On the nature of causation in digital computer systems

Alan Turing Centenary talk

George Ellis
University of Cape Town
Prologue: The Deep Mystery

How did life come about the universe?
How did computers come about?
- The most complex things that humans have created

There was no complexity 14 billion years ago!
How did this come about?

By emergence of complex structures – cells, animals, human beings, where they did not exist before – through processes of natural selection: leading to intellects that are causally effective in the physical world

What are the mechanisms allowing this emergence?
- In historical terms?
- In functional terms?
Fine tuning: Just Six Numbers

1. $N = \frac{\text{electrical force}}{\text{gravitational force}} = 10^{36}$
2. $E = \text{strength of nuclear binding} = 0.007$
3. $\Omega = \text{normalized amount of matter in universe} = 0.3$
4. $\Lambda = \text{normalized cosmological constant} = 0.7$
5. $Q = \text{inhomogeneous seeds for cosmic structures} = \frac{1}{100,000}$
6. $D = \text{number of spatial dimensions} = 3$

Fourteen billion years after the big bang.
Expansion history

- Afterglow Light Pattern 380,000 yrs.
- Dark Ages
- Development of Galaxies, Planets, etc.
- Dark Energy Accelerated Expansion
- Inflation
- Quantum Fluctuations
- 1st Stars about 400 million yrs.
- Big Bang Expansion 13.7 billion years

Time
Expansion history

- Quantum fluctuations
- Unpredictable galaxies

(time)

- Afterglow Light Pattern 380,000 yrs.
- Dark Ages
- Development of Galaxies, Planets, etc.
- Inflation
- Dark Energy Accelerated Expansion
- 1st Stars about 400 million yrs.
• **The cosmic context:** Physics cannot give a complete causal account of what happens today from initial data in the early universe because random quantum fluctuations determined which galaxies would exist

• **The content of this talk cannot be implied by initial data:** quantum uncertainty makes it impossible, for even existence of the Earth is not guaranteed by that data. Quantum uncertainty means that *physical outcome is not uniquely determined even in principle*

• **Meaning comes into being that did not exist at earlier times. This is possible only because the higher levels develop their own autonomous powers independent of the lower levels of causality and structure,**

• **Allowing us to exist and have minds that are causally effective.**
• **As allowed by the possibility space created by the underlying physics**
SECTION 1: Creating viable complex systems

Herbert A Simon: The sciences of the artificial

Grady Booch: Object oriented analysis and design
Complexity and Structure

True complexity, with the emergence of higher levels of order and meaning, occurs in modular hierarchical structures because this is the only viable ways of building up real complexity on the basis of a set of simple components (Booch).

A hierarchy, with many layers of structure built upon each other, represents different levels of abstraction, each built upon the other, and understandable by itself.

There will be a different description and vocabulary suitable at each level of the hierarchy, related to the effective entities that occur at that level.

It is composed of modules that are quasi-autonomous. They are combined in a specified way to form the structure.
Hierarchy: A core feature of biology
Hierarchy

• The key to handling complexity is *hierarchical information structure and analysis*, and associated physical structuring.

• A *hierarchy* represents a decomposition of the problem into constituent parts, and into processes to handle those constituent parts, each requiring less data and processing, and more restricted operations, than the problem as a whole.

• The levels of a hierarchy represent *different levels of abstraction*, each built upon the other, and each understandable by itself. This is the phenomenon of *emergent order*.

• The success of hierarchical structuring depends on:
  (a) implementing modules to handle lower-level processes,
  (b) integration of these modules into a higher-level structure.
“We find separate parts that act as independent agents, each of which exhibit some fairly complex behaviour, and each of which contributes to many higher level functions. Only through the mutual co-operation of meaningful collections of these agents do we see the higher-level functionality of an organism. This is emergent behaviour – the behaviour of the whole is greater than the sum of its parts” (and cannot even be described in terms of the language that applies to the parts)

“Intra-component linkages are generally stronger than inter-component linkages. This fact has the effect of separating the high-frequency dynamics of the components – involving their internal structure – from the low-frequency dynamics – involving interactions amongst components” (this is why we can sensibly identify the components)
Modularity: Abstraction and Naming

**Abstraction:** Unable to master the entirety of a complex object, we choose to ignore its inessential details, dealing instead with a generalised idealised model of the object. *An abstraction* denotes the essential characteristics of an object that distinguishes it from all other kinds of objects.

It focuses on the **outside view of the object**, and so serves to separate its essential behaviour from its implementation; it emphasises some of the system’s details or properties, while suppressing others. “Information has to be thrown away by the billion bits all the time, because all the alternatives cannot be examined”.

Key feature: *Compound objects can be named and treated as units by appropriate labelling*. This leads to the power of abstract symbolism, recursion, and symbolic computation.

e.g. Euclidean geometry proof (Tony Hoare)
Modularity: Encapsulation

**Encapsulation and Information Hiding**: consumers of services only specify what is to be done, leaving it to the object to decide how to do it; this is an aspect of decentralisation of control.

**Encapsulation** is when the **internal workings are hidden from the outside**, so its procedures can be treated as black-box abstractions. “No part of any complex system should depend on the internal details of any other part”.

It involves **information hiding** – hiding all the internal aspects of an object that do not contribute to its essential characteristics. Thus variables are declared locally; they are not visible from outside their scope. In biology: cell walls contain interior molecules and mechanisms.

e.g. cells in animals; subroutines in programs
“Cellular compartmentalisation is an effective way to build gene circuits capable of complex logical operations, in which binary inputs are converted into binary outputs according to user defined rules.”

“It conceals the implementation of each encapsulated logic gate, which can be individually designed and optimized. To assemble a complex multicellular circuit, one need only be concerned with the input-output function of each cellular gate, and the output-input matching between layers.”

“The same variable can assume different roles in different subroutines without mutual dependency, thus lowering the demand for variable diversity. “

[In digital computers: **scoping of variables**]
Modularity: Inheritance

- *Inheritance* is an important feature of an “is a” hierarchy:

- it allows an object class to inherit all the properties of its superclass, and to add further properties to them (it is a `is a’ hierarchy): inheritance with variation

Language and databases:
- How we classify things

Key to biological evolution and development
- For example: cells specialise to neurons.

Computer systems:
- Object oriented programming makes this feature explicit
What about computers?

Orthogonal hierarchy systems:
Tanenbaum and Booch hierarchies
Andrew Tanenbaum: *Structured computer organisation* (two incarnations!)

**The vertical hierarchy**
Computers: Hardware (Tanenbaum)

Level 7: Global Network
Level 6: Local Network
Level 5: Computer
Level 4: Motherboard, Memory banks
Level 3: CPU, memory circuits
Level 2: ALU, primary memory, bus
Level 1: Logic circuits, Registers
Level 0: Transistors, resistors, capacitors
Level -1: Atoms
Level -2: Nucleons
Level -3: Quarks
Level -4: Superstrings

What determines what happens?
Software and Information

Level 7: Applications programs
Level 6: Problem oriented language level
Level 5: Assembly language level
Level 4: Operating system machine level
Level 3: Instruction set architecture level
Level 2: Microarchitecture level
Level 1: Digital logic level
Level 0: Device level

Data, Classes, Objects
Program structure
Symbolic names
Virtual memory, paging
Machine language states
Microprogram states
Gates, registers
Transistors, connectors

In broad correspondence with the hardware

The software and data power the hardware
The outcome depends on the specific software and data
The logical and data cross-cutting hierarchy (Booch)

Level 7: Problem oriented language level
Level 6: Assembly language level
Level 5: Operating system machine level
Level 4: Instruction set architecture level
Level 3: Microarchitecture level
Level 2: Digital logic level
Level 1: Device level
Level 0: Program structure
Level 6: Symbolic names
Level 5: Virtual memory, paging
Level 4: Machine language states
Level 3: Microprogram states
Level 2: Gates, registers
Level 1: Transistors, connectors

In broad correspondence with the hardware

The software and data power the hardware
The outcome depends on the specific software and data
Like all truly complex systems, ordinary language has a **modular hierarchical structure**:

- Text (book/ magazine/ newspaper/…)
- Sections
- Chapters
- Paragraphs
- Sentences (the dog in the room bit the cat)
- Phrases (the dog in the room, …)
- Words (sit, saturate, emphasize, ….)
- Phonemes (hat, bit, cat, ….)
- Letters (a, b, c, d, e, ….)

Each higher level provides the **context for the next level down**, within which its meaning is interpreted.

**Computer languages** have a similar hierarchy (Object Oriented or not): **extraordinary power of binary code**

**Biology**: So does genetic code (quaternary code)
Computer modelling hierarchy: the model reflects the hierarchy of the system being modelled. *Virgínia Brilhante J. Braz. Comp. Soc. vol.11 no.2 Campinas Nov. 2005*
Aerial image understanding is an important field of research for tackling the problems of automated navigation, large scale 3D scene construction, and object tracking for use in event detection.

http://vcla.stat.ucla.edu/Aerial_Image_Parsing/index.html
Hierarchical and Contextual Model:
To handle the large structural variations of aerial images we model scenes as groups of like objects, such as cars aligned in rows or roofs clustered into city blocks we present an algorithm to automatically add statistical constraints to the hierarchy in a minimax entropy framework.
Causation in computers: Control

Level 7: Applications programs
Level 6: Problem oriented language level
Level 5: Assembly language level
Level 4: Operating system machine level
Level 3: Instruction set architecture level
Level 2: Microarchitecture level
Level 1: Digital logic level
Level 0: Device level

The software drives the hardware:

Physics is controlled by the logic of the algorithms
Imbedded in these often: an is-a hierarchy with inheritance:

- Animals
- Dogs
- Alsations
- Henry

Used for classification and understanding
- indexes, library classification systems,
- enables understanding by class properties and associated expectations (“it’s a dog, it might bite”)

Basis for evolutionary development:
- specialisation by inheritance with modification

- Cells
- Neurons
- Pyramidal neurons
The logic that drives it all ......
Levels 5 and 6 are in the OSI model

*Computer Networking: a Top-Down approach: Kurose/Ross*

The higher levels drive the lower levels: Their logic determines what happens  **The software drives the hardware**
Computer networking

The process is top down at the source,
Bottom up at the destination
Results in effective same level action
Encapsulation takes place: extra information added at each level on the way down, stripped on the way up
SECTION 2: Bottom up and Contextual Effects.

True complexity emerges through the interplay in this hierarchy of bottom-up and contextual effects.

Does not contradict the laws of physics: uses them by setting constraints

(as well as evolution equations, constraint equations are a key feature of physics)

Almost all causal effects are in fact contextual: altering context alters outcomes

5 Types of contextual effects
Bottom-up causation alone:
Micro forces determine what happens at the higher levels

- Micro-physics underlies macro physics, e.g. kinetic theory of gases, theory of solids (conduction, thermal capacity)

- Physics and chemistry underlie the functioning of the brain
Computers: Bottom-up action

Lower level action underlies what happens at higher levels in each hierarchy

• Variables assigned, values expressed in bits

• Evaluation of expressions to update variables

• Sequential outcomes build up subroutines

• Looping and Branching build up logical layers

• Sequential Action of Subroutines give programs

• High level logic emerges from combinations of simple actions
Contextual effects occur when the higher levels of the hierarchy causally effect what happens at the lower levels, in a coordinated way.

This occurs by higher level features setting the context for lower level actions, the resulting constraints thereby creating new possibilities.

- The initial higher level state (described by high level variables) then influences the dynamics; if this state is changed, the lower level processes and consequent high level outcome changes.

- This is what characterizes it as an contextual effect. It is ubiquitous in the real world.

Royal Society Interface Focus Theme issue: Top-down causation 2 (1) : 1 - 140 (2012) Ed: Ellis, Nobel, O’Connor
Contextual effects

Hazard warning:

Previously “Top-down causation”

Does NOT mean Top-down Design
Is not necessarily deterministic

Contextual effects ↔ Top-down effects

Whole part constraints

They occur whenever the context has a demonstrable influence on the lower level dynamics and hence on system outcomes, e.g. Turing’s morphogenetic patterns (boundary conditions!)

Additionally the higher level variables change causal dynamics at the lower levels. This allows inter-level feedback loops.

Claim 1: Emergence of genuine complexity is possible only because of top down causation.

Claim 2: There are 5 different types of contextual effects
Contextual effects in the hierarchy of causation works

- by altering the nature of lower level entities through setting higher level contexts

- by setting constraints for lower level causation, channels interactions which paradoxically creates new possibilities

Gas: random motions

Network: channeled interactions
Structures constrain and so channel causal effects
Reliable behaviour of complex systems: arises from carefully designed complex structuring
Details matter, they change the macro behaviour

The micro-details essentially alter the macro behaviour
- You can’t coarse grain to determine macro laws
Structures constrain and so channel causal effects

Reliable behaviour of complex systems:
arises from carefully designed complex structuring
Details matter, they change the macro behaviour

This is what is missing in cellular automata –
broken symmetries: anisotropies are introduced
at the micro level by the macro system structure
(connections imply a network of preferred directions)

The micro-details essentially alter the macro behaviour
- You can’t coarse grain to determine macro laws
TYPE 1: Predictive algorithmic causation

Data uniquely determines outcome (given context)

Biology:

• Epigenetics: reading of genome to determination amino acids from DNA base pairs
• Top down because context dependent
  (Scott F. Gilbert and David Epel: *Ecological Developmental Biology: Integrating Epigenetics, Medicine, and Evolution*)

Computers:

• accounting systems
• dynamical systems simulations \(\rightarrow\) attractors
• physics simulations/ random data: Monte Carlo simulations,
• using artificial neural nets, once trained
Hardware, Software, and Data

• The *physical structuring* of the computer (hardware) embodies patterns of connection that constraints what happens at gate level
• The loaded *high level software* establishes further constraints on the logical structure of the lower level interactions
• Finally the *data* establishes sufficient further constraints on the lower level interactions to give a unique output:

\[(\text{physical structure, program, data}) \rightarrow \text{output}\]

In a given context: the first two will be fixed and unchanging in a given run (same high level software loaded) and can be taken for granted then. So, *given this context*: the given constraints imply

\[(\text{data}) \rightarrow \text{gate level operations} \rightarrow \text{output}\]

Change program: changes data $\Rightarrow$ changes output relation (Turing)
Purely algorithmic yet able to model (arbitrarily?) complex systems

Reliable Predictions? Not necessarily:
• Rounding errors: chaos, catastrophes, phase changes

A crucial difference is between autonomous systems – no external data influences it once it has started – and interactive systems – those where external data is received and changes what happens in the system.

e.g. stock control systems tied to bar code readers

The first are deterministic as systems in themselves (initial data determines the outcome); the second are not.
TYPE 2: Non-adaptive information control

Goals determine outcomes

Feedback control (*cybernetic systems*)

Examples - a thermostat

Information flows through specifically constructed feedback loops

*It is contextual (emergent) because if you disconnect the parts it won’t work. The system is connected so as to give the outcome.*
The role of goals in dynamics

The nature of causality is different when feedback control systems are guided by goals

**Standard Physics**

(physics, equations of state, initial conditions) → (outcomes)

(initial conditions) → (outcomes)

**Feedback control systems**

(physics, physical structure, goals) → (outcomes)

(goals) → (outcomes)

• The outcome of a feedback control system is determined by the goals rather than the initial data
Feedback control: *Cybernetic systems*

**Ubiquitous in Biology:**
- blood pressure, body temperature
- internal workings of cell

**Computer:**
- stock control system that automatically sends out orders
- aircraft automatic pilots
- chemical plant control
Adaptive processes take place when many entities interact and variation takes place in the properties of these entities, followed by selection of preferred entities that are better suited to their environment or context.

Higher level environments provide niches that are either favorable or unfavorable to particular kinds of lower level entities; those variations that are better suited to the niche are preserved and the others decay away.

Example: polar bear of black bear.

A different environment will lead to a different outcome. Hence this is a contextual effect.
Biology:

• evolutionary development of genetic information
• Basic learning processes

Computers:

• Stock control: choosing what items to order
• Training neural nets
• Genetic algorithms

Many others: see George Dyson: *Darwin’s Machines*
The role of selection in dynamics

*The nature of causality is different when adaptive selection takes place*

Adaptive selection (once off or repetitive)

(physics, physical structure, adaptive criteria) \(\rightarrow\) (outcomes)

But now there is randomness in the ensemble

(adaptive criteria) \(\rightarrow\) (unique outcomes)

• The outcome of a selective system is not determined uniquely by the selection criteria or the initial data

• Particularly as there may be multiple adaptive peaks in the selection landscape: you may end up on any of them
Any optimisation routine:
There may be multiple peaks in the fitness landscape: you may end up on any of them

Random processes in biology allow freedom to adapt
Turing: allow random processes in computation
Selection is the way meaningful information is created for a jumble of disordered objects
- Everything you need to know is there but it’s hidden in the ensemble: you have to select what is relevant

**Notional picture:**

- Meaningful information is gained by discarding all the information received that is not meaningful
- e.g. deleting files and emails
- Local entropy is decreasing as order increases (Maxwell Demon)
Erasing information is a dissipative process (Landauer)
Where do the goals in a feedback control system come from?

Case when goals are determined by adaptive selection

Animals: Pavlovian conditioning

Evolution: Genetically determined goals determine how homestasis works: blood pressure, body temperature

Computers:
- Evolutionary game theory models
TYPE 5: Adaptive selection of selection criteria

Where do selection criteria come from?
- choosing criteria for by adaptive selection!

E.g: selection of SPAM filter criteria
    chess playing programs: criteria of play

Reflection on purpose: meta-analysis
==> symbolically based adaptive selection

Reasoning about goals based in symbolic
representation of relevant issues, purpose, and
meaning via language and mathematics

Intelligent top-down causation.
TYPE 5: Adaptive selection of selection criteria

Where do selection criteria come from?
- choosing criteria for by adaptive selection!

E.g: selection of SPAM filter criteria
chess playing programs: criteria of play

Reflection on purpose: meta-analysis
==> symbolically based adaptive selection

Reasoning about goals based in symbolic representation of relevant issues, purpose, and meaning via language and mathematics

Intelligent top-down causation.
SECTION 4. The nature of computer programs

Computer programs are not physical entities, but are nevertheless causally effective in numerous ways, for example

• facilitating causal effectiveness of social conventions such as the rules of chess and the value of money,

• enabling application in engineering and science of non-physical entities such as mathematical relationships and our understanding of physical laws,

• facilitating social interaction through numerous applications of the internet.
Last three: are *complex adaptive systems*:

Murray Gellman: *The Quark and The Jaguar*

**Only adaptive selection creates new information**

Does so by selection and rejection

Clearing out what is not wanted an important part of creating useful information

- enable learning: new meaningful information collected and brought into play
- not possible in first two cases

(dynamical systems and social sciences)
Top down effects and complexity

**Hypothesis:** bottom up emergence by itself is strictly limited in terms of the complexity it can give rise to

**Conjecture:** Emergence of genuine complexity is characterised by a reversal of information flow from bottom up to top down


**Corollary:** degree of complexity that can arise by bottom-up causation alone is strictly limited.

Sand piles, the game of life, bird flocks, even do not compare in complexity with a single cell or an animal body. Some kind of central coordination is needed.
**Aircraft Design:** Plans for a Jumbo Jet aircraft result in billions of atoms being deployed to create the aircraft in accordance with those plans (intelligent to down causation).

This is a non-trivial example: it costs a great deal of money to employ experts in aerodynamics, structures, materials, fuels, lubrication, controls, etc. to design and then to manufacture the aircraft in accordance with those plans.

The plan itself is not equivalent to any single person’s brain state: it is an abstract hierarchically structured equivalence class of representations (spoken, drawn, in computers, in brains, etc.) that together comprise the design.

It is clearly causally effective (the aircraft would not exist without it). It could not occur without language and mathematics, as well as the social systems in which it is embodied.
Interactive Ray Tracing of Large Models Using Voxel Hierarchies
Attila T. Áfra  (Boeing 777)
The key analytic idea

In all cases, the key idea is that of functional equivalence classes: each equivalence class is a set of lower level states all that correspond to the same higher level state

- When you coarse grain, all of these lower level states correspond to the same higher level state
- Can replace the lower level elements, higher entity remains (e.g. water in river; cells in human body)

- Whenever you can identify existence of such equivalence classes, that is an indication that top-down causation is taking place
- “Top-down causation by information control: from a philosophical problem to a scientific research program” G Auletta, G.F.R Ellis and L Jaeger: J. R. Soc. Interface 2008 vol. 5 1159-1172
-- Philosophers: multiple realisability
Equivalence classes

An equivalence relation is a binary relation \( \sim \) satisfying three properties:
For every element \( a \) in \( X \), \( a \sim a \) (reflexivity),
For every two elements \( a \) and \( b \) in \( X \), if \( a \sim b \),
then \( b \sim a \) (symmetry), and
For every three elements \( a \), \( b \), and \( c \) in \( X \), if \( a \sim b \) and \( b \sim c \),
then \( a \sim c \) (transitivity).

The equivalence class of an element \( a \) is denoted \([a]\) and may be defined as the set of elements that are related to \( a \) by \( \sim \).

This is the ontological nature of the higher level effective entity:
It enters into the higher level effective relations

Entropy is a measure of number of lower level states (Penrose)
Computers: Equivalence classes all over the place

• at circuit level: using Boolean algebra to find equivalent circuits

• at implementation level: compiling or interpreting (completely different lower level functioning for same higher level outcome)

• running same high level software on different microprocessors

• at implementation level: equivalence of hardware and software (completely different nature of lower level entities for same higher level outcome)

• In each case this indicates top-down effects: the higher level function drives the lower level interactions, and does not care how it is implemented (information hiding!)
Hence although they are the ultimate in algorithmic causation, as characterized so precisely by Turing machines, digital computers embody and demonstrate the causal efficacy of non-physical entities.

**Physics allows, does not control**

**Emergence** of new kinds of causation out of physics not implied by physics but rather by logic of higher level possibilities

**Phenomenology**: described by effective higher level laws, and appropriate language

**Autonomous**: higher levels independent of lower levels; many realizations possible, not dependent on any specific one of them: just need one specific implementation
Don’t try to build a complex system straight off!

Develop it bit by bit: ongoing process of testing

• Let it evolve from simple to complex: testing and modifying all the time (a process of adaptive selection)

• Develop modules and see that they work: then connect them together

• Adapt working modules to new purposes

This process is one of the major reasons modular structuring is crucial in complex systems (H A Simon)
Three contexts of emergence:

1\textsuperscript{st}: Evolutionary history of the universe and the world: Once upon a time they did not exist!

2\textsuperscript{nd}: Developmental history of each living being: Once upon a time they were a single cell.

3\textsuperscript{rd}: Functional nature of each complex object: built up out of components that do not have the higher level properties.

\textit{Biology: Adaptive selection takes place in each case.}

\textit{In each case, it is contextual constraints that enable emergence of true complexity, which in turn enables more complex forms of top-down causation.}
In biological systems, individual phenotypes are typically adopted by multiple genotypes. Examples include protein structure phenotypes, where each structure can be adopted by a myriad individual amino acid sequence genotypes. These genotypes form vast connected ‘neutral networks’ in genotype space. The size of such neutral networks endows biological systems not only with robustness to genetic change, but also with the ability to evolve a vast number of novel phenotypes that occur near any one neutral network. Whether technological systems can be designed to have similar properties is poorly understood. Here we ask this question for a class of programmable electronic circuits that compute digital logic functions.
The functional flexibility of such circuits is important in many applications, including applications of evolutionary principles to circuit design. The functions they compute are at the heart of all digital computation.

We explore a vast space of $10^{45}$ logic circuits (‘genotypes’) and $10^{19}$ logic functions (‘phenotypes’). We demonstrate that circuits that compute the same logic function are connected in large neutral networks that span circuit space. Their robustness or fault-tolerance varies very widely. The vicinity of each neutral network contains circuits with a broad range of novel functions. Two circuits computing different functions can usually be converted into one another via few changes in their architecture.

These observations show that properties important for the evolvability of biological systems exist in a commercially important class of electronic circuitry. They also point to generic ways to generate fault-tolerant, adaptable and evolvable electronic circuitry.
“It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. That process could follow the normal teaching of a child. Things would be pointed out and named, etc.” (Turing, 1950).

Give it the ability to learn!
– Embodied intelligence  *Rodney Brooks and robots*

Adaptive Selection!

But then what guides learning?
What are the selection criteria?
Evolution of the human brain – Triune model

<table>
<thead>
<tr>
<th>Area</th>
<th>Evolution Period</th>
<th>Functions</th>
<th>Related Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-Complex (Reptile)</td>
<td>Triassic Period 248-206 million years</td>
<td>Hunger, temperature control,</td>
<td>Reptiles such as fish.</td>
</tr>
<tr>
<td></td>
<td>ago</td>
<td>fight-or-flight response.</td>
<td></td>
</tr>
<tr>
<td>Limbic System</td>
<td>Jurassic Period 206-144 million years</td>
<td>Mood, memory, hormone control.</td>
<td>Older mammals such as dogs, cats</td>
</tr>
<tr>
<td></td>
<td>ago</td>
<td></td>
<td>and mice.</td>
</tr>
<tr>
<td>Neocortex</td>
<td>Eocene &amp; Oligocene Epochs 55-24</td>
<td>Logic and thought required for</td>
<td>Monkeys and chimpanzees.</td>
</tr>
<tr>
<td></td>
<td>million years ago</td>
<td>complex social situations etc.</td>
<td></td>
</tr>
</tbody>
</table>

[http://lboeckl.net/model/cognitive.html](http://lboeckl.net/model/cognitive.html)
Emotion is crucial to the way the intellect functions. Examples show mind can’t function if emotional systems damaged. Why? They guide the intellect.
**The individual mind:** The emotional systems and value systems provide (competing) sets of selection criteria for thoughts and actions. Individual responsibility: choose between them.

**Emotions guide brain plasticity, as well as immediate focus.**
Emotional experiences guide our decision making.
In daily life: rationality plays a secondary role.
• Need computers with causally effective emotional systems, as well as emotive expressions
KISMET (MIT)
Rodney Brooks

- Learns by exploring
- Simulated emotional states
- Facial expressions
- Guides attention
- Used to guide intellectual development?

- Need these systems to guide “brain plasticity” and learning
Evolutionarily determined primary emotional systems shape the way the intellect works, functionally and developmentally. Provide the needed guidance for learning circuits.
Isaac Asimov, *I Robot* (over a million copies in print)

The Three laws of robotics:
Ethical guidelines for robot behaviour
The Three laws of robotics (Asimov):

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.

2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.

3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

A higher level value system constraining their behaviour: setting out what is not acceptable

Contrast:

computer that goes wrong: HAL (Arthur C Clarke, 2001)
Intelligent computers

1. Intellect = logic: rationality

2. Emotions = priorities

3. Values = allowed/preferred behaviour

4. Instinct = automatic responses

5. Intuition = fast pattern recognition and response (Kahneman)

6. Imagination = finding possibilities

7. Choice = selection of action option
Computers are dualist machines

- The hardware does nothing without the software
- which is not physical: it is relational
- It does not exist in the nature of the parts alone

- It is realised in energy states in the hardware
- Its essence is in the specific *patterns* of those energy states, which embodies the logic of the algorithms used
- that is, it is a *relational* rather than material entity

They function by a combination of bottom up causation and contextual affects (top-down influences) that are ubiquitous in biology and computer systems
This figure shows how gossip was found to flow among every layer of the Enron hierarchy. The thickness of the lines represents the level of gossip, which flowed down the hierarchy (left side of ladder), up the hierarchy (right side) and between employees of the same rank.

The nature of computer programs

Their development embodies the combined experience of numerous workers in aspects ranging from basic concepts to system design to effective algorithms to high level design patterns,

and is based in the extraordinary ability of digital systems to represent language, pictures, sound, mathematical relationships, and indeed all human knowledge.

Hence although they are the ultimate in algorithmic causation, as characterized so precisely by Turing machines, digital computers embody and demonstrate the causal efficacy of many kinds of non-physical entities.
1. $N = \text{electrical force/gravitational force} = 10^{36}$

2. $E = \text{strength of nuclear binding} = 0.007$

3. $\Omega = \text{normalized amount of matter in universe} = 0.3$

4. $\Lambda = \text{normalized cosmological constant} = 0.7$

5. $Q = \text{inhomogeneous seeds for cosmic structures} = 1/100,000$

6. $D = \text{number of spatial dimensions} = 3$

Postscript

We also do not know how the universe came into existence, whatever propaganda you may have heard.

We do not know how life came into existence: the key step (functioning DNA) is not understood; nor do we know how consciousness arose.