum for another. We can then ask whether and how a software VM running in a computer is like or unlike a mind running in brain.

[1] One of the exceptions is (Scheutz 1999)

This is a sort of dualism (Ryle's "ghost in the machine"), but dualism with a twist.

Do we need a ghost inside the machine?

No, it's the other way round!

An intelligent ghost must contain an information processing machine

Despite claims of behaviourists, etc., intelligence requires internal processes involving information manipulation (more precisely manipulation of objects, states or structures that convey information), for instance in:

- perceiving (including analysing, parsing, interpreting, combining, sensory data and more abstract percepts),
- learning
- wanting
- preferring
- evaluating
- reacting (mentally or physically)
- deciding
- wondering
- having emotions
- ....

These are mental events and processes in an information processing machine, i.e. a machine which acquires, transforms, interprets, infers, stores, combines, and uses INFORMATION.

Information processing machines, doing at least some of these things, have existed for millions of years (i.e. in organisms).

Humans have made simple artificial versions for hundreds of years, e.g. calculators, sorters, mechanical looms, clocks, sundials, etc.

But it is only very recently that we have begun to study and build really flexible and powerful information processing machines.

WE STILL UNDERSTAND ONLY A SMALL SUBSET, HOWEVER.

10 Most information processing involves events in a "virtual machine"

The objects involved in information processing, their properties, their relationships, the processes in which they occur, their interactions are *not physical. They may be:*

- syntactic: e.g. information items can have a syntactic structure
- logical: e.g. this information is inconsistent with that
- semantic: e.g. this refers to that – for instance a variable in virtual machine in a computer may refer to a number, a list, a procedure, etc.

Concepts describing *mental* phenomena, e.g. "infer", "interpret", "contradict", "refer", "decide", "learn", "concept", "proof", "plan", "belief", "preference" etc. are not concepts of *physics*.

We typically cannot use concepts from physics to describe the entities or their relationships, apart from temporal relationships, perhaps. They cannot be *defined* in terms of concepts of physics.
(If they can be, show me how.)

This is true also of concepts used to describe processes in much simpler virtual machines, e.g. a Prolog VM, a word processor, or a chess machine. The chess concept of "capture", and the prolog concept of "unification" are not definable using concepts of physics.

Of course, when the processes occur in the virtual machine they depend on (different) processes happening in lower level virtual or physical machines. They are *implemented* in physical mechanisms, though not always the same ones.

NB "Virtual" does not mean "unreal", or "imaginary" or "lacking in causal powers".

Virtual machines in computers are as real as poverty, economic inflation, cultural change, and other abstract processes that impact on our lives.

All of these have causal powers, and are therefore not "epiphenomena",

though there are problems about the causal powers, discussed later.

What we are talking about is just one facet of a much larger picture

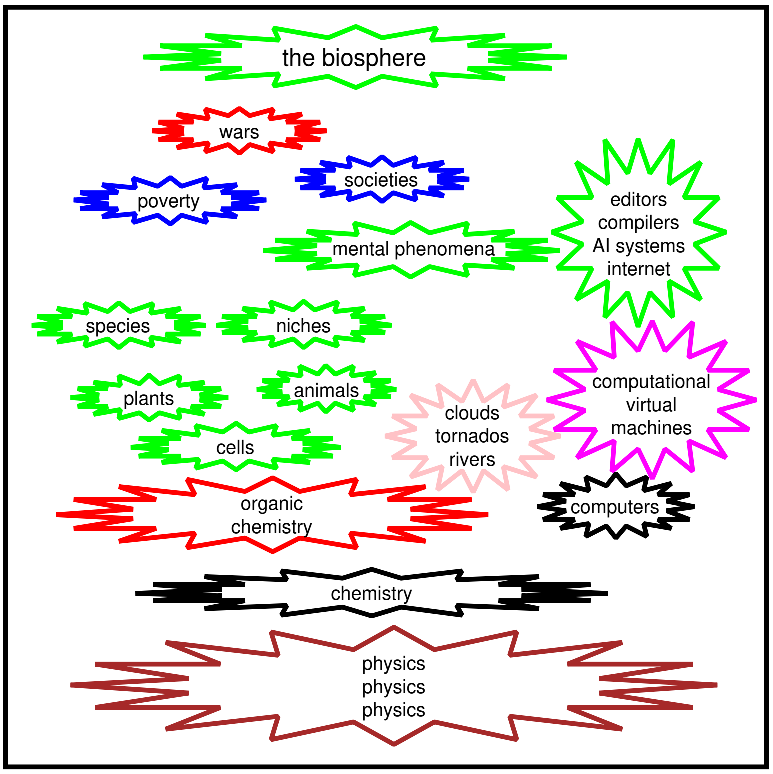
A simplified sketch follows ....

11 Features of supervenience: A relation between layers in reality

Some features of supervenience, where supervenience is a relationship between a running virtual machine and some lower level machine.

- Supervenience is asymmetric.
  E.g. there are many physical phenomena that can exist without any mental events, and without a running Prolog VM. But no mental phenomena, or Prolog VM events can occur without some physical substratum.
- Some physical phenomena are *sufficient* for the existence of the VM phenomena.
  When certain physical systems exist and work normally, then that produces mental states, events, processes, etc. When certain other physical processes occur, they are sufficient for the existence of a

Reality is multi-layered



This does not imply that it is all stratified: the layers define at best a *partial* ordering, not a *total* ordering. Some implementation relations may be circular.

running Prolog VM, with certain events occurring.

- "Multiple realizability": Different physical phenomena can suffice for the existence of the same mental phenomena.
  E.g. the very same type of word processor or chess program can run on different hardware architectures, or on the same digital hardware architecture implemented in different physical materials, or can run on the very same physical machine on different occasions but using different parts of the physical memory or discs. Even during a single execution of a prolog program, the location of the physical implementation of a particular list may change because of paging, swapping or garbage collection.

12 The multiple realizability of mental phenomena

There are different ways in which phenomena in information processing virtual machines, e.g. mental phenomena, may be multiply realizable.

For instance, suppose two people both have the thought that physics is a deep science.

If one is a Chinaman who knows no English and one is a Scotsman who knows no Chinese, then it is very likely that how they think will be related to the languages they know and having a thought which we describe in terms of its content will map onto different physiological structures and processes if they know different languages.

Even two speakers of the same language will probably have different physiological processes when they have the same thought, e.g. because the processes of learning language involve many self-organising physiological processes, and very different histories may produce functionally equivalent but physically different physical infrastructures.

Even if the same person has the same thought in different contexts, the differences in recent history and differences in the context of the thought could cause the thought to occur with many different physical details regarding which neurons are in which physiological state. So multiple realization is likely to be a general feature of mental processes in animals.

13 Non-features of supervenience

Philosophers (and brain scientists) sometimes propose, or presuppose, conditions for supervenience that are violated by examples of VMs in computers.

Here are some proposed conditions, which must be rejected as *necessary* conditions (though they sometimes hold):

- Components of a supervenient system must correspond to fixed physical components which realize them: NO.
  Counter-examples were mentioned above. Even a weaker formulation which requires certain *types* of VM objects or events to be always implemented in the same *types* of physical objects or events, is refuted by the recent history of computing, since an old VM-type can be implemented in new physical materials. It may also be false for the relationship between mental and neural phenomena, because of differences between individuals, and changes within individuals.
- The structural decomposition of a VM (i.e. the part-whole relations) must map onto isomorphic physical structures: NO.
  This is refuted by such facts as that list A can be an element in list B while B is an element in list A. It is impossible for two physical objects to contain each other as parts, but possible in a VM.
- If a VM, M, is implemented in a physical machine, P, then P must have at least as many physical components as M: NO.
  This is refuted by sparse arrays in computational virtual machines. A sparse array can have more locations than there are electrons in the universe, let alone components in the machine implementing it.

14 Some minimal requirements

(1) A minimal requirement for a working instance of a virtual machine to exist is that there be some physical mechanism that implements it.

I.e. virtual machines depend on physical systems. It doesn't follow that they are "nothing but" physical systems. (The "nothing buttery" fallacy.) For their ontologies are different, as we have seen.

When a theoretical computer scientist investigates the properties of a VM, there is no presumption that any actual running version exists. and so no physical implementation is required for the existence of virtual machines as mathematical objects in such theoretical studies. But when there is a *running* virtual machine, there must be a physical system in which it is implemented.

(2) A VM difference requires some physical difference.

(This requirement is close to what G.E.Moore wrote about ethical properties supervening on natural properties)

Requirement (2) has two facets:

- If a VM is running and changes in some way, that implies that there has been a physical change.
- If there are two virtual machines M1 and M2, and one has a certain feature which the other lacks then there must also be a physical difference underlying the difference at the virtual machines.

Difference in physical machines does not imply difference in VMs, but difference in VMs implies physical differences.

VM events may depend on, be implemented in, "external", even remote, physical events.

A physical change, or difference, that accounts for a VM change, or a difference between VMs M1 and M2, need not be located *within* what is naturally referred to as the physical machine containing M1 or M2.

E.g. a change in a VM may involve a semantic relation to something external. I may cease to know where Fred is because he moves from Canada to Japan without telling me.

Similarly, I can cease to be the tallest person in the room if someone else comes in; and whether I own a certain house can change if documents in a registry office change.

So it is wrong to say that ALL mental states of a person are fully implemented within the brain, or even within the body of that person.

My ability to think about a *particular individual* such as the Eiffel tower depends in part on the existence of the individual. Of course I can think about something of that *type* which does not exist.

So even when a VM is associated with a bounded physical machine, the actual implementation, what the VM supervenes on, need not be *local* to the physical machine.

SUPERVENIENCE NEED NOT BE A "LOCAL" RELATION.

Trying to study only the relation between mind and brain, ignoring the physical (and social) environment, is a serious mistake.

Is there some kind of identity between between minds and brains, or between computational virtual machines and the computers that implement them?

Some have argued that despite all the differences between virtual machines and their physical implementations, and despite the non-reducibility of virtual to physical machines through definition and deduction, there may be some form of identity or strong reducibility, that goes beyond causal dependence.

One problem with identity theories is that if M is identical with P, then if M supervenes on P, then P also supervenes on M.

I.e. identity is a *symmetric* relation whereas supervenience between ontologies generally is not symmetric.

This issue needs further discussion, another time. A problem is that notions of identity are usually inherently ambiguous and indeterminate, as the ancient Greeks discovered.

Compare the identity of rivers, over space, over time.

There are many interesting questions about multi-layered reality.

- What sorts of relationships are there between levels?
- Can events and processes at higher levels have causal powers?
- Can causal influences go up and down between levels (circular causation)?
- What are the temporal relationships between events at different levels?
- Many of the higher level phenomena admit of multiple-realizations at lower levels (e.g. multiple implementations of a Sparc virtual machine, multiple implementations of beliefs, desires, percepts, in different organisms, different people, the same person at different times): can we characterise the relationships if they are so variable?
- Is there a bottom level? If so what is it like?
- Will physicists discover a new, more fundamental, type of physics, one day?
- Given a physical machine, can we (in theory, or in practice) determine whether or not a particular virtual machine M is running on it or not? (Compare decompiling a machine code trace.)
- How can we check out which of our theories about the phenomena at various levels are correct? (The history of science reveals many of the difficulties....)

15 Two kinds of research

Two kinds of research are needed: scientific and philosophical.

(a) Scientific research explores the contents of the various layers and their relationships

This includes:

- Developing ontologies to describe the layers. I.e. exploring the "form" of the world, what sorts of things *can* exist.
  This can take thousands of years, with many mistakes on the way!
  E.g. space can be curved, neutrinos can exist, genes might be involved in biological reproduction, economic inflation can occur, a perception of relative deprivation can occur in some communities, an operating system can 'thrash'.
- Discovering the "contents" of the world
  I.e. finding out which things actually exist, where they are, what sorts of processes occur, ...
- Discovering limitations in possible co-occurrences: the "laws" of the world, including causal connections.
  This presupposes the form, i.e. what can exist, and then finds limits. You can't discover that pressure, volume and temperature of a gas are related by $P * V = kT$ unless you presuppose that such things as pressures, volumes and temperatures can exist.

NOTE: Only the last kind of theory is empirically falsifiable (sometimes). The first two may be confirmed, but not refuted.

See also: Chapter 2 of *The Computer Revolution in Philosophy* 1978.
Out of print but photocopies available in School Library.
See: http://www.cs.bham.ac.uk/~axs/

(b) Philosophical research includes attempting to clarify concepts, and to analyse paradoxes that arise out of confusions in our concepts. However, it needs to be informed by, and can contribute to, scientific research.

A paradox:

- We assume that physics is causally closed backwards
  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, by the physical features of the circuit, the previous states, the most recent physical inputs from the environment.
- We assume that events in virtual machines can cause other events in the virtual machines, and can also produce physical effects
  E.g. inserting a word in a paragraph in a document can cause reformatting to occur, which changes what glows on the screen.
  Detection of a syntactic error in a program can cause a compiler to print messages, etc. etc.
  Ignorance can cause poverty, poverty can cause crime, and crime can involve movement of cars, TV sets, bullets, etc.
  Having a desire can cause you to take a decision, and to walk out of the room...
- So events in virtual machines can cause physical events.

So physics is not causally closed after all??

Or perhaps our desires do not cause our actions??

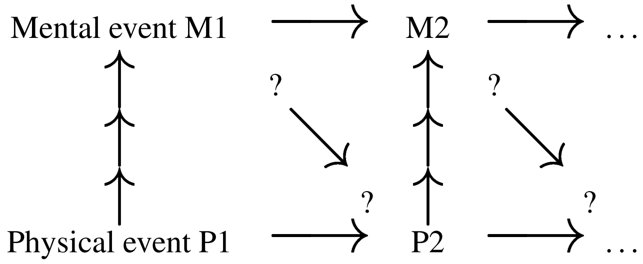
16 Problem

In philosophical moods, many people tend to think of causation as inherently physical.

(But not in "everyday" thinking.)

This causes problems for the "multi-layer" view of reality (pluralism). Some claim that only the physical objects have "real" existence and a "real" capability to engage in causal interactions.

They reject pictures like this, where the arrows imply causal influence (and time goes left to right):



If P2 is fully explained by P1 (physical causal closure), then it looks as if there cannot be any scope left for M1 to cause P2, or even some aspects of P2.

Moreover if the second physical event, P2 completely accounts for M2, and P1 completely accounts for P2, then there is nothing M1 can do to produce or modify M2. So M1 can have neither physical nor mental effects. It is purely ‘epiphenomenal’!

So, for many people, there is a conflict between their “everyday ontology” (*desires and poverty can have effects*) and their “philosophical ontology” (*an event cannot both be fully explained physically and also have mental causes*).

CAN THIS BE RESOLVED?

17 ASSERTION: Not only physical things have causal powers

Problems with the ‘monistic’, ‘reductionist’, physicalist view that non-physical events are epiphenomenal:

1. It presupposes a layered view of reality with a well-defined bottom level. IS THERE ANY BOTTOM LEVEL?
2. There are deep unsolved problems about which level is supposed to be the real physical level, or whether several are.
3. It renders inaccurate or misleading much of our ordinary and scientific discourse, e.g.

- *Was it the government’s policies that caused the depression or would it have happened no matter which party was in power?*
- *Moving white’s knight caused black’s knight to be pinned by white’s bishop.*
- *Your anger made me frightened.*
- *Changes in a biological niche can cause changes in the spread of genes in a species.*

Of course, it is possible that our ordinary discourse is totally confused, but, if so, that would undermine our legal system, much of our social and political thinking, and a great deal of our ordinary thought and language about ourselves and others.

Could software engineers give up thinking that virtual machine events can have effects (e.g. bugs in software cause miscalculations or wrong decisions, which cause planes to crash)?

DIAGNOSIS: WE ARE CONFUSED BECAUSE OUR NOTIONS OF CAUSATION HAVE HIDDEN COMPLEXITY.

18 Conjecture: Towards a schema for causation

In the “everyday” ontology, used in our practical interactions with one another and the rest of the world, we use a notion of “causation” that is POLYMORPHIC.

“X caused Y” does not have a fixed, context-independent meaning. Rather it expresses a general schema, which has to be filled out differently in different contexts, according requirements of those contexts.

E.g. we can correctly say of a particular person that his death was caused by smoking, that his death was caused by lung cancer, or that his death was caused by certain physiological processes that occurred in the last few minutes of his life. The assertions do not contradict one another. (Why not?)

Likewise, we can say that a car crash was caused by poor driving or by ice on the road. These statements, though both true, are relevant to different contexts of enquiry. E.g asking why the driver did not crash when he drove on this road previously could be answered by saying the crash was caused by ice. Asking why other drivers did not crash on that road could be answered by saying that this person was a poor driver.

REDUNDANT CAUSATION IS THE NORM.

Each question about causation is linked to a range of possible circumstances (same driver, different occasions, different drivers same physical conditions, etc.).

THERE IS NO UNIQUE, GLOBAL, CONTEXT DETERMINING WHICH STATEMENTS ABOUT CAUSAL CONNECTIONS ARE TRUE.

19 Causation is a "high order" relationship

There is no uniquely correct, context-independent, answer to the question: "Did X cause Y?"

Ordinary thinking and communication about causation is based on presuppositions about the LAWLIKE RELATIONSHIPS and the truth of rather subtle counterfactual conditionals linking possible situations and events.

There is (usually) some implicit or explicit context which determines which factors are relevant to answering the question. So "X caused Y" is not just a statement about events X and Y. *There is implicit reference to some context.*

When the context is unspecified, disputes about causation can be at cross-purposes, lacking any correct answer.

Context is important because the question is not a purely factual one, but is relevant to practical decision making.

Compare another case of implicit existential quantification: "Which of machines A and B is best for mowing the lawn?"

The answer depends on (at least):

a) circumstances in which the lawn is to be mown (e.g. height of grass, who is doing the mowing, size of lawn),
b) how various aspects of performance are valued by the questioner (e.g. cost, ease of use, reliability, ease of maintenance, etc.)

20 Conjecture: What "X caused Y" means.

"X caused Y" says something quite complex, with many unobvious and subtle features, with at least the following three implications:

1. X happened and Y happened.
2. In a certain variety of possible circumstances, C1, if X had happened then Y would also have happened.

3. In a certain variety of possible circumstances, C2, if X had not happened and nothing else had occurred capable of producing Y, then Y would not have happened.

Which sets of circumstances C1 and C2 are relevant will depend, in subtle and complex ways, on the practical context in which the question about causation is asked.

E.g. attempting to assign blame leads to different questions from attempting to decide how to behave in future.

Of course, this analysis of causation will be unsatisfactory if we cannot find a good analysis of conditionals, including counterfactual conditionals.

However that is a problem that needs to be solved independently of this. My claim is that the notion of "what would happen if" is more general than the notion of causation, and is presupposed by our concept of causation.

21 Defeasibility of statements about causation

Whenever anyone tries to specify precisely the range of possible circumstances under consideration, it is always possible to produce a refinement of the specification which makes the consequent of the conditional false.

So for instance, you may have good reason to think that in circumstance C1, if X had happened then Y would have happened.

E.g. if Fred had drunk less he would have avoided the crash.

But you may not have considered what could occur if Fred had a heart attack, or if aliens from another planet with very advanced technology had turned on some powerful remote-acting machine which interfered with Fred's driving.

A disputant may or may not be able to persuade you that a previously unnoticed possible situation is relevant: depending on your high level practical goals. E.g. trying to prevent disasters in the next 20 years is not the same as trying to prevent disasters in the next 2000 years.

Statements about causation, like statements about counterfactual conditionals, are inherently (partly) indeterminate in meaning.

In general it is impossible to produce a non-trivial, non-circular, context-independent, specification of the relevant variety of circumstances.

Specifying the circumstances as those in which X suffices for Y is, of course, circular if you are arguing about whether X caused Y.

(Compare: C.N.Taylor, DPhil Thesis, Sussex University, 1992.)

22 Multiple realizability and causation

If we are considering whether X caused Y, and X is an event in a virtual machine, the difficulty in specifying the relevant variety of circumstances to fill in schemata 2 and 3 is compounded by multiple realizability of virtual machine states and processes.

We may not know enough to specify the variety of physical circumstances in which X will occur, let alone those in which the occurrence of X will produce Y.

For instance, if X is a chimp's decision to select one berry rather than another, there is a wide variety of circumstances in which that decision would be followed by the action of picking up the berry, because we believe organisms have many interacting mechanisms (including perceptual and motor control mechanisms) produced by evolution specifically to *ensure* that decisions are carried out, if necessary by counteracting or compensating for many possible perturbations during the process.

But (apart from relatively simple homeostatic mechanisms) we usually don't know precisely what the mechanisms are, or what the variety of circumstances in which they suffice for their biological function, nor how various kinds of growth, learning, or damage-repair will modify the underlying physical implementation, nor how the implementation can vary from one member of the species to another.

23 Similar problems arise for virtual machines in computers

Likewise we can say that an event in a virtual machine in a computer (e.g. an attempt to access a file) will cause some other event (e.g. checking the access rights of the program).

But we may not be able to predict precisely all the future technologies that could produce a physical implementation of such processes, nor the variety of types of intrusions that could interfere with normal functioning of the mechanisms.

Moreover, if the computing system is the result of design and implementation work done by different people (or companies) solving different sub-tasks, and if the system has done some self-optimisation or self-modification (e.g. self-tuning schedulers or file managers), then our ignorance is comparable to our ignorance about biological designs.

So in both cases we don't know precisely which range of circumstances we are quantifying over.

However, despite all that, I can be confident that if my program for sorting numbers runs on some machine of the future, or if the machine on which it is current running is upgraded with a CPU modification while running, then, after the change, if the program is given the list [3 99 1 5 6] it will return [1 3 5 6 99].

I have that confidence because of my general trust in the processes of production of computers, operating systems, compilers, etc. But occasionally the confidence is misplaced!

WE DON'T KNOW ALL CONTEXTS IN WHICH THINGS CAN GO WRONG

24 Causation has some counterintuitive properties

A corollary of the above is that "causes" is not in general a transitive relation.

That is because different sets of circumstances can be referred to when we say that A causes B and B causes C.

Suppose X has a fall producing a fractured bone.

Then it may be natural to say:

(1) X’s fracture causes him pain

(2) X’s being in pain causes Y to feel unhappy

but misleading to say that

(3) X’s fracture causes Y to feel unhappy,

because if there had been pain without the fracture Y would have been unhappy in the same way.

Whether such a transitive inference from X caused Y and Y caused Z to X caused Z is valid may depend on the sort of contexts in which the first two relations are considered. If the same sets of conditions are relevant to both, then the third relation holds.

Another corollary is that multiple causes of the same event are possible.

That has already been illustrated with the smoking and car crashing examples. We could say that the ice on the road and the poor driving caused different aspects of the crashing event.

Over-determination often involves multiple aspects.

Similar remarks apply to physical events (e.g. walking) which are caused both by mental events (e.g. deciding to leave the room) and physical events (e.g. previous states of the person’s brain and the perceivable environment).

25 Resolving the paradox

How both P1 and M1 can cause P2:

This framework allows at least an outline resolution. When we say that M1 causes P2 (or some aspect of P2), this is not refuted by saying that P1 causes P2, because even if P1 did cause P2, it may still be true that:

1. M1 happened and P2 happened
2. In a certain variety of possible circumstances, C1, if M1 had happened then P2 would also have happened.
3. In a certain variety of possible circumstances, C2, if M1 had not happened and nothing else had occurred capable of producing P2, then P2 would not have happened.

If 2 and 3 are correct they will be correct because the variety of ways in which M1 can exist or not exist is constrained.

This will limit the variety of physical conditions under which M1 exists, and the variety of conditions in which M1 does not exist. I.e. the variety of ways in which M1 could be kept true, or made false, in changing circumstances is limited to those which also keep P2 happening, or prevent it happening. Thus we get no contradiction between the above and these:

4. In a certain variety of possible circumstances, C3, if P1 had happened then P2 would also have happened.

5. In a certain variety of possible circumstances, C4, if P1 had not happened and nothing else had occurred capable of producing P2, then P2 would not have happened.

I.e. both M1 and P1 can be causes of P2. The assumption that physics is causally closed backwards does not follow from the assumption that physical events suffice for the causation of physical events.

One way of dealing with this is to remove the puzzle by saying that M1 and P1 are the same thing: i.e. adopting the mind-brain identity theory. This can cause problems if you want to be able talk about identity of virtual machines across possible worlds (e.g. "What would the operating system have done about allocating memory if process P25 had terminated just before P32 requested additional memory, instead of after?)

However it is not clear that discussions regarding identity are discussions of substance: what is treated as identical with what may be partly a matter of convenience, or conceptual clarity, rather than truth. This is a topic for another occasion.

Exactly similar cases occur in control systems.

For example, a chemical plant may be controlled by a computing system.

Then a decision taken by the software system, i.e. an event in the virtual machine M1, may cause some later physical event P2, such as a valve being opened. An earlier physical event P1, involved in the implementation of M1, can also be seen as a cause of P2. There is no contradiction here, given the normal interpretation of 'cause'.

This sort of multiple causation is commonplace in the engineering world.

Very often the only relation that is of interest to the engineers is the relation between the VM events and the physical events, e.g. because the VM process involves a software bug which has to be removed, or because the VM can be generalised to deal with more situations.

The precise physical details when the VM is running with the bug may vary and those when it runs after the bug has been fixed may vary.

The software engineers typically neither know nor care about them.

However, they would care if a physical fault, e.g. a memory fault, cause the event P2 not to occur, or to occur in an undesirable modified form.

Likewise, *we neither know nor care about events in our brain, when our deliberations or desires produce appropriate or inappropriate actions.*

But we do care about brain events when there's damage or disease.

This commonplace view of "biological mental causation" (as it occurs in humans and animals) seems to parallel the case of "artificial mental causation" (i.e. causation in software virtual machines produced by engineers).

At present the former are simpler and easier to understand than the latter.

So if we analyse carefully the products of engineers and scientists building working models and systems that control complex machinery, we may be able to develop a conceptual framework that enables us to ask, and perhaps answer, refined and clarified versions of old questions.

It is also necessary to get clearer about counterfactual conditionals, and explain why the "politician's semantics" for counterfactuals is incorrect. (I.e. when someone says "What would you do if XYZ happens?" the politician answers, inappropriately, "XYZ won't happen".)

26 Our CogAff Virtual Machine Architecture Schema

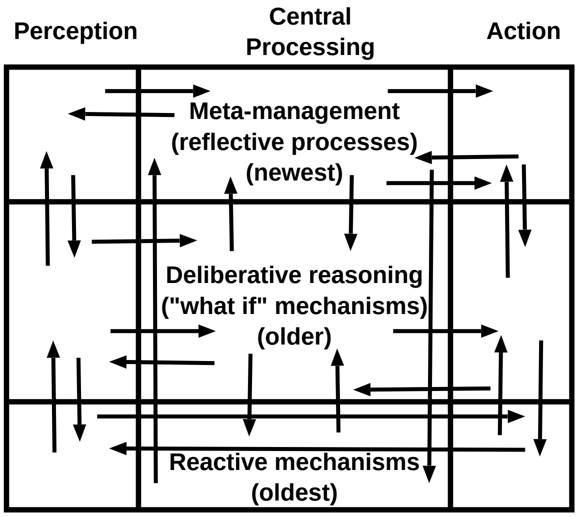


Figure 1: The CogAff Architecture Schema

Our Cognition and Affect project has been developing an architectural schema depicted in Fig 1 and described more fully in the CogAff project directory. Instances of this schema will have large numbers of concurrently active, causally interacting components which can change over time and which co-evolved (a sort of mental ecosystem).

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

To a first approximation it distinguishes 9 classes of architectural components in a 3 by 3 grid.

The *horizontal* divisions ("layers") reflect differences in evolutionary age, level of abstraction of processing, and differences in function. The *vertical* divisions ("pillars") correspond to different relations with the environment. Compare chapter 25 of (Nilsson 1998)

These VM divisions need not all correspond closely with physiological divisions.

Not all functioning agent architectures will have all the components. E.g. it is likely that insects have only reactive components. Many of the older AI systems had only deliberative components, and for some purposes they suffice, though when applied to complex problem-solving or planning tasks the lack of the self-monitoring capabilities provided by the meta-management components can lead to very poor performance. (Many AI researchers, instead of diagnosing the main cause, an inadequate architecture, assumed that the solution was to switch to a different class of representations and algorithms, e.g. neural net or evolutionary mechanism. This sometimes led to improved performance, but merely shifted the problem to a different region in the problem space, where the efficiency gains were not enough.)

We conjecture that evolution discovered the need for all three layers and provided them in their most sophisticated form in humans, though it may be that the chimpanzee architecture is not very different.

27 The H-Cogaff architecture

The H-CogAff architecture is a conjectured architecture that includes all the types of components permitted in H-Cogaff, including a special class of reactive mechanisms which can be seen as performing the function of a sort of "Alarm" system as depicted in Fig. 2. Further information about CogAff and H-Cogaff can be found in papers here:

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

though our terminology changed recently and we did not clearly separate out the CogAff *schema* and the H-Cogaff *special case* of the schema.

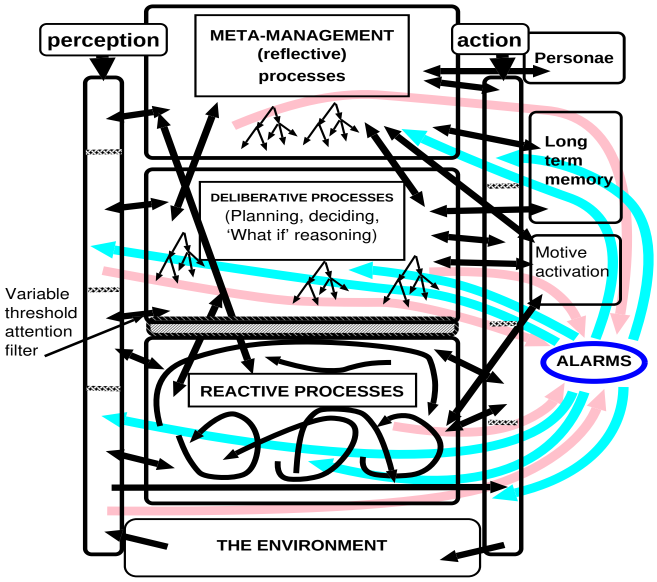


Figure 2: The H-Cogaff architecture instance of CogAff

We argue that in some ways this is more like an "ecosystem" than a society of mind (Minsky), because of the ways the components have co-evolved.

Because such a virtual machine architecture includes very large numbers of concurrently active, constantly interacting, sometimes competing and sometimes collaborating components, with many sorts of short term and long term feedback loops, the collection of counter-factual conditional statements that are true of such a system will be very complex, and possibly very hard to discover, because they can depend on very subtle aspects of the internal states of short term and long term memories. The interactions will certainly be more complex than those involved in the long term meteorological effects of a butterfly's movements.

We still know very little about the varieties of virtual machine that are possible, how they might work, how they can be implemented, how they might interact, etc. etc.

There's lots more for philosophers and scientists (including software engineers and brain scientists) to do.

That includes analysing in much more detail the types of ontologies and relationships, especially ontologies produced by virtual machines that inspect themselves and construct new ways of thinking about themselves.

Filling in the details still requires a lot of work and until this has been done it will be very difficult to be clear about the requirements for a physical system that can implement such an architecture. Until then we cannot really know in detail what sorts of causal relationships can hold between the virtual machine events and the underlying physical machine events, and what sorts of counterfactual conditionals are made true by the causal interactions within the architecture.

Future papers on these topics will be posted at the Birmingham Cognition and Affect web site:

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

References

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

D. C. Dennett. *Consciousness Explained*. Penguin Press, London and New York, 1991.

J. Kim. *Supervenience and Mind: Selected philosophical essays*. Cambridge University Press, Cambridge, 1993.

J. Kim. *Mind in a Physical World*. MIT Press, Cambridge, Mass, 1998.

G.E. Moore. *Principia Ethica*. Cambridge University Press, 1903.

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

M. Scheutz. When physical systems realize functions.... *Minds and Machines*, 9:161–196, 1999. 2.