The history, nature, and significance of virtual machinery

Aaron Sloman

School of Computer Science, University of Birmingham

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

Related slides are available in my 'talks' directory:
`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/`

and on Slideshare.net
`http://www.slideshare.net/asloman/presentations`

Robin Milner[*]

About 25 years ago, I remember Robin Milner giving an invited talk at a conference (an Alvey conference, I think) about how theory repeatedly follows practice in computing.

His main thesis was something like this:

> Engineers solve practical problems and in the process invent new things and do new things that they don't necessarily characterise explicitly.
>
> Theorists notice what's happening and invent formalisms and models to account for it, and mathematical techniques for reasoning about it.
>
> Meanwhile the engineers move on, and later the theorists follow, and so on.
>
> and the cycle continues.

I think my topic illustrates that cycle, though this particular development seems to me to be still in its early stages.

[*]Robin Milner died on Saturday 20th March 2010, and will be much missed.

ABSTRACT (Part 1)

Abstract:

Over the last six or seven decades there have been a lot of separate developments adding functionality of different sorts to computing systems including (in no significant order):

memory management, paging, cacheing, interfaces of many kinds, interfacing protocols, device drivers, adaptive schedulers, privilege mechanisms, resource control mechanisms, file-management systems, interpreters, compilers and run-time systems for a wide variety of types of programming language, garbage collectors, varied types of data-structure and operations on them, tracing and debugging tools, pipes, sockets, shared memory systems, firewalls, virus checkers, security systems, network protocols, operating systems, application development systems, etc. etc.

All this is very familiar to computer scientists and software engineers, though different experts know about different sub-sets of these developments and the whole package is not often described adequately.

In a way it is very familiar to millions of users, who are incapable of describing what they are using.

A consequence of all these developments is that we can now have, in addition to all the physical computing machinery that we use, varying collections of non-physical machinery made up of various kinds of interacting components with causal powers that operate in parallel with the causal powers of the underlying machines, and can help to control those physical machines, but with different kinds of granularity and different kinds of functionality from the physical machines.

These are running virtual machines (RVMs), as opposed to the mathematical abstractions that are sometimes called virtual machines (e.g. a universal turing machine, the java virtual machine, the/a linux virtual machine) whose instances are among the RVMs.

ABSTRACT (Part 2)

I shall try to characterise some of the forms of control and self-control that are made possible by the use of RVMs and suggest that biological evolution discovered the need for them long before we did and probably produced more complex and varied kinds than we have so far.

All this seems to me to have implications for the future of machine intelligence (and more generally for development of increasingly robust, autonomous systems).

It also has implications for the future of philosophy of mind, and looks likely to achieve the final removal of what T.H.Huxley called “the explanatory gap” between the physical and the mental, which was, and still is, a serious problem for Darwin’s theory of evolution.

To illustrate what I am talking about I’ll start with a simple demo.

This is a request for help on my part:

I would like corrections if I make any incorrect historical or other factual claims,

and suggestions for improved ways of summarising and explaining the significance of all these developments,

especially for the benefit of people who do not work in computer science or software engineering.

KEY IDEA: Running Virtual Machine (RVM)

The idea of a running virtual machine (RVM) should not be confused with the abstract mathematical structure defining a type of VM, which can be thought of as a “Mathematical Model” (MM), about which theorems can be proved, etc., but which does not **do** anything, anymore than numbers do.

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

Distinguish: **PMs** **MMs** **RVMs**

Illustrate with demos.

E.g. Sheepdog demo. See Movie 5 here:
`http://www.cs.bham.ac.uk/research/projects/poplog/figs/simagent`

The “Explanatory Gap”

A problem that puzzled Darwin and fired up his critics:

- There’s lots of evidence for **evolution of physical forms**.
- There’s no evidence that human minds could be products of evolution.
- There seems to be no way that physical matter can produce mental processes.

This is the so-called “explanatory gap”

(T.H. Huxley – and various precursors).

- Until the last few decades, explanatory mechanisms linking physical and mental phenomena were not even conceivable to most scientists: hence Huxley’s “explanatory gap” and Chalmer’s “Hard problem of consciousness”, etc.
- Now, as a result of a great deal of work on hardware, software, firmware, and CS theory, we know how to make things that have some of the features required for working explanatory models of mental processes, with some of the key features of mental processes (including having causal powers, without being physical processes) – but only in very simplified form.
- Most philosophers, psychologists and neuroscientists have ignored or misunderstood this, and so have many AI/Computing/Software researchers.

  I’ll give some pointers, but not explain in detail, below.
- There are deep implications for philosophical analyses of causation.

  E.g. **downward** causation from mind-like events and processes to physical events and processes.

Let's vote

Does **your** ontology include virtual machines?

Who agrees with the following?

- Ignorance can cause poverty?
- Over-zealous selling of mortgages can cause hardship for people in many countries?
- Voting can influence decisions?

If you AGREE with any of these, i.e. if you think such an effect CAN occur, then it appears that you (wittingly or unwittingly) have social and/or socio-economic virtual machines in your ontology.

What that means is another (hard) question, partially answered below.

In order to explain what a non-physical virtual machine is we need to:

- Explain what a machine is
- Define "physical machine" in terms of the sorts of concepts that suffice to describe the structure and operation of the machines.
- Provisionally define "virtual machine" as a machine that is not (fully) physically describable.
  (The phrase "virtual machine" is unfortunate, but it's too wide-spread to change.)
- Later on generalise the notion "virtual" by defining it as relative.

What is a machine (natural or artificial)? (1)

The word “machine” often refers to a complex enduring entity with parts

(possibly a changing set of parts)

that **interact causally**[*] with other parts, and other “external” things, as they change their properties and relationships.

[*]Causation is discussed later.

The internal and external interactions may be

- **discrete** or **continuous,**
- **concurrent** (most machines), or **sequential** (e.g. row of dominoes, a fuse(?))
- if concurrent then **synchronised** or **asynchronous**

In Turing machines, everything is:

- Internal
- Discrete
- Sequential
- Synchronous

Concurrent and synchronized TMs are equivalent to sequential TMs.

I.e. parallelism in TMs adds nothing new.

But some machines have concurrent parts that are not synchronised, so they are not TMs, even if they have TM-like components.

And systems interacting with a physical or social environment are not TMs, since a TM, by definition, is a self-contained: machine table+tape.

What is a machine (natural or artificial)? (2)

The word “machine” often refers to a complex enduring entity with parts

(possibly a changing set of parts)

that **interact causally** with other parts, and other “external” things, as they change their properties and relationships.

The internal and external interactions may be

- **discrete** or **continuous,**
- **concurrent** (most machines), or **sequential** (e.g. row of dominoes, a fuse(?))
- if concurrent then **synchronised** or **asynchronous**

NOTEs

1. Machines, in this general sense, do not have to be artificial, or man-made, or deliberately designed to do what they do.

2. The perception of machines and how they work is one of the important functions of human visual perception, and haptic/tactile perception, (possibly also in some other species).

That includes the perception of structures, processes and causal relationships (proto-affordances).

This is generally ignored by vision researchers.

Perception of affordances is a special case of this. E.g. See

Architectural and Representational Requirements for Seeing Processes, Proto-affordances and Affordances,

`http://drops.dagstuhl.de/opus/volltexte/2008/1656`

Typical features of machines (natural and artificial):

Machines

- can have various degrees and kinds of complexity
  (often hierarchical – machines composed of machines)
- allow changes/processes to occur within them
  usually concurrent changes (e.g. gear wheels turning, ends of lever moving in opposite directions)
- can acquire, manipulate, use, produce, and/or transfer matter, energy or information.
- include processes that involve not mere change, but also **causation**
  - within the machine
    E.g. parts moving other parts, forces transmitted, information stored, retrieved, derived or transmitted, parts controlling or activating, other parts.
  - partly within the environment
    E.g. if there are sensors, motors, and communication channels
  - involving matter, motion, forces, energy, **information**, ... and more
- are usually embedded in a complex environment with which they interact. Often the boundary between machine and environment is different for different sub-systems of the machine.

  As every mathematician knows, you can use pen and paper as an extension of your mind.
  Sloman IJCAI 1971:
  `http://www.cs.bham.ac.uk/research/cogaff/04.html#200407`
- may include some internal processes whose effects are not externally detectable,
  e.g. a machine playing chess with itself and learning to play better as a result. In some cases the unobservable internal processes can be inferred indirectly. (More on this later)

What is a **physical machine** (PM)?

Some, but not all, machines satisfying the previous definition are physical.

If a machine and its operations (processes, and causal relationships) are fully describable using concepts of the physical sciences (plus mathematics), it is a physical machine (PM).

That’s a first draft specification.
(There is probably a variant definition that does not mention concepts, but I am not sure.)

The contents of the physical sciences, expand over time,
so the broadest notion of “physical machine” must refer to the indefinite future.

Examples of physical machines include:

levers, assemblages of gears, mechanical clocks, audio amplifiers, electronic devices, wireless control systems, clouds, tornadoes, plate tectonic systems, atoms, bacteria, brains, and myriad molecular machines in living organisms.

There is much we don’t know about what sorts of machine can be built out of chemical components. E.g. read recent issues of *Scientific American*.

Virtuality: absolute and relative

I shall first introduce a disinction between machines that are physical and machines that are not, even though the ones that are not are “fully implemented” in physical machines.

Later we’ll see that that is one of many examples of “layering” – one aspect of reality is layered on another, so that certain things are virtual relative to others.

But first we start with a single distinction.

Our preliminary notion of a running virtual machine (RVM), refined later, is defined as a machine in the sense defined earlier, but is not a physical machine, in the sense of “physical machine” defined above:

i.e. a RVM can be complex, with parts that interact with one another and with things outside the machine, but describing the parts and their operations requires use of concepts that are **not definable in terms of concepts of the physical sciences.**

E.g. spelling checker, chess program, proof checker, winning, invalid inference.

This notion will now be elaborated.

It is relative to a concept of a physical science, a concept that has changed over centuries, making the distinction a fluid one.

Non-physically-definable (NPD) concepts

Certain states, processes and interactions of some machines

cannot be described using **only** concepts
that are definable in terms of concepts
of the physical sciences
(E.g. the concepts of physics, chemistry, plus mathematics.)

Information-processing machines are examples.

“Information” is not used here in Shannon’s sense,
but in the sense that includes “reference”, “meaning”,
with properties and relations like:

truth, consistency, contradiction,

These concepts are not **definable** in terms of concepts of the physical sciences.

Though every information-using machine must be implemented (realised) in a physical machine.
See `http://www.cs.bham.ac.uk/research/projects/cogaff/misc/whats-information.html`

Non-physical machines include: socio-economic machines, ecosystems, many biological control systems, and many of the things that run in computers, including games, spelling checkers, operating systems and networking systems.

More on non-physically describable machines

Non-Physically-Describable Machines (NPDMs) are the subject matter of common sense, gossip, novels, plays, legends, history, the social sciences and economics, psychology, and various aspects of biology.

An important common features is use of non-physically definable concepts.

Examples of such non-physically-definable concepts:

"information", "inference", "contradiction", "strategy", "desire", "belief", "mood", "promise", "contract", "checking spelling", "file access violation", "sending email", "playing chess", "winning", "threat", "defence", "plan", "poverty", "crime", "economic recession", "election", "war", ...

For now I'll take that indefinability as obvious:
It would take too long to explain and defend.

This is connected with the falsity of "concept empiricism" and "symbol grounding theory". See

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

In computer science and software engineering NPDMs are often called "virtual machines"
(terminology possibly derived from some of the earliest examples: virtual memory systems).

This terminology is unfortunate – since it can suggest that such machines don't really exist – like the entities represented in virtual reality systems.

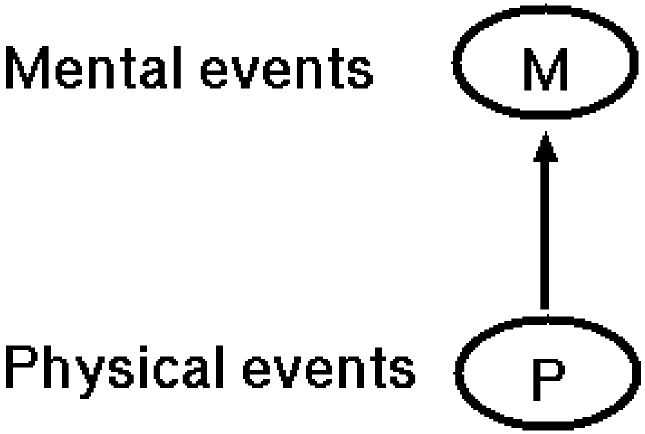
Nevertheless we are stuck with it.

(Like "Artificial Intelligence", which includes the study and modelling of natural intelligence.)

Later we'll talk about layers of relative virtuality.

The 20th C Philosophical breakthrough: Virtual machinery

Brief introduction to the philosophical significance of the technology of virtual machinery (not virtual reality) developed over the last six decades: Processes and events in running virtual machines can be causes and effects, despite being implemented in deterministic physical mechanisms.

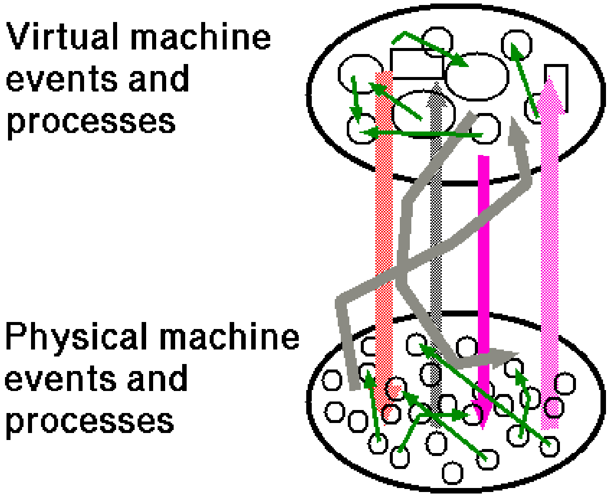


Mental events

M

Physical events

P



Virtual machine events and processes

Physical machine events and processes

The erroneous picture on the left implies that there is only one-way causation from physical to mental (epiphenomenalism).

As the picture on right indicates, we need to think about running virtual machinery that co-exists with, and influences, underlying physical machinery, which it helps to control, even though the virtual machinery is all fully implemented in the physical machinery.

I.e. running software can cause changes in physical hardware, just as increases in poverty can cause increases in crimes involving movement of stolen objects.

An example: memory management

A memory-management software process can detect a problem allocating a large portion of contiguous memory required by some running process.

That detection event can then trigger a "garbage-collection" process which:

- reclaims unused space and
- relocates many bit patterns,

  (by altering states of many switches in the computer's memory)
- while altering many redirection addresses so that reference relations are preserved.

The relocation of patterns from one part of physical memory to another does not require similar movement of matter, but does involve transfer of energy in signals that produce alterations of large numbers of bi-stable switches, e.g. when patterns are copied from one location to another and the old locations given new patterns.

Saying that the software events (detection of a need for space) cause certain physical changes is justified because there are many true conditional statements, including counterfactual conditionals, about what will happen if, what would have happened if, what would not have happened if, ... etc.

Their truth is the result of complex engineering design decisions.

This is not a translation of the statements about VM events causing hardware changes, but is a partial explication, and the truth of the conditionals helps to justify the claim about causal influences from RVMs to physical changes.

Granularity differs at different levels

An indivisible virtual machine process, such as copying of a symbol from one abstract memory location to another can involve a large number of physical changes: e.g. electrical pulses flowing along conductors and transistors being switched from one physical state to another.

This is a common feature of mappings from virtual machine structures and events to the physical structures events that underpin them: typically there are many distinguishable physical changes that correspond to each “minimal” virtual machine change.

Another feature is that the same virtual machine can make use of different physical machine processes on different occasions e.g.

- because the same structures are located in different parts of the machine at different times,
- because a faulty physical component can be replaced by another one using different technology, as a result of which a repetition of a previous virtual machine event can create a new physical process.

Some philosophers have assumed that when a non-physical process “supervenes” on a physical process the two processes are isomorphic.

This ignores the differences in granularity described above.

One reason why coarser granularity is sometimes **essential** for human developers and users, is to allow processes to be monitored and managed by human brains that would be unable to function with full low-level details.

That could apply to some but not all abstract processes.

Give some demos of RVMs

E.g.

- sheepdog,

  An interactive demo showing a virtual machine with a collection of concurrently active components (sheepdog, sheep) also causally linked to physical devices, e.g. computer mouse and screen (as well as internal registers, memory, etc.).

- “emotional agents”.

  Another interactive demo.

Some simple movies are here

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

(including showing non-interactive versions of the above – i.e. video-recordings of the RVMs not the running programs themselves):

The demos show that it is possible to generate a running program in which there are various sub-programs driving different entities, e.g. entities inhabiting a 2-D virtual space in which they move, that interact with one another in ways that are continually displayed on a screen, and with which a person can interact by using mouse or keyboard, directing actions selectively at different entities at different times: e.g. in the sheep-dog program, moving one of the sheep, one of the trees, or the dog, with consequent effects on other things in the virtual machine, because they sense the changes in the moved object (and other objects that move autonomously) and the sensed changes produce further reactions.

This sort of demo is not particularly impressive by current standards but depends on hardware and software developments in the last few decades: it would have been very difficult to produce such a collection of interacting pieces of virtual machinery four decades ago.

Some of the relevant technological advances

A small sample of technical developments supporting use of increasingly sophisticated RVMs in computing systems over the last half century is:

- The move from bit-level control to control of and by more complex and abstract patterns.
- The move from machine-level instructions to higher level languages (using compilers that ballistically translate to machine code and especially interpreters that "translate" dynamically, informed by context).

  A deep difference between compiled and interpreted programs: the compilation process makes the original program irrelevant, unlike an interpretation process: so altering interpreted program code at run time can have effects that would not occur if the program had been compiled and run.
- Memory management systems make physical memory reference context-dependent.
- Virtual memory (paging and cacheing) and garbage collection switch virtual memory contents between faster and slower core memories and backing store, and between different parts of core memory: constantly changing PM/VM mappings. (These support multiple uses of limited resources.)
- Networked file systems change apparent physical locations of files.
- Device interfaces translate physical signals into "standard" RVM signals and vice versa.
- Devices can themselves run virtual machines with buffers, memories, learning capabilities...
- Device drivers (software) handle mappings between higher level and lower level RVMs – and allow devices to be shared between RVMs (e.g. interfaces to printers, cameras, network devices).
- Context-dependent exception and interrupt handlers distribute causal powers over more functions.
- Non-active processes persist in memory and can have effects on running processes through shared structures. (It's a myth that single-cpu machines cannot support true parallelism.)
- Multi-cpu systems with relocatable RVMs allow VM/PM mappings to be optimised dynamically.
- Multiplicity of concurrent functions continually grows – especially on networked machines.
- Over time, control functions increasingly use monitoring and control **of RVM states and processes.**

Different requirements for virtual machinery

The different engineering developments supporting new kinds of virtual machinery helped to solve different sorts of problems. E.g.

- Sharing a limited physical device between different users efficiently.
- Optimising allocation of devices of different speeds between sub-tasks.
- Setting interface standards
  so that suppliers can produce competing solutions, and new technology can be used for old functions.
- Allowing re-use of design solutions in new contexts.
- Simplifying large scale design tasks by allowing components to "understand" more complex instructions (telling them what to do, leaving them to work out how to do it).
- Letting abstract functionality be instantiated differently in different contexts (polymorphism).
- Improving reliability of systems using unreliable components. (E.g. error-checking memory.)
- Allowing information transfer/information sharing to be done without users having to translate between formats for different devices (especially unix since mid 1970s).
- Simplifying tasks not only for human designers but also for self-monitoring, self-modulating, self-optimising, self-extending systems and sub-systems.

These are solutions to problems that are inherent in the construction and improvement of complex functioning systems: they are not restricted to artificial systems, or systems built from transistors, or ...

Conjecture: Similar problems were encountered in biological evolution (probably many more problems) and some of the solutions developed were similar, while some were a lot more sophisticated than solutions human engineers have found so far.

Benefits of using running VMs

We don’t know exactly what problems evolution faced, what solutions it came up with, and what mechanisms it used, in creating virtual machinery running in animals, to control increasingly complex biological organisms (and societies).

Perhaps we can learn from the problems human engineers faced, and the solutions they found.

- For example, a chess-playing computer can look for moves that achieve a win, or make a win likely.
- In performing that search it need not have the ability to represent the physical details of either the end result of such a search process or the physical details of the process of searching.
- Likewise a machine that aims to observe and understand processees running in itself is likely to do better by using a level of abstraction that has all the details needed for self monitoring, without representing all the physical details of the process.
- It may turn out to be essential for self-monitoring intelligent systems not to make any assumptions in advance about how they work.

Virtual machinery and causation

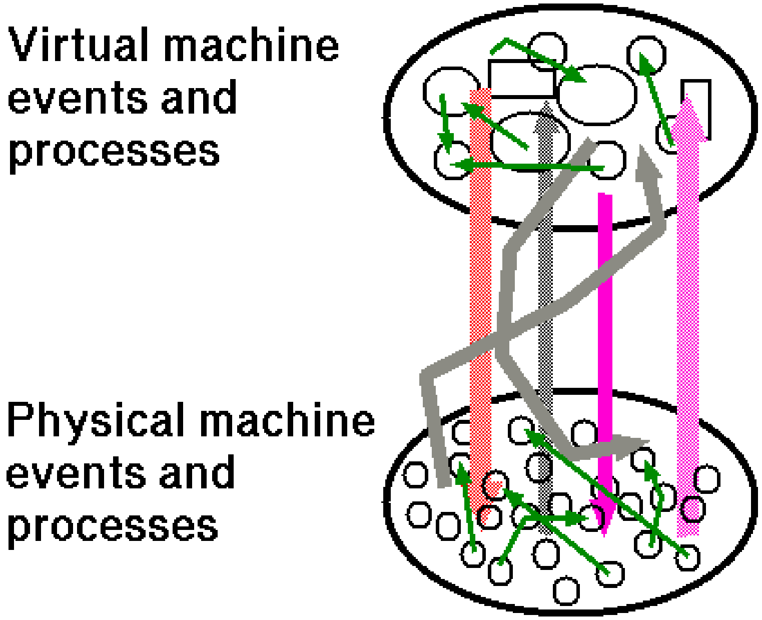
Virtual machinery works because “high level” events in a RVM can control both other virtual machinery and physical machinery.

Accordingly, bugs in the design of virtual machines can lead to disasters, even though nothing is wrong with the hardware.

As stated previously

Processes and events in running virtual machines can be causes and effects, despite being implemented in deterministic physical mechanisms.

Engineers (but not yet philosophers, psychologists, neuroscientists?) now understand how running virtual machinery can co-exist with, and influence, underlying physical machinery, which it helps to control, even though the virtual machinery is all fully implemented in the physical machinery.



Virtual machine events and processes

Physical machine events and processes

System designers who are concerned with providing the functionality, making it robust, or modifying it to cope with changing circumstances, often do not think only about the hardware involved.

They often think about, and write code to deal with, events like: information arriving, a context changing, a request being received, a plan being executed, a rule being followed with unexpected results, etc.

In doing that they depend on vast amounts of technology some of which they may not even be aware of, e.g. the use of garbage collection.

How does all that work?

- We have learnt how to set up physical mechanisms that enforce constraints between abstract process patterns (unlike mechanisms that merely enforce constraints between physical or geometric relations).
- Chains of such constraints can have complex indirect effects linking different process-patterns.
- Some interactions involve not only causation but also meaning: patterns are **interpreted** by processes in the machine as including descriptive information (e.g. testable conditions) and control information (e.g. specifying what to do).

  See "What enables a machine to understand?" (IJCAI 1985)
  `http://www.cs.bham.ac.uk/research/projects/cogaff/81-95.html#4`

Could biological evolution have solved similar problems?

BIOLOGICAL CONJECTURE: Over time, control functions increasingly used monitoring and control of VM states and processes.

CONJECTURE:

- The pressures for such developments that drive human engineers towards more and more “virtual” machinery of different sorts, were just as strong, in biological evolution
- as biological machines are their control functions became more and more complex
- with increasingly complex decisions taken “at run time”
- about how to process sensory information, what ontologies to use, what information to store, how to use the information, how to generate hypothese, goals, plans, etc.
- how to use them
- how to detect bugs in them, and debug them, ...

Controlling astronomically large numbers of interacting tiny physical subsystems directly is far too difficult.

But the solution involving RMVs depends crucially on finding the right, re-usable, levels of abstraction, and modules.

Whether there are any, and what they are depends on the type of environment.

Physical $\Longleftrightarrow$ virtual interfaces at different levels

Starting with simple physical devices implementing interacting discrete patterns, we have built layers of interacting patterns of ever increasing spatial and temporal complexity, with more and more varied functionality.

- Physical devices can constrain continuously varying states so as to allow only a small number of discrete stable states (e.g. only two)

  (e.g. using mechanical ratchets, electronic valves (tubes), aligned magnetic molecules, transistors etc.)
- Networks of such devices can constrain relationships between discrete patterns.

  E.g. the ABCD/XY example: a constraint can ensure that if devices A and B are in states X and Y respectively then devices C and D will be in states Y and X (with or without other constraints).
  So, a device network can rule out some physically possible combinations of states of components, and a new pattern in part of the network will cause pattern-changes elsewhere via the constraints.
  Compare: one end of a rigid lever moving down or up causes the other end to be moving up or down.
- Such networks can form dynamical systems with limited possible trajectories, constraining both the possible patterns and the possible sequences of patterns.
- A network of internal devices can link external interfaces (input and output devices) thereby limiting the relationships that can exist between patterns of inputs and patterns of outputs, and also limiting possible sequences of input-output patterns.
- Patterns in one part of the system can have meaning for another part, e.g.
  - constraining behaviour (e.g. where the pattern expresses a program or ruleset) or
  - describing something (e.g. where the pattern represents a testable condition)
- Such patterns and uses of such patterns in interacting computing systems may result from design (e.g. programming) or from self-organising (learning, evolving) systems.
- Some useful patterns need not be describable in the language of physics.

The “gap” closing

The last half century’s advances in development of **running** virtual machines at last provide a basis for closing what has been called “the explanatory gap”.

(But only if the right notion of virtual machine is used: a concept many people understand only dimly.)

- This gap was identified as a real problem for Darwin’s theory of evolution, even among people who were convinced by evidence for evolution of physical forms

  (e.g. T.H. Huxley, though the physicist Tyndall had earlier discussed it at as a mystery)

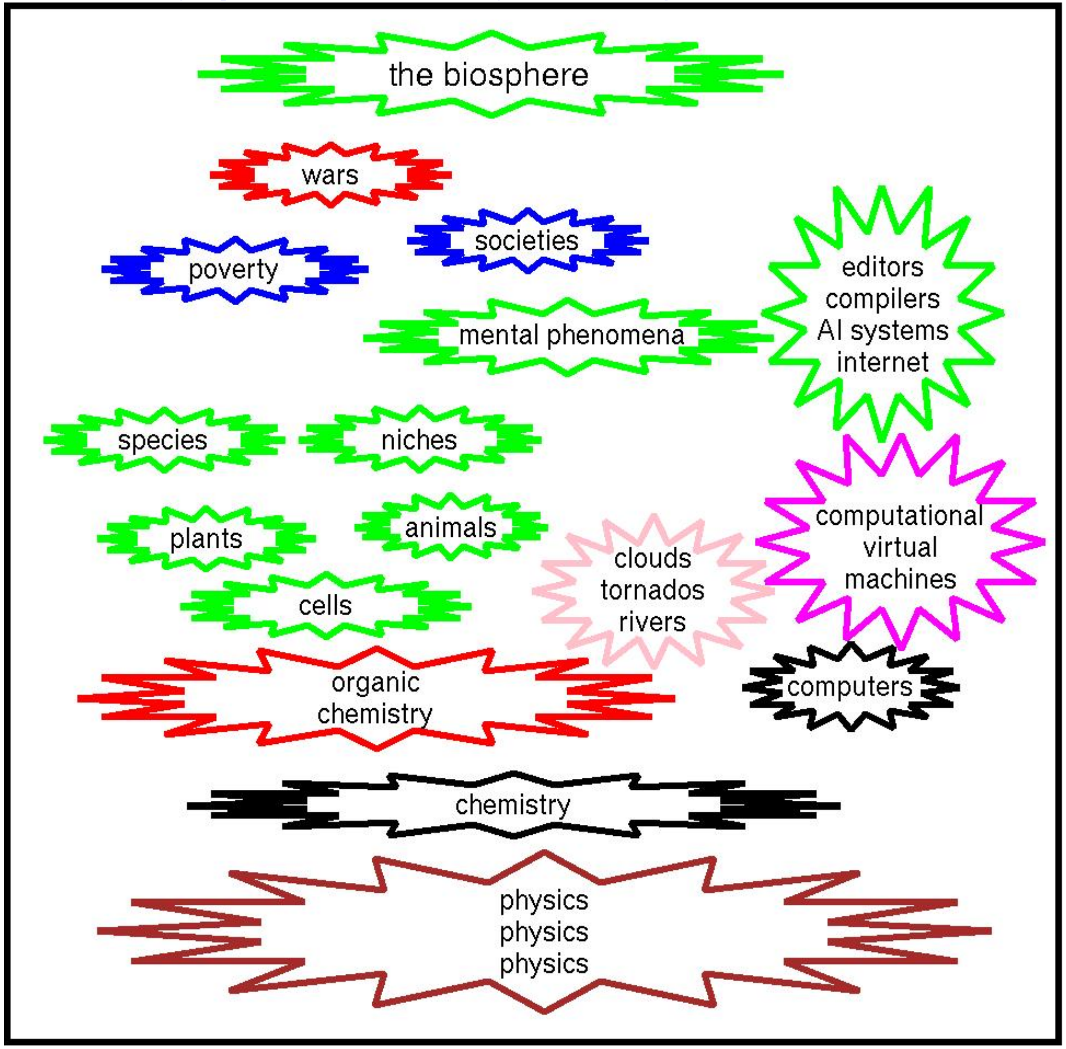
- The problem for Darwinians was that there was plenty of evidence for gradual evolution of physical forms between various species, including humans, but the transition from non-human to human minds seemed to involve such a big discontinuity that there was no comparable evidence.

- Further, some people thought it was inconceivable that an explanation of how physical bodies could produce minds would ever be forthcoming.

- Only now are we on the threshold of understanding what evolution might have had to do.

Virtual machines are everywhere

Many produced by biological evolution



How many levels of (relatively) virtual machinery does physics itself require?

What follows from all this?

- There are many empirical facts about human experience (most of them easily checked) that support claims about the existence of introspectively accessible entities, often described as privately accessible contents of consciousness.

  Various labels are used for these entities: “phenomenal consciousness”, “qualia” (singular “quale”), “sense-data”, “sensibilia”, “what it is like to be/feel X” (and others).
  For a useful, but partial, overview see `http://en.wikipedia.org/wiki/Qualia`

- What is not clear is what exactly follows from the empirical facts, and how best they can be described and explained.

- In the following slides I’ll demonstrate some of the empirical facts about contents of consciousness that raise a scientific problem of explaining how such entities arise and how they are related to non-mental mechanisms, e.g. brains (and future machines).

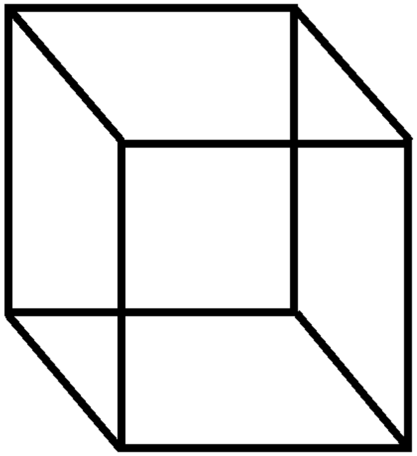
- Philosophers and scientists, understandably (in the past) ignorant of what we now know about virtual machinery have referred to “the explanatory gap” later redescribed by Chalmers as “the hard problem” of consciousness: a problem for Darwin

- Unfortunately, some philosophers, trying to characterise what needs to be explained have ended up discussing something incoherent.

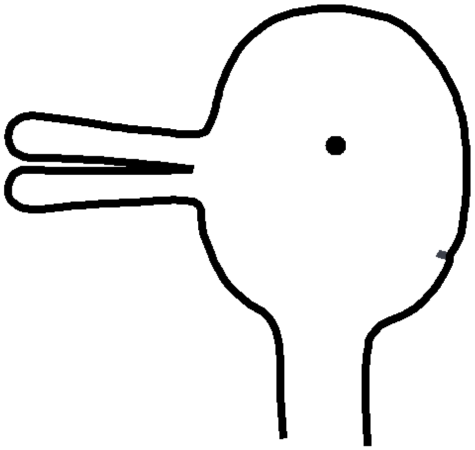
  E.g. qualia, or phenomenal consciousness **defined** as incapable of causal/functional relations.
  For more on this see my online presentations:
  `http://www.cs.bham.ac.uk/research/projects/cogaff/talks/`

Do some “consciousness science”

Stare at each of these two pictures for a while.



Necker cube



Duck-rabbit

Each is ambiguous and should flip between (at least) two very different views.

Try to describe exactly what changes when the flip occurs.
What concepts are needed for the different experiences?

In one case geometrical relations and distances change. In the other case geometry is unchanged, but biological functions change. Can a cube be experienced as “looking to left or to right”? If not, why not?

Nothing changes on the paper or in the optical information entering your eyes and brain.

Compare the kind of vocabulary used to describe parts and relationships in the two views of the Necker cube, and in the two views of the “duck-rabbit”.

So contents of consciousness can be three dimensional and can include information about entities with functional roles, mental states (e.g. looking left), ...

Empirical basis for referring to contents of consciousness

Introspection provides empirical evidence for mental contents “inside” us, distinct from external causes or internal physical processes:

- Ambiguous figures: as you stare at them, nothing changes out there, only the experiences/qualia (in your mind) “flip” (E.g. Necker cube, face-vase, old/young lady, etc.)

  This one rotates in 3-D in either direction: `http://www.procreo.jp/labo/labo13.html`
- Optical illusions (of many kinds): Muller-Lyer, Ebbinghaus, motion/colour after-effects.
- Dreams, hallucinations, hypnotic suggestions, effects of alcohol and other drugs.
- Different people see the same things differently. E.g. short-sighted and long-sighted people.

  Identical twins look different to people who know them well, but look the same to others.
  Cf. Botanical expertise makes similar plants look different. Colour blindness.
- Pressing eyeball makes things appear to move when they don’t, and can undo binocular fusion: you get two percepts of the same object; crossing your eyes can also do that.
- Put one hand into a pail of hot water, the other into a pail of cold water, then put both into lukewarm water: it will feel cool to one hand and warm to the other. (A very old philosophical experiment.)
- People and other things look tiny from a great height – without any real change in size.
- Aspect ratios, what’s visible, specularities, optical flow – all change with viewpoint.
- We experience only portions of things. A cube has six faces but we can’t see them all: what you experience changes as you move.
- Thinking, planning, reminiscing, daydreaming, imagining, can be done with eyes closed ....
- Composing poems, or music, or proving theorems with your eyes shut.
  Rich mental contents (not perceptual contents) can be involved in all of these.

We tend not to notice the diversity of content

The adjectival phrase “conscious of” is less deceptive than the noun “consciousness”.

We can be conscious of many very different things, with different implications (the concept is polymorphic):

E.g. there are great differences between being conscious of

- something moving towards you
- something looking at you
- a door being opened
- being close to a cliff edge
- being half asleep
- being horizontal
- being unpopular
- being 25 years old
- being in France
- being able to cook a meal without help
- being unknown to anyone in the room
- being deaf
- being more knowledgeable than most of the population
- being on the verge of a great discovery

Can one be conscious of: being unconscious, dead, or a lamp-post ???

Contents of consciousness can be inconsistent

This contradicts some theories of consciousness – e.g. Baars global workspace theory.

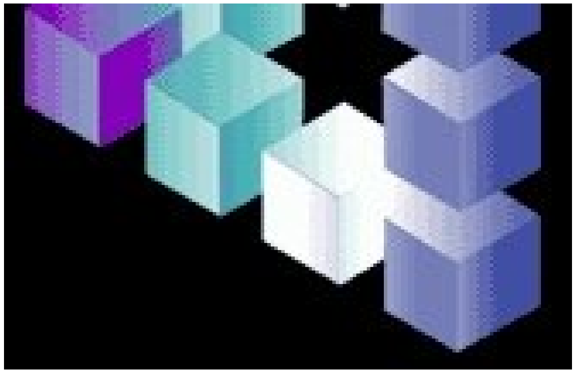
Motion after effects:

You are conscious of motion, but nothing is seen to change its location

More examples follow:

2-D and 3-D Qualia

Here (on the right) is part of a picture by Swedish artist, Oscar Reutersvärd (1934) which you probably see as a configuration of coloured cubes.

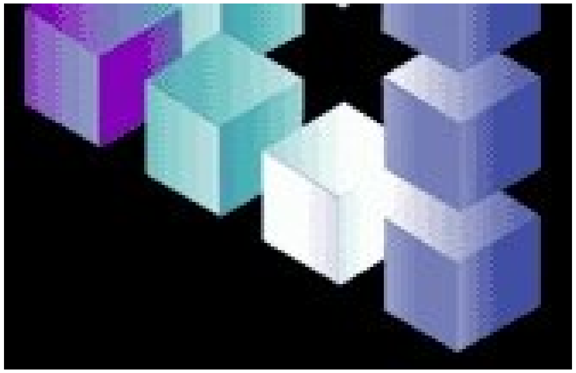


As with the Necker cube you have experiences of both 2-D lines, regions, colours, relationships and also 3-D surfaces, edges, corners, and spatial relationships.

You probably also experience various affordances: places you could touch the surfaces, ways you could grasp and move the various cubes (perhaps two are held floating in place by magnetic fields).

E.g. you can probably imagine swapping two of them, thinking about how you would have to grasp them in the process – e.g. swapping the white one with the cube to the left of it, or the cube on the opposite side.

2-D and 3-D Qualia

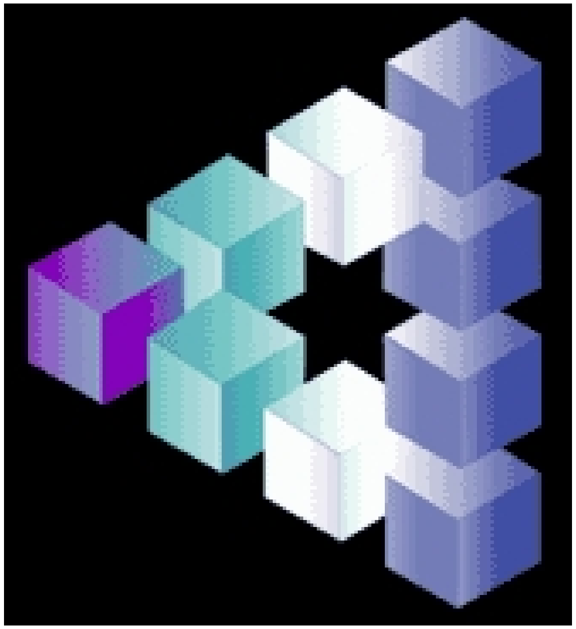


Here (on the right) is part of a picture by Swedish artist, Oscar Reutersvärd (1934) which you probably see as a configuration of coloured cubes.

As with the Necker cube you have experiences of both 2-D lines, regions, colours, relationships and also 3-D surfaces, edges, corners, and spatial relationships.

You probably also experience various affordances: places you could touch the surfaces, ways you could grasp and move the various cubes (perhaps two are held floating in place by magnetic fields).

E.g. you can probably imagine swapping two of them, thinking about how you would have to grasp them in the process – e.g. swapping the white one with the cube to the left of it, or the cube on the opposite side.



The second picture on the right (from which the first one was extracted) has a richer set of 2-D and 3-D contents.

Again there is a collection of 2-D contents (e.g. a star in the middle), plus experience of 3-D structures, relationships and affordances: with new possibilities for touching surfaces, grasping cubes, moving cubes.

The picture is outside you, as would the cubes be if it were not a picture. But the contents of your experience are in you: a multi-layered set of qualia: 2-D, 3-D and process possibilities.

But the scene depicted in the lower picture is geometrically impossible, even though the 2-D configuration is possible and exists, on the screen or on paper, if printed: the cubes, however, could not exist like that.
So your qualia can be inconsistent!

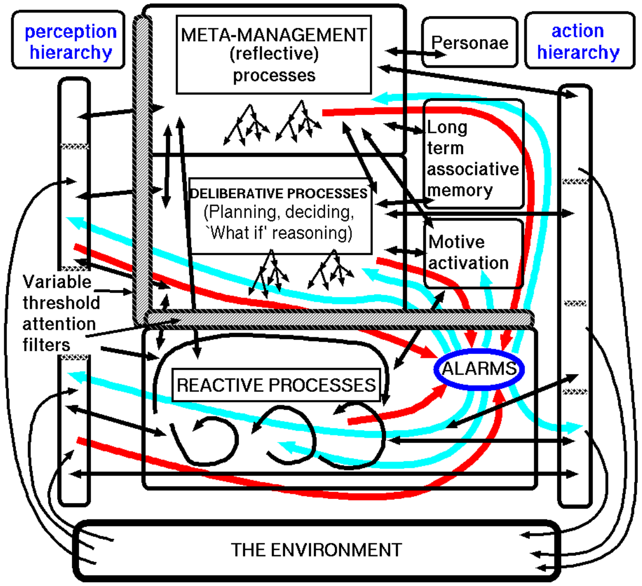
Architectural pre-requisites

We need good conceptual tools for talking about information-processing architectures.

The CogAff architecture schema crudely depicted on the left specifies types of components of an architecture.

Particular architectures can be specified by filling in the boxes and indicating connections (information flow and causal influences) between subsystems.

Perception	Central Processing	Action
	Meta-management (reflective processes) (newest)	
	Deliberative reasoning ("what if" mechanisms) (older)	
	Reactive mechanisms (oldest)	


perception hierarchy
META-MANAGEMENT (reflective) processes
Personae
action hierarchy
Long term associative memory
DELIBERATIVE PROCESSES (Planning, deciding, `What if' reasoning)
Motive activation
Variable threshold attention filters
ALARMS
REACTIVE PROCESSES
THE ENVIRONMENT

A particular way of filling in the boxes has been explored in the cognition and affect project as a move towards more human like systems.

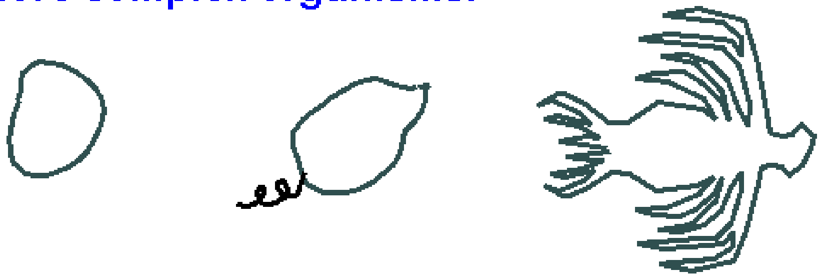
A crude schematic representation of that option, the H-CogAff architecture (strictly another, more constrained, architecture schema), is shown on the right.

For more detail see the CogAff papers and presentations. Also Marvin Minsky *The Emotion Machine*

A brief history of evolution!

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.

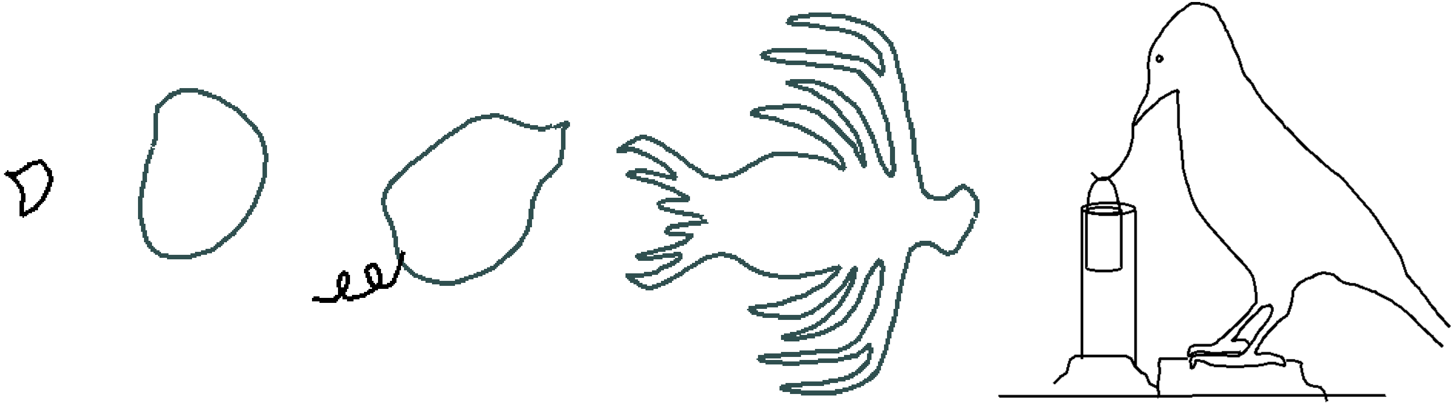
That achievement required the ability to acquire, process, and use *information*.

Growing environmental demands, and increasing complexity of the organisms themselves, led to many important changes in the requirements for information-processing mechanisms and architectures, forms of representation, and ontologies.

Eventually the complexity of the control problems demanded use of virtual machinery running on the biological physical machinery.

(Physical reconfiguration (e.g. neural re-growth) is much too slow for animals in dynamic environments!)

All organisms are information-processors but the information to be processed has changed and so have the means



Types of environment with different information-processing requirements

- Chemical soup
- Soup with detectable gradients
- Soup plus some stable structures (places with good stuff, bad stuff, obstacles, supports, shelters)
- Things that have to be manipulated to be eaten (e.g. disassembled)
- Controllable manipulators
- Things that try to eat you
- Food that tries to escape
- Mates with preferences
- Competitors for food and mates
- Collaborators that need, or can supply, information
- One’s own “buggy” planning, reasoning, etc.

What about the hard problem?

1. The "hard" problem can be shown to be a non-problem because it is formulated using a seriously defective concept (e.g. the concept of "phenomenal consciousness" defined so as to rule out cognitive functionality, or causal powers).
2. So the hard problem is an example of a well known type of philosophical problem that needs to be dissolved (fairly easily) rather than solved.

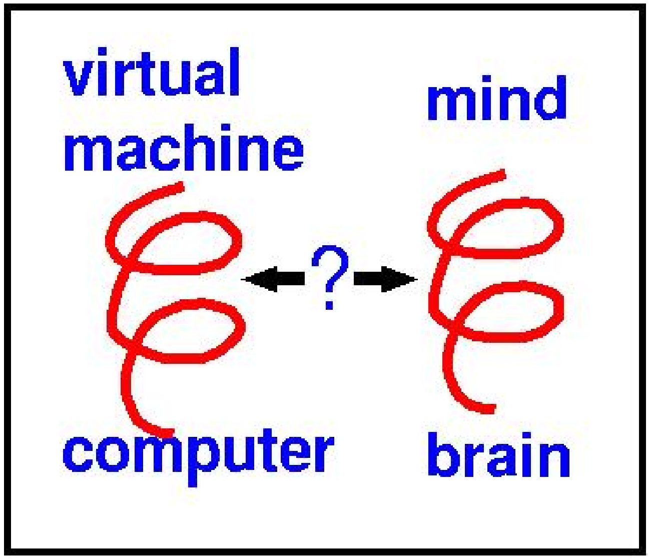
   For other examples, and a brief introduction to conceptual analysis, see
   `http://www.cs.bham.ac.uk/research/projects/cogaff/misc/varieties-of-atheism.html`
3. In contrast, the so-called "easy" problem requires detailed analysis of very complex and subtle features of perceptual processes, introspective processes and other mental processes, sometimes labelled "access consciousness": these have cognitive functions, but their complexity (especially the way details change as the environment changes or the perceiver moves) is considerable and very hard to characterise.
4. "Access consciousness" is complex also because it takes many different forms:

   what individuals can be conscious of, and what functions being conscious has, varies hugely, from simple life forms to sophisticated adult humans, and can vary between humans at different stages of development from conception to senile dementia. The concept is highly polymorphic.
5. Finding ways of modelling these aspects of consciousness, and explaining how they arise out of physical mechanisms, requires major advances in the science of information processing systems – including computer science and neuroscience.

Other parts of my website attempt to justify these claims

An old idea.

The idea of the analogy expressed in this diagram is very old, but we are only slowly understanding the variety of phenomena on the left hand side, extending our appreciation of what might be going on on the right.

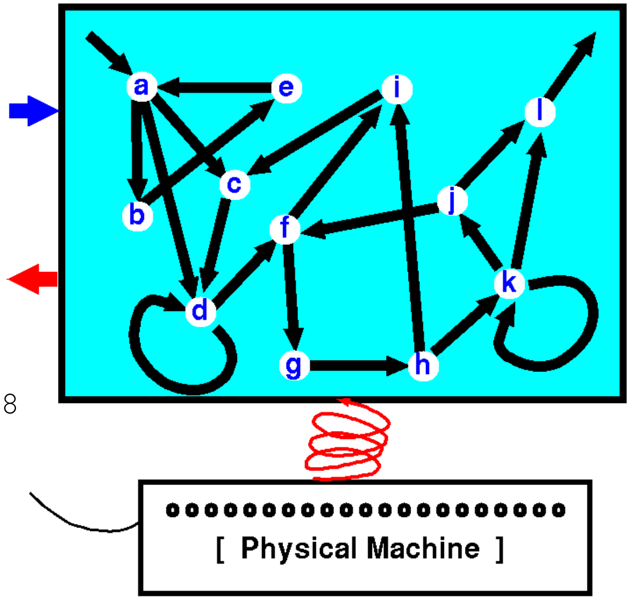


The simple-minded notion that the left hand side involves a program in a computer is seriously deficient, (a) because it ignores the requirement for the program to be running and (b) because we know now that there are far more types of information-processing system than a computer running a single program, as explained in other slides.

Simple-minded views about virtual machines lead to easily refutable computational theories of mind, e.g. the theory that virtual machines are simple finite state machines, as illustrated on the right ("Atomic state functionalism"). See

Ned Block: Functionalism `http://cogprints.org/235/`

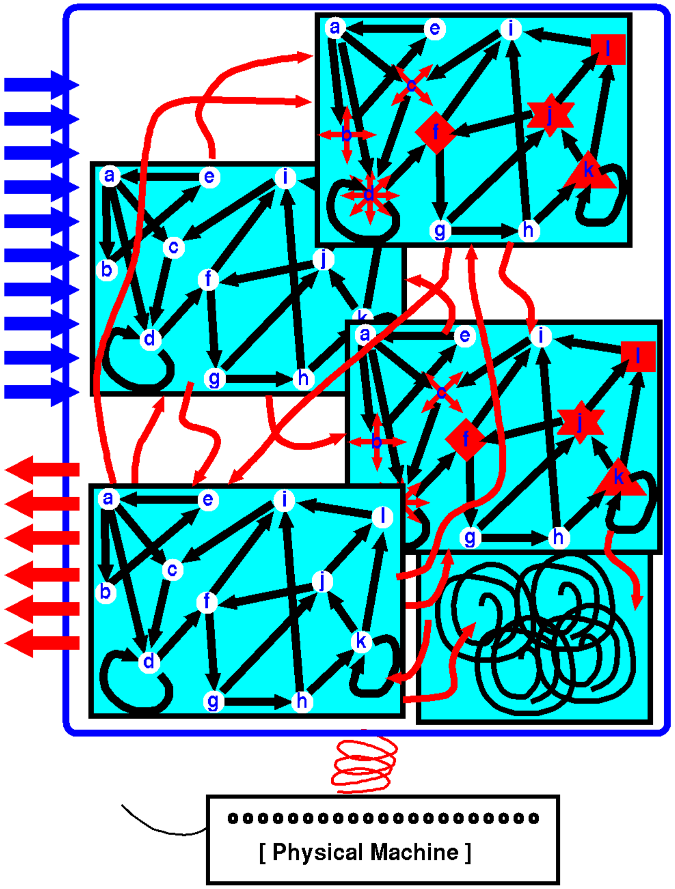
A. Sloman, The mind as a control system, 1993,
`http://www.cs.bham.ac.uk/research/projects/cogaff/81-95.html#18`

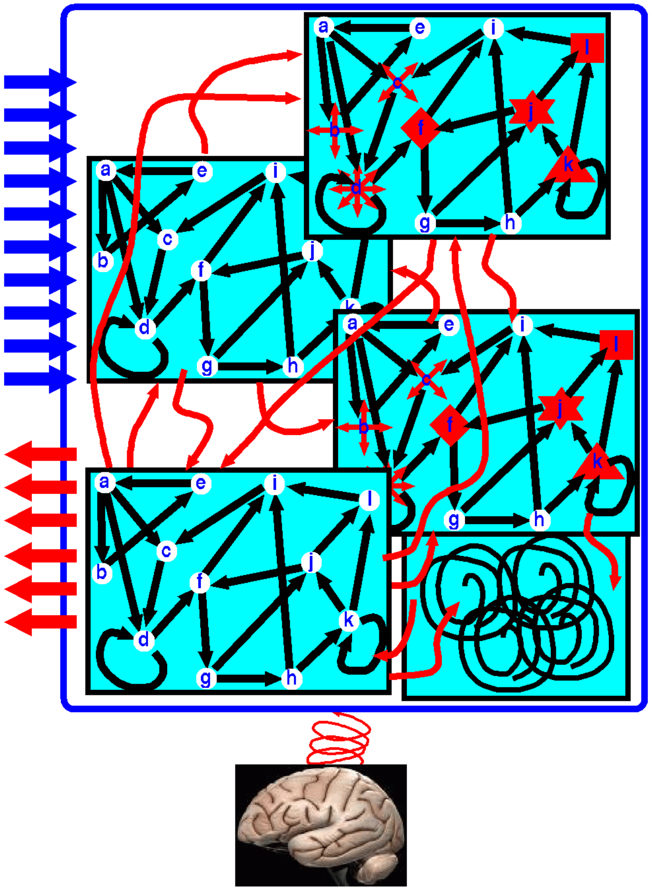


Forget what you have learnt about Turing machines: that's a simple abstraction which is surprisingly useful for theorising about classes of computations – but not so useful for modelling complex multi-component systems interacting asynchronously with a rich and complex environment.

More realistic models

As crudely indicated here we need to allow multiple concurrent inputs (blue) and outputs (red), multiple interacting subsystems, some discrete some continuous, with the ability to spawn new subsystems/processes as needed.





Artificial VM on artificial PM **Biological VM on biological PM**

In both cases there are multiple feedback loops involving the environment.

What it is like to be X

I find it astounding that so many thinkers treat the phrase "what it is like to be" as having sufficient precision to play a role in defining a serious problem for philosophy or science.

It was originally brought into philosophical discussion by Tom Nagel in "What is it like to be a bat?" in 1970

Compare my "What is it like to be a rock?"

`http://www.cs.bham.ac.uk/research/projects/cogaff/misc/rock`

As a piece of colloquial language the phrase is used in different contexts to express different things. E.g. there are great differences between what it is like to be

- half asleep
- horizontal
- unpopular
- 25 years old
- in France
- able to cook a meal without help
- drowning
- deaf
- more knowledgeable than most of the population
- unwittingly on the verge of a great discovery
- dead
- unknown

The phrase expresses a polymorphic concept: what is being said depends on the whole context in which it is being used, and there need not be anything in common to all the possible interpretations of "what it is like to be" – as with the word "conscious".

There is another way: notice that minds **DO** things

What many people forget when they discuss these issues is that minds don’t just contain bits of consciousness: they contain **processes** and those processes **do things**, including producing, modifying and aborting, other processes, mental and physical.

- In other words, minds are machines and the philosophical and scientific task is to find out what they do, and what sorts of physical and non-physical machinery they use.

  This is deeper and harder than asking what things minds contain, and how those things are produced, and what non-mental things correlate with them.

- Intelligent minds produce and use theories, questions, arguments, motives, plans, percepts: all of which are information structures (including descriptive and control structures).

- Such minds require information-manipulating machinery.

  Some people may think their minds contain only sequences of sentences and pictures, or sentence-like and picture-like entities.

  But sentences and pictures are merely two sorts of formats for encoding various kinds of information.

  A mind that understands sentences and pictures must have more than sentences and pictures, since if understanding sentences and pictures amounted to having and understanding more sentences and pictures, that would require an infinite regress of sentences and pictures.

- The implications of all that tend to be ignored in many discussions of consciousness.

- **Most researchers investigating these topics lack powerful conceptual tools developed in computer science/software engineering for creating, analysing, understanding information processing machinery of various kinds.**

Information processing models of mind

Some researchers have attempted to explain what consciousness is in terms of information-processing models of mentality, and they often assume that such models could be implemented on computers.

Several decades ago, Ulric Neisser in “The Imitation of Man by Machine” *Science* (1963)
`http://www.sciencemag.org/cgi/reprint/139/3551/193`

distinguished what Robert Abelson later called “cold cognition” (perception, reasoning, problem solving, etc.) and “hot cognition” (desires, emotions, moods, etc.).

Neisser claimed that the former could be replicated in computers but not the latter, offering detailed examples to support his case.

Herbert Simon responded to this challenge in “Motivational and emotional controls of cognition” *Psychological Review* 1967 `http://circas.asu.edu/cogsys/papers/simon.emotion.pdf`
arguing that computation can include *control* as well as “cold” cognition.

The idea was that emotions and other forms of “hot cognition” could be analysed as states and processes produced by control mechanisms, including motive generators and interrupt mechanisms.

My own earliest attempts were in *The Computer Revolution in Philosophy* 1978, e.g. chapters 6–10.
`http://www.cs.bham.ac.uk/research/projects/cogaff/crp/`
Similar ideas were put forward before that and after that, e.g. in P.N. Johnson-Laird’s book *The Computer and the Mind: An Introduction to Cognitive Science* (1988)

People objecting to such ideas claim that machines with all the proposed forms of information processing could still lack consciousness: no matter how convincing their behaviour, they could be mere “zombies” (defined in the next slide).

What's a zombie?

Many of the discussions of consciousness by philosophers refer to zombies: e.g. claiming that no matter how closely you build a machine that looks and behaves like a human and which contains a physical information-processing machine mimicking all of the functionality of a human brain it could still be a zombie.

The idea of a zombie occurs in a certain kind of grisly science fiction.

Here are some definitions:

- "A zombie is a creature that appears in folklore and popular culture typically as a reanimated corpse or a mindless human being. Stories of zombies originated in the Afro-Caribbean spiritual belief system of Vodou, which told of the people being controlled as laborers by a powerful sorcerer. Zombies became a popular device in modern horror fiction, largely because of the success of George A. Romero's 1968 film Night of the Living Dead."
- "zombi: a dead body that has been brought back to life by a supernatural force"
- "zombi: a god of voodoo cults of African origin worshipped especially in West Indies"
- "automaton: someone who acts or responds in a mechanical or apathetic way"

Many philosophers have come to regard the explanatory gap as a challenge to demonstrate that certain sorts of physical machines could not be zombies: can we find a design that **guarantees** that its instances have minds and are conscious, and do not merely **behave** as if they were conscious.

However if being conscious means having P-C that's an incoherent challenge!

(As explained in more detail elsewhere.)

Reactive vs deliberative interacting patterns

A Conway machine uses real or simulated concurrency: behaviour of each square depends only on the previous states of its eight neighbours and nothing else.

On a computer the concurrency is achieved by time-sharing, but it is still real concurrency.

Consider what happens when two virtual machines running on a computer compete in a chess game, sharing a virtual board, and interacting through moves on the board, each can sense or alter the state of any part of the (simulated) chess board.

- In general, programs on a computer are not restricted to local interactions.
- In some cases, the interacting processes are purely reactive: on every cycle every square immediately reacts to the previous pattern formed by its neighbours.
- If two instances of a chess program (or instances of different chess programs) interact by playing chess in the same computer, their behaviour is typically no longer purely reactive. Good ones will often have to search among possible sequences of future moves to find a good next move – and only then actually move.

  In addition, one or both of the chess virtual machines may do some searching in advance while waiting for the opponent's next move.
- Then each instance is a VM with its own internal states and processes interacting richly, and a less rich interaction with the other VM is mediated by changes in the shared board state (represented by an abstract data-structure).

  For more on varieties of deliberation see:
  `http://www.cs.bham.ac.uk/research/projects/cosy/papers/#dp0604`

Intentionality in a virtual machine

A running chess program (a VM) takes in information about the state of the board after the opponent moves, and builds or modifies internal structures that it uses to represent the board and the chess pieces on it, and their relationships, including threats, opportunities, possible traps, etc.

- In particular it uses those representations in attempting to achieve its goals.

  So, unlike the interacting Conway patterns mentioned earlier, some of the patterns in the chess virtual machine are treated by the machine as representations, that refer to something.

- During deliberation, some created patterns will be treated as referring to non-existent but possible future board states, and as options for moves in those states.

  They are treated that way insofar as they are used in considering and evaluating possible future move sequences in order to choose a move which will either avoid defeat (if there is a threat) or which has a chance of leading to victory (check-mate against the opponent).

- In this case the chess VM, unlike the simplest interacting Conway patterns, exhibits intentionality: the ability to refer. (NB. The programmer need not know about the details.)

  Since the Conway mechanism is capable of implementing arbitrary Turing machines, it could in principle implement two interacting chess virtual machines, so there could be intentionality in virtual machines running on a Conway machine – probably requiring a very big fairly slow machine.

- The intentionality of chess VMs is relatively simple because they have relatively few types of goal, relatively few preferences, and their options for perceiving and acting are limited by being constrained to play chess:

  For a human-like, or chimp-like, robot the possibilities would be much richer, and a far more complex architecture would be required. See
  `http://www.cs.bham.ac.uk/research/projects/cogaff/03.html#200307`

Adding meta-semantic competence

If a virtual machine playing chess not only thinks about possible board states and possible moves, and winning, moving, threats, traps, etc. but also thinks about what the opponent might be thinking, then that requires meta-semantic competences: the ability to represent things that themselves represent and use information.

- It is very likely that biological evolution produced meta-semantic competences in some organisms other than humans because treating other organisms (prey, predators, conspecifics to collaborate with, and offspring as they learn) as mere physical systems, ignoring their information-processing capabilities, will not work well (e.g. hunting intelligent prey, or avoiding intelligent predators).
- Another application for meta-semantic competences is self-monitoring, self evaluation, self-criticism, self-debugging: you can't detect and remedy flaws in your thinking, reasoning, planning, hypotheses etc. if you are not able to represent yourself as an information user.
- It is often assumed that social meta-semantic competences must have evolved first, but that's just an assumption: it is arguable that self-monitoring meta-semantic competences must have evolved first

  e.g. because an individual has relatively direct access to (some of) its own information-processing whereas the specifics of processing in others has to be inferred in a very indirect way (even if evolution produced the tendency to use information about others using information).
  See A. Sloman, 1979, The primacy of non-communicative language,
  `http://www.cs.bham.ac.uk/research/projects/cogaff/81-95.html#43`

How to think of qualia – in general

Insofar as a fly can be conscious of something approaching it rapidly, it can be said to have qualia, even if it does not know it has them, because it lacks meta-semantic competences.

A small subset of biological species seem to have an information processing architecture in which some parts can monitor, summarise, reason about the processes occurring in other parts, and perhaps in some cases debug, the processes.

Humans, in particular, are able to reflect on much of what they see, hear, smell, think etc (like noticing what flips when an ambiguous figure flips – e.g. the necker cube).

So humans not only have contents of consciousness, they are also able to learn and remember that they have had them, and compare them on different occasions, whereas the fly has sensory contents, but not the meta-cognitive detection or memory of the contents.

On that basis I tentatively define “qualia” as:

The contents of virtual machine states that refer to something, and which could, if the system had an appropriate information-processing architecture, be themselves the content of internal monitoring and control sub-systems.

This allows for the existence of qualia in portions of a mind that are normally inaccessible to self-conscious analysis.

However, some people may prefer to call those information contents “potential qualia” rather than actual qualia.

The labels don’t matter, as long as we can understand the functional roles, and evolutionary advantages in many cases, of these mechanisms: we must also understand differences between sub-cases.

Emerging varieties of functionality

Computer scientists and engineers and AI/Robotics researchers have been learning to add more and more kinds of control, kinds of pattern, and ways of interpreting patterns of varying levels of abstraction.

- A simple machine may repeatedly take in some pattern and output a derived pattern,
  e.g. computing the solution to an arithmetical problem.
- More complex machines can take in a pattern and a derivation-specification (program) and output a derived pattern that depends on both.
- Other machines can continually receive inputs (e.g. from digitised sensors) and continually generate outputs (e.g. to digitally controlled motors).
- More sophisticated machines can
  - solve new problems by searching for new ways of relating inputs to outputs, i.e. learning;
  - interpret some patterns as referring to the contents of the machine (using a somatic ontology) and others to independently existing external entities, events, processes (using an exosomatic ontology)
  - extend their ontologies and theories about the nature and interactions of external entities
  - perform tasks in parallel, coordinating them,
  - monitor and control some of their own operations – even interrupting, modulating, aborting, etc.
    (Including introspecting some of their sensory and other information contents: qualia.)
  - develop meta-semantic ontologies for representing and reasoning about thinking, planning, learning, communicating, motives, preferences, ...
  - acquire their own goals and preferences, extending self-modulation, autonomy, unpredictability, ...
  - develop new architectures which combine multiple concurrently active subsystems.
  - form societies, coalitions, partnerships ... etc.
- Biological evolution did all this and more, long before we started learning how to do it.

Causal networks linking layered patterns

How can events in virtual machines be causes as well as effects, even causing physical changes?

The answer is

> **through use of mechanisms that allow distinct patterns of states and sequences of patterns to be linked via strong constraints to other patterns of states and sequences of patterns (as in the ABCD/XY example, and the Conway machines, mentioned above).** (Some VMs may use probabilistic/stochastic constraints.)

What many people find hard to believe is that this can work for a virtual machine whose internal architecture allows for divisions of functionality corresponding to a host of functional divisions familiar in human minds, including

- interpreting physical structures or abstract patterns as referring to something (intentionality)
- generation of motives,
- selection of motives,
- adoption of plans or actions,
- perceiving things in the environment,
- introspecting perceptual structures and their changes,
- extending ontologies,
- forming generalisations,
- developing explanatory theories,
- making inferences,
- formulating questions,
- and many more.

Biological unknowns: Research needed

Many people now take it for granted that organisms are information-processing systems, but much is still not known, e.g. about the varieties of low level machinery available (at molecular and neuronal mechanisms) and the patterns of organisation for purposes of acquiring and using information and controlling internal functions and external behaviours.

Steve Burbeck's web site raises many of the issues:

> "All living organisms, from single cells in pond water to humans, survive by constantly processing information about threats and opportunities in the world around them. For example, single-cell E-coli bacteria have a sophisticated chemical sensor patch on one end that processes several different aspects of its environment and biases its movement toward attractant and away from repellent chemicals. At a cellular level, the information processing machinery of life is a complex network of thousands of genes and gene-expression control pathways that dynamically adapt the cell's function to its environment."
>
> `http://evolutionofcomputing.org/Multicellular/BiologicalInformationProcessing.html`
>
> "Nature offers many familiar examples of emergence, and the Internet is creating more.
>
> The following examples of emergent systems in nature illustrate the kinds of feedback between individual elements of natural systems that give rise to surprising ordered behavior. They also illustrate the trade off between the number of elements involved in the emergent system and the complexity of their individual interactions. The more complex the interactions between elements, the fewer elements are needed for a higher-level phenomenon to emerge. ... networks of computers support many sorts of emergent meta-level behavior because computers interact in far more complex ways than air and water molecules or particles of sand ... Some of this emergent behavior is desirable and/or intentional, and some (bugs, computer viruses, dangerous botnets, and cyber-warfare) are not."
>
> `http://evolutionofcomputing.org/Multicellular/Emergence.html`

Biological conjecture

I conjecture that biological evolution discovered those design problems long before we did and produced solutions using virtual machinery long before we did – in order to enable organisms to deal with rapidly changing and complex information structures (e.g. in visual perception, decision making, control of actions, self-monitoring etc.).

- You can't rapidly rewire millions of neurons when you look in a new direction
- or when you switch from approaching prey to deciding in which direction to try to escape from a new predator, using visible details of the terrain.
- So there's no alternative to using virtual machinery.
- But we know very little about biological virtual machinery.

  Nobody knows how brain mechanisms provide virtual machinery that supports proving geometric theorems, thinking about infinite sets of numbers, or algebra, or wanting to rule the world.
- The visual competences demonstrated here remain unexplained
  `http://www.cs.bham.ac.uk/research/projects/cogaff/misc/multipic-challenge.pdf`

We know that humans can use very different languages to say and do similar things (e.g. teach physics, discuss the weather); but evolution could not have produced special unique brain mechanisms for each language (since most are too new) – it's more likely that language learning creates specialised VMs running on more general physiological mechanisms.

Some conjectures about evolution of language are here:
`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#glang`

Work to be done:
Biology, psychology, neuroscience, robotics

There is much work still to be done.

That includes finding out precisely what the problems were that evolution solved and how they are solved in organisms, and why future intelligent robots will need similar solutions.

There are more slide presentations on related topics here:

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

Many of the papers in the Birmingham CogAff project (Cognition and Affect) are relevant, especially papers on architectures.

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

But the problem of explaining how a genome can specify types of virtual machinery to be developed in individuals, including types that are partly determined by the environment at various stages of development is very difficult.

We need to understand much more about the evolution and development of virtual machinery.

See Jackie Chappell and Aaron Sloman, “Natural and artificial meta-configured altricial information-processing systems”
`http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0609` (IJUC, 2007)

Closing Huxley’s Explanatory Gap

If we can learn more about:

- varieties of virtual machinery and their roles in generating, controlling, modulating and extending behaviours in organisms;
- how they are implemented in various types of biological organism;
- how their features can be specified in a genome (e.g. the control mechanisms for mating, web-making, and eating in a spider seem, for many species to be genetically determined, although specific behaviours are adapted to the precise details of environment);
- how in some species the virtual machinery instead of being fully specified genetically is built up within an individual as a result of operating of genetic, environmental and cultural processes (see Chappell and Sloman, IJUC, 2007, mentioned above);
- how and why self-monitoring mechanisms came to include mechanisms able to focus on intermediate information-structures within sensory/perceptual sub-systems

  (e.g. how things look, how they feel, how they sound, etc.)

then we may be able to understand how a Darwinian evolutionary process that is already demonstrably able to explain much of the evolution of physical form might be extended to explain evolution of information processing capabilities, including the phenomena that lead to philosophical theories of consciousness.

But we should not expect there to be **one** thing, one **it** that evolved.

Darwin and his contemporaries knew nothing about virtual machines, alas.

Importance and implications of VMs

There are additional slides available on Virtual Machines, e.g.

`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#mos09`
Virtual Machines and the Metaphysics of Science Expanded version of presentation at: Metaphysics of Science'09)

Topics include:

- Explanation of the importance of virtual machines in sophisticated control systems with self-monitoring and self-modulating capabilities.
- Why such machines need something like "access consciousness"/qualia – and why they too generate an explanatory gap – a gap bridged by a lot of sophisticated hardware and software engineering developed over a long time.
- In such machines, the explanations that we already have are much deeper than mere correlations: we know **how** the physical and virtual machinery are related, and what difference would be made by different designs.

More to read

We need much better understanding of nature-nurture issues, and requirements for educational systems.

John McCarthy on “The well-designed child”.
`http://www-formal.stanford.edu/jmc/child.html`

Chappell and Sloman on “Natural and artificial meta-configured altricial information-processing systems”
`http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0609`

Expand on varieties of metacognition, and differences between introspection and other aspects of metacognition.
`http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0803`

See other presentations in

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

and CogAff and CoSy papers:

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

`http://www.cs.bham.ac.uk/research/projects/cosy/papers/`

Further Reading

Novelists have traditionally regarded themselves as the experts on consciousness, with some justification. See for example, David Lodge’s essays and his novel on consciousness:
David Lodge, *Consciousness and the Novel: Connected Essays,* Secker & Warburg, London, 2002.
David Lodge, *Thinks ....*, Penguin Books, 2002.

A huge and important topic: disorders of consciousness, self-consciousness and control.

We need to explain development of kind of self-awareness that enables people to tell the difference between what they treat as empirical generalisations and what they understand as (mathematically) provable – e.g. facts about topological relations, geometry, mechanics, and numbers. (The roots of mathematical thinking.)