How to do AI-inspired biology, as a change from biology-inspired AI.

Some thoughts about the past and future of AI as science and philosophy

Aaron Sloman, School of Computer Science, UoB.

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

Slides here

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

Apologies and Notes

Apologies/warnings:

- My work straddles several disciplines and I cannot keep up with most of what has been written that is relevant.
  I welcome pointers to things I should have read or known about.
- My slides are too cluttered for presentations: I write them so that they can be read by people who did not attend the presentation.
  So please ignore what's on the screen unless I draw attention to something.

Notes

- In what follows the word "information" is not restricted to what is **true**.
  Some philosophers mistakenly think the idea of **false information** is inconsistent.
  They ignore the fact that it is possible to have or acquire **false**, or **partly incorrect** information, e.g. government propaganda and bad philosophy. There is also control information, which is neither true nor false.
- I am grateful to Gill Harris for helpful comments regarding the diversity of developmental routes to similar end-points in both normal children and children with genetic or other disabilities.
  `http://psychology-people.bham.ac.uk/people-pages/detail.php?identity=harrisg`
- Several of these ideas arise from or were refined by discussions with Margaret Boden, Jackie Chappell, Marvin Minsky, and various colleagues in AI/Robotics at Sussex and Birmingham. Some of the ideas come from John McCarthy's papers. Some from Immanuel Kant. Bernard Melzer kindly hosted me in his department in Edinburgh in 1972-3 where I think I learnt more in one year than at any time since I was a toddler. My early reading of Piaget had an influence.
- I produce my slides using Linux, Ved (in Poplog) and LaTeX, and use 'xdvi' or 'xpdf' for live presentations (with multiple virtual desktops, on linux).
  PDF versions are produced by pdflatex. (Some day I may switch to 'beamer'.)
  Lots more here: `http://www.cs.bham.ac.uk/research/projects/cogaff/talks/`

Summary

Some personal notes (not presented at the workshop).

AI as Science and Philosophy vs AI as engineering.

How to think about an AI/Cognitive science research roadmap.

All living things are information processors: informed control systems

We need a better understanding the problems evolution solved – in order to understand the solutions.

That's hard: many problems are invisible. So people focus on their favourite problem.

Vision is the hardest problem – to see and to solve.

Tradeoffs-between evolution and development
precocial vs altricial species,
preconfigured vs meta-configured competences (Chappell&Sloman 2005, 2007)

Some ways to think about evolution and development.

Embodiment is important but not in the way most embodiment theorists think.

The physical environment is a deeper driver of the evolution and development of animal intelligence than most people have noticed.

(Ulric Neisser and John McCarthy noticed. "Artificial General Intelligence" theorists mostly don't.)

Social drivers came much later.
(And built on the results)

Human language has evolutionary and developmental precursors required by other species and pre-verbal children: languages for thought (including perceiving, deciding, acting, ...) rather than for communication. (Generalised Languages: GLs)

Can AI become more collaborative and multi-disciplinary?

Some personal notes 1: Conversion

I am honoured to be part of this workshop on the Heritage of AI, though I am not part of the mainstream

(a) because I knew nothing about AI till Max Clowes came to Sussex university in 1969, by which time AI was already thriving, e.g. at MIT, CMU, Stanford, Edinburgh ..., and

(b) because my own work has been mostly on the side-lines of AI, partly because of differences of interest, and partly because of a different view on how to achieve some of our common goals.

My initial interest was in trying to find better ways to support the claim, defended in my 1962 Oxford DPhil thesis (now online), that Kant's philosophy of mathematics (mathematical knowledge is substantive but not empirical) was basically right, despite the apparent refutation by the discovery that Euclidean geometry does not describe physical space. (There's more to Euclid than the parallel axiom.)

My interest was in showing that there is a mode of mathematical reasoning (e.g. geometrical reasoning) that is different from both analysing definitions and their consequences (Hume?) and empirical discovery

though, as Imre Lakatos pointed out, mathematics is quasi-empirical, since exploration, mistakes and de-bugging of concepts, formalisations and proofs all play a role.

Recently I have been trying to draw out implications for evolution and developmental psychology, e.g.
`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#toddler`

Because a lot of mathematical reasoning is visual/spatial, I got very interested in AI research on vision (Max's specialism) and my first AI paper, at IJCAI 1971 in London, was a critique of the logicist philosophy of AI propounded by McCarthy and Hayes in 'Some philosophical problems from the standpoint of AI' (1969).

(My paper was reprinted in AIJ 1971, then in a book of readings edited by J.M. Nicholas, then as Chapter 7 in my 1978 book, and is now online.)

Max gave me things to read by Marvin Minsky, John McCarthy, Herb Simon, Allen Newell, and others. I was very impressed and deeply influenced in my thinking about philosophy – about what minds are and how they work, even when I disagreed. I became convinced that AI provided a better way to do philosophy.

I also sat in on Max's lectures and learnt to program in Algol60, though I read about other programming languages – including the IPL-V Manual by Newell et. al. which had many deep insights about software engineering and structured programming before Dijkstra's work, even though the language IPL-V was terrible, compared with later languages for AI.

Notes 2: A year in Edinburgh 1972-3

As a result of the IJCAI paper I was invited by Bernard Melzer (founding editor of the AI Journal) to spend a year at Edinburgh University (1972-3), funded by SRC. That provided an opportunity for me to become proficient at programming in Pop-2, though I also learnt Lisp (taught by Bob Boyer and J.S Moore), and Popler – the Pop-2 implementation of Hewitt's Planner (implemented by Julian Davies).

I also read some general books on computing and programming languages and learnt about virtual machines for the first time in the context of designs for machines that could implement other machines.

That was one of my major mind-extending experiences that year, and I am still trying to work out the ramifications.
http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#m0s09

While in Edinburgh I met Danny Bobrow who spent some time on a visit prior to moving from BBN to Xerox PARC. He used Pop-2 to implement a version of Logo that was later used for teaching AI in Edinburgh, and I learnt a lot about Logo and other things by talking to him. Bob Boyer and J Moor gave me a half-day tutorial on Lisp (to demystify it). The lisp was implemented in Pop-2.

Seymour Papert visited and Sylvia Weir (also from South Africa) introduced us. He taught me a way to think about how gyroscopes work that he said I would never forget. He was right.

During that year the Freddy II robot developed in Donald Michie's group (by Popplestone, Ambler, Barrow, Brown, Burstall and others) was demonstrated (in a speeded up video) assembling a number of toy objects made of wood (including a toy car with axles to be pushed through holes in the car body and wheels to be added to the ends of the axles). Amazing on a computer with about 256Kbytes RAM. There's now a wikipedia web site for Freddy. http://en.wikipedia.org/wiki/Freddy_II

While I was in Edinburgh I wrote a paper on the importance of architectures in the design of intelligent systems, made available as an Edinburgh AI memo, which became chapter 6 of my 1978 book:

http://www.cs.bham.ac.uk/research/projects/cogaff/crp/chap6.html

I first met Geoff Hinton that year: he was working on his PhD. This meeting resulted in his working with me in Sussex on the Popeye project, a few years later.

Notes 3: Back at Sussex – Pop-11 and CogSci

On my return to Sussex, Max Clowes, Margaret Boden and others were involved in starting up a new undergraduate programme in SOCS, the School of Social Sciences. The programme was called "Communication Studies", then later "Cognitive Studies". In it we taught AI, psychology, linguistics, and philosophy, to students majoring in all those subjects except AI which was only available in "Preliminary" and "Contextual" courses. Later we introduced an undergraduate major in AI, and shortly after that split off from social sciences to form a new school: COGS (Cognitive and Computing Sciences). I think ours was the first undergraduate degree in AI.

The university provided funds to buy a DEC PDP11/40 computer in 1975. Choosing a language for teaching AI to non-scientists was hard, but there were few options because the PDP11 was a new machine. Logo seemed a candidate. Danny Bobrow, was in Palo Alto and gave me access to a MAXC computer emulating a PDP-10 with Logo, which made me one of the first international internet users (typing on a paper 10cps terminal). I did a lot of experiments with LOGO, but decided the language was too primitive and too badly structured to meet our teaching needs – because beginners need powerful systems:

`http://www.cs.bham.ac.uk/research/projects/cogaff/81-95.html#45`

Lisp wasn't available, and in any case the syntax is not good for normal humans. Steve Hardy, who had been supervised by Mike Brady at Essex University joined us as a lecturer in AI, and implemented a version of Pop-2 which he called Pop-11, using a byte-code interpreter, on Unix, in a 32KB address space.

Inspired by Max Clowes, we prepared a lot of AI teaching materials about parsing, planning, vision, use of analogy, and general programming. Many of the teaching materials are still available in Poplog:

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

By around 1980 quite a lot of university departments had bought our Unix Pop-11 and were teaching AI using it in CS and Psychology departments, since DEC PDP11 computers running Unix were spreading fast. This began to produce a new breed of psychologist. Alas that trickle was killed off a decade later because the British Psychological Society failed to recognise AI as relevant to psychology.

Notes 4: POPLOG and COGS

Later Pop-11 was ported to a Vax running VMS, with a much larger address space, and with help from John Gibson it became incrementally compiled to machine code. Since the incremental compiler was available at run time, its subroutines were all made available to users, and that allowed Chris Mellish to implement an incrementally compiled Prolog, with full Prolog syntax, in Pop-11. People in academe and industry who wanted to do AI teaching, research or development on Vax/VMS started buying the system, which we renamed Poplog. Even Boeing and Lockheed bought copies. Income helped to pay for further development.

Later still, after further extensions to the Poplog virtual machine, an incremental compiler for Common Lisp was added, and after that an incremental compiler for ML. The Poplog Common Lisp compiler was the first one that passed all the tests for lexical scoping, according to a testing project in Edinburgh University.

I did not like being a salesman, so in 1983 sales of Poplog (initially on VMS, then later unix) were taken over by a commercial company (SDL), and later a spinoff, ISL, took over all sales and support while development and maintenance continued at Sussex university. Several academic and industrial research projects as well as much teaching made use of Poplog, including the ability to support mixed language programming, e.g. combining two packages implemented in different languages, the most common pair being Pop-11 and Prolog.

Mainly because of the software engineering genius of John Gibson, Poplog was very robust and surprisingly (to some) fast. E.g. a Lisp user at HP research labs in Bristol switched from using a Lisp machine to Poplog Lisp on a unix workstation because garbage collection was so much faster.

Sales of Poplog supported a substantial amount of AI teaching and research at Sussex and helped the growth of a school of Cognitive and Computing Sciences, including an undergraduate major in AI. Part of the reason for its success was the support in Poplog for collaborative development and hypertext in documentation from the early 1980s, making heavy use of the fact that the emacs-like editor Ved was implemented in Pop-11 and could interact directly with all the Poplog compilers.

I managed the development of Poplog from around the time Steve Hardy left to join Teknowledge (1982?) until I moved to the University of Birmingham in 1991.

Notes 5: Birmingham, AI Tools, Minsky

In 1991 I wanted a change and Birmingham offered me a research chair in the CS department. I persuaded them to offer an AI half-degree, which started in 1993, using Pop-11 and Poplog for the introductory teaching and project work. AI was available combined with CS, psychology, mathematics and Arts subjects.

I also had PhD students and I watched them struggling to develop AI systems and wasting much time because they all started from the programming language level (Pop-11, similar to Lisp).

So with their help, and later the help of other users, academic and non-acadmiec, I developed a toolkit, SimAgent, allowing them to build fairly complex architectures quite quickly. Unlike better known AI toolkits, like SOAR, PRS, Clarion, and others, SimAgent was not designed to support a particular architecture but to support exploration and experimentation with different architectures combining different sorts of components, reactive, deliberative, reflective/metamanagement, etc., as described here:

`http://www.cs.bham.ac.uk/research/projects/poplog/packages/simagent.html`
Some demos of teaching materials and videos of student projects based on SimAgent are here:
`http://www.cs.bham.ac.uk/research/projects/poplog/figs/simagent`

Later, around 1998 SPSS bought ISL for Clementine, which ISL had developed using Poplog. It was one of the most commercially successful data-mining packages – mainly because it allowed users to combine different AI and statistical techniques as needed, with a very user friendly interface, making heavy use of the features of Poplog, especially Pop-11. SPSS decided not to sell Poplog. So, with the agreement of Sussex University it became free and open source, hosted at Birmingham, where it is still available.

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

I assume I was invited to this workshop because of my interactions with Marvin Minsky.

Since the mid 1980s, I had been interacting with Marvin and others via usenet lists, and in 1995 invited Minsky and McCarthy to join me in a special session "A philosophical encounter" at IJCAI, in Toronto in 1995. Later I visited MIT (invited by Roz Picard) in 1996, and when Marvin was in the UK in 1999 invited him to talk at Birmingham. He came again in 2000, for AISB2000, and we have interacted intermittently since then at various workshops and by email.

We have shared views and interests, except that my focus is more on the space of natural and artificial possible minds, including evolutionary trajectories, and I think his is more on how to design a human-like system.

AI as Science and Philosophy vs AI as engineering.

It was obvious to the main founders of AI that

AI is as relevant to science (e.g. attempting to model aspects of human or animal mental functioning) as to engineering,

but not all who are involved in AI teaching and research appreciate that aspect.

- A scientific goal need not involve building some kind of complete intelligent machine, since the work can focus on a subset of what is required,

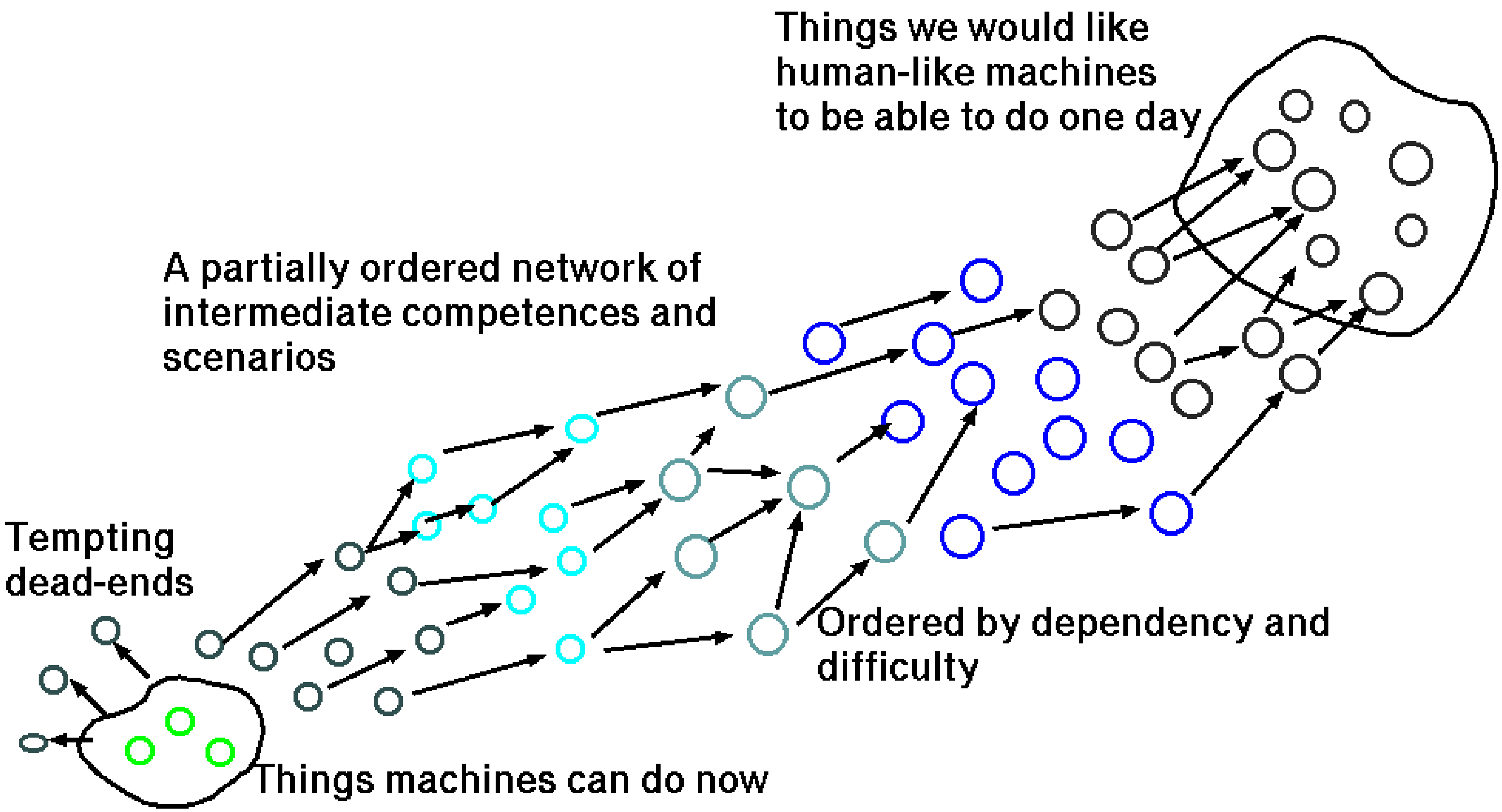
  as often happens in AI research on vision, language processing, learning, planning, reasoning, game-playing, etc.

- However, building models of some sub-system of a human or animal without ever testing their ideas in the context of a multi-component architecture

- can lead to a failure to notice that the models cannot account for all major aspects of the subsystem,

- including ways in which the subsystem cooperates with, supports, or interferes with other subsystems.

One way to think about an AI/Cognitive science roadmap.



But we must not think only about designing one type of system, e.g. a system with human-like capabilities.

- Deep understanding of any design requires understanding how it is like and unlike other designs. That requires comparative analysis of multiple designs.
- Moreover, even "human-like" covers a multitude of cases: neonates, infants, toddlers, teen-agers, professors of psychology, poets, philosophers, philatelists, people with genetic or trauma-induced abnormality, people deprived of any systematic education, etc.

Living things are informed control systems

Multiple designs are found in living things.

Living things are information processors.

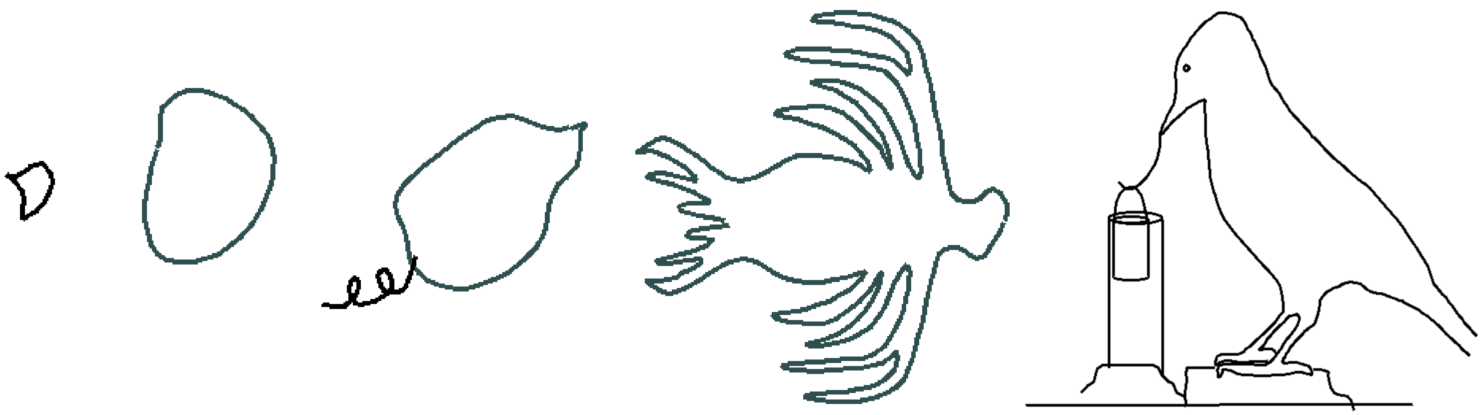
The world contains: matter, energy, information

Organisms are control systems:

They acquire and use information (external and internal),
in order to control how they use matter, energy and information

(in order to acquire more matter, energy and information,

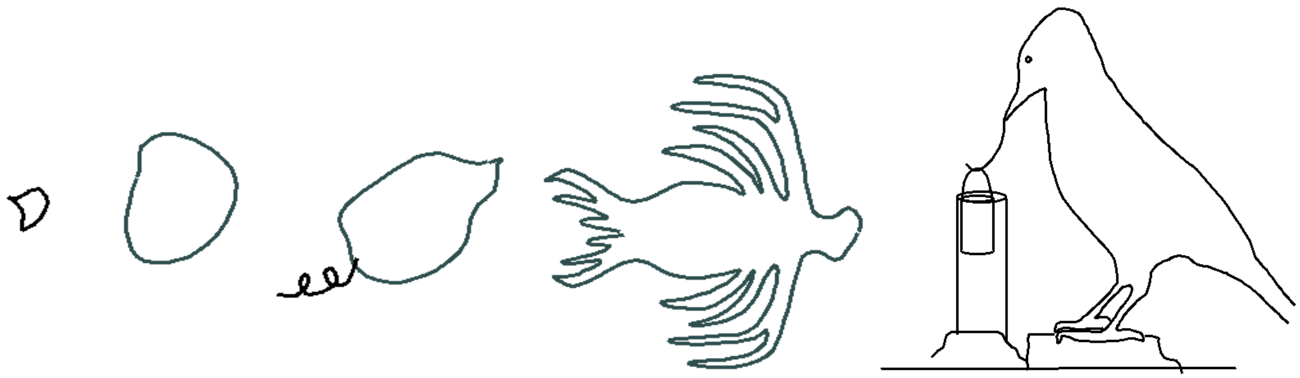
and also reproduce, repair, defend against intruders, dispose of waste products...).



Evolution produced more and more sophisticated information processors,
driven partly by changes in environments, partly by results of prior evolutionary history.

Because of the way organisms use information they are informed control systems.

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

(A small, illustrative, sample)

- 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.

The role of the environment

Ulric Neisser:

"We may have been lavishing too much effort on hypothetical models of the mind and **not enough on analyzing the environment** that the mind has been shaped to meet."

Neisser, U. (1976) *Cognition and Reality*, San Francisco: W. H. Freeman.

Compare: **John McCarthy**: "The well-designed child"

"Evolution solved a different problem than that of starting a baby with no a priori assumptions.

....

"Instead of building babies as Cartesian philosophers taking nothing but their sensations for granted, evolution produced babies with innate prejudices that correspond to facts about the world and babies' positions in it. Learning starts from these prejudices. What is the world like, and what are these instinctive prejudices?"

`http://www-formal.stanford.edu/jmc/child.html` Also in AI Journal, December 2008

All biological organisms are solutions to design problems that cannot be specified without specifying in detail the relevant features of the environment.

Turing, surprisingly got this wrong: he thought human-like learning was possible from a "clean slate".

J.J. Gibson understood the general point, but missed many important details.
Other relevant authors: Piaget, Fodor, Chomsky, Mandler, Keil, Gopnik, Tenenbaum, Thomasello, Karmiloff-Smith, Spelke, ...

To gain deep understanding, don't study just one species

Trying to do AI as science by developing/modelling only one type of system
(e.g. "human level AI")

is like trying to do physics by investigating only how things behave near the leaning tower of Pisa.

Or studying only the motion of planets.

we need to understand spaces of possibilities
and the tradeoffs between alternative designs: so look at different species.

Don't assume all effective information-processing mechanisms have to be rational
(as in Dennett's "intentional stance", Newell's "knowledge level".)

Evolution provided useful reflexes: reflexes are neither rational nor irrational.

Some useful reflexes are cognitive

That can include reflexes that generate goals/motives.

Not all goals are chosen because the individual knows associated rewards/costs/benefits:

See
`http://www.cs.bham.ac.uk/research/projects/cogaff/misc/architecture-based-motivation.html`
Architecture-based motivation vs reward-based motivation.

Example: Two ways to study architectures

1. Try to come up with a specification for a powerful, general (possibly human-like) architecture.

   vs

2. Try to understand ways in which architectures can differ –
   (The space of possibilities, and tradeoffs.)
   - what types of functionality (or combinations of types) they can support,
   - what design options there are for a given sort of functionality
   - what the tradeoffs are between the options
   - how different types could have evolved
   - how instances can develop or grow
   - what sorts of mechanism may or may not be available for particular sub-functions
   - what forms of representation are available and how they relate to particular sub-functions
   - tradeoffs between more or less parallel implementations.
   - tradeoffs between prior "hardware" commitments and run-time resource allocation.
     e.g. special hardware for meta-cognition or sharing?

Doing only (1) will produce limited understanding of the system studied, simulated, etc.

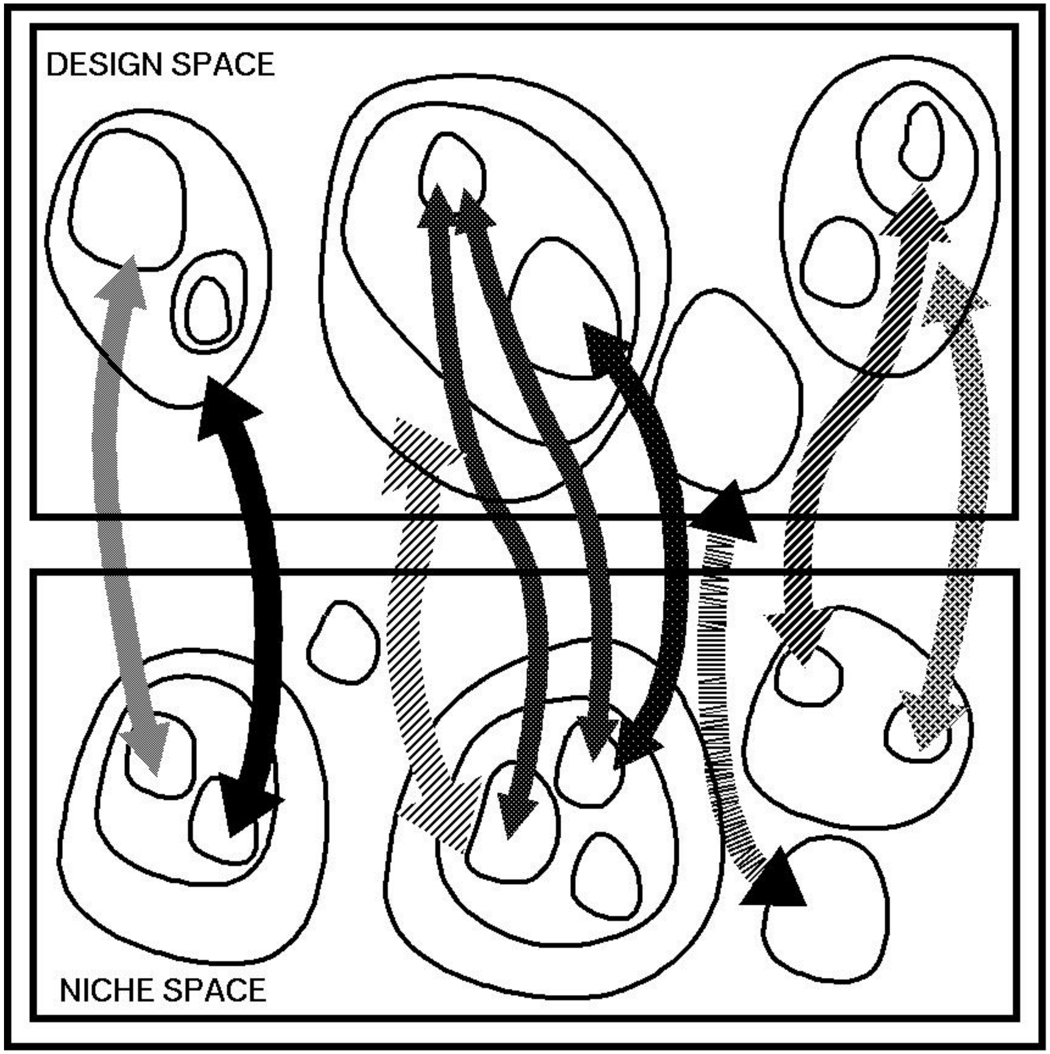
Biological evolution has already "explored" many such issues – though not completely

It will be useful to identify and explain some of the limits in what evolution has already done.

Two spaces: designs and niches

A niche is a set of possible requirements.

A design is a specification for a class of solutions.



Some solutions are meta-solutions: their instances grow solutions, driven partly by conditions at “run-time”.

Too many people (in various disciplines) think only of dichotomies and continuous spaces.

We need spaces with different levels of abstraction allowing both discontinuities and some continuous regions and trajectories.

Tradeoffs-between evolution and development

Biologists distinguish

Precocial species: individuals born physiologically advanced and competent

e.g. chicks follow hen, and peck for food; new born deer run with the herd

Altricial species: individuals born under-developed and incompetent – very dependent on parents.

Jackie Chappell convinced me in 2005 that:

It's not species but competences that should be distinguished.

Even humans have some "precocial" competences (e.g. sucking is highly developed very early).

We argued in a paper at IJCAI 2005 that individuals need a "spectrum" of competences:

Some largely genetically determined, others developed through interactions between products of genome and environment.

A similar spread will be required for advanced robots of the future.

`http://www.cs.bham.ac.uk/research/cogaff/05.html#200502`

A later joint paper (invited for a special issue of IJUC, 2007) enlarged the theory and introduced new terminology

There are preconfigured vs meta-configured competences

The same distinction can be applied to meta-competences and meta-meta-competences.

The idea is that an architecture is grown in stages, or layers, where new layers learn by observing not only the environment but also contents of prior layers, and their interactions with the environment.

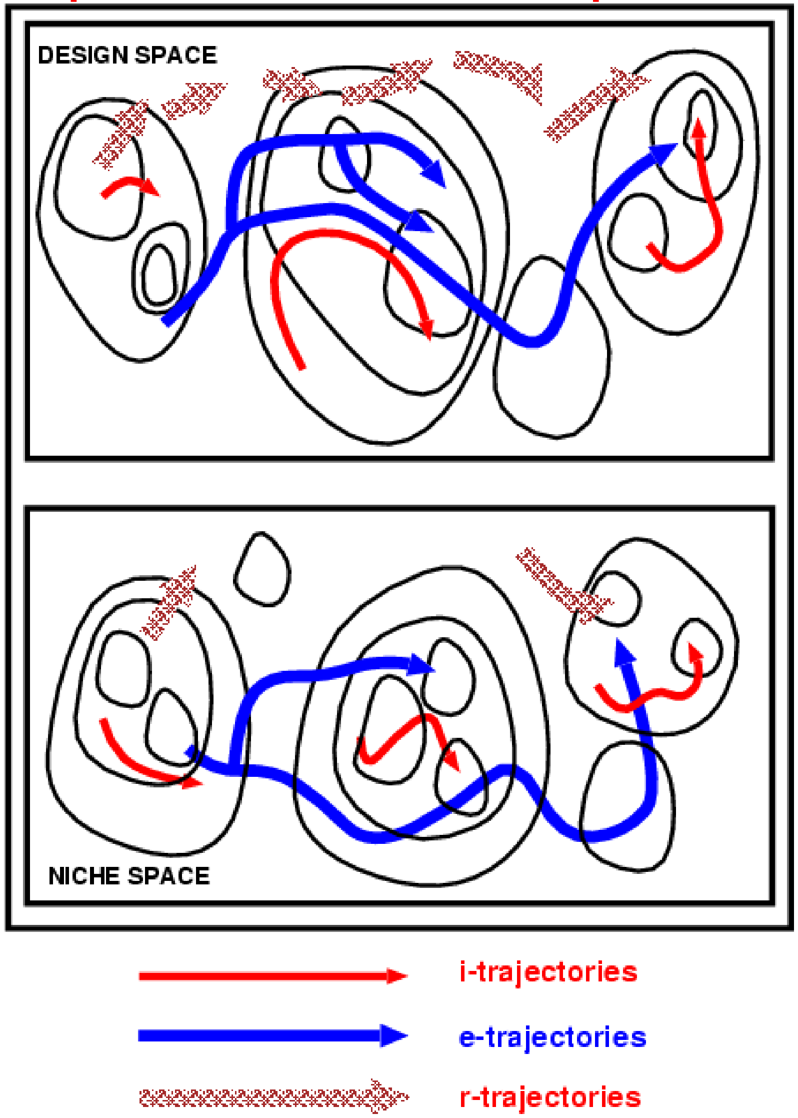
Staggered brain development is a prerequisite for this.

It is very likely that humans are not unique in this. (We are looking at some other species.)

Interacting trajectories in design space and niche space

Various interacting trajectories, with complex feedback links, are possible within design space and niche space and across both: dynamics of biological virtual machines in an ecosystem.

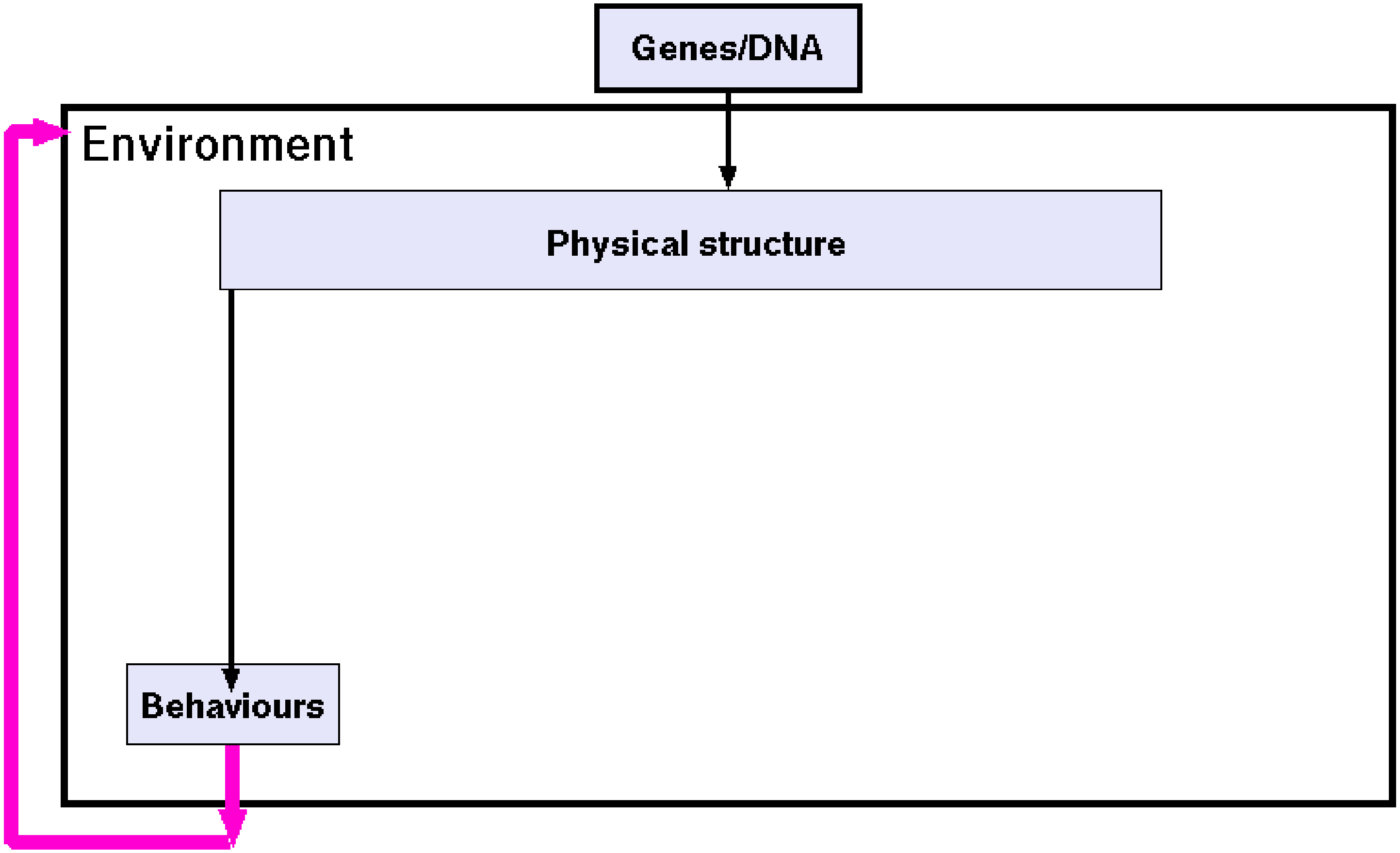
- **i-trajectories:** individuals develop and learn
  (altricial – precocial spectrum)
- **e-trajectories:** species evolve across generations
- **r-trajectories:** a 'repairer' takes things apart and alters them
  (hence discontinuities – e.g. heart transplants).
- **s-trajectories:** societies and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



Changes in designs cause changes in niches and vice versa: modelling all this may require new kinds of (mostly non-numerical) mathematics.

Individual trajectories

Routes from genome to behaviour

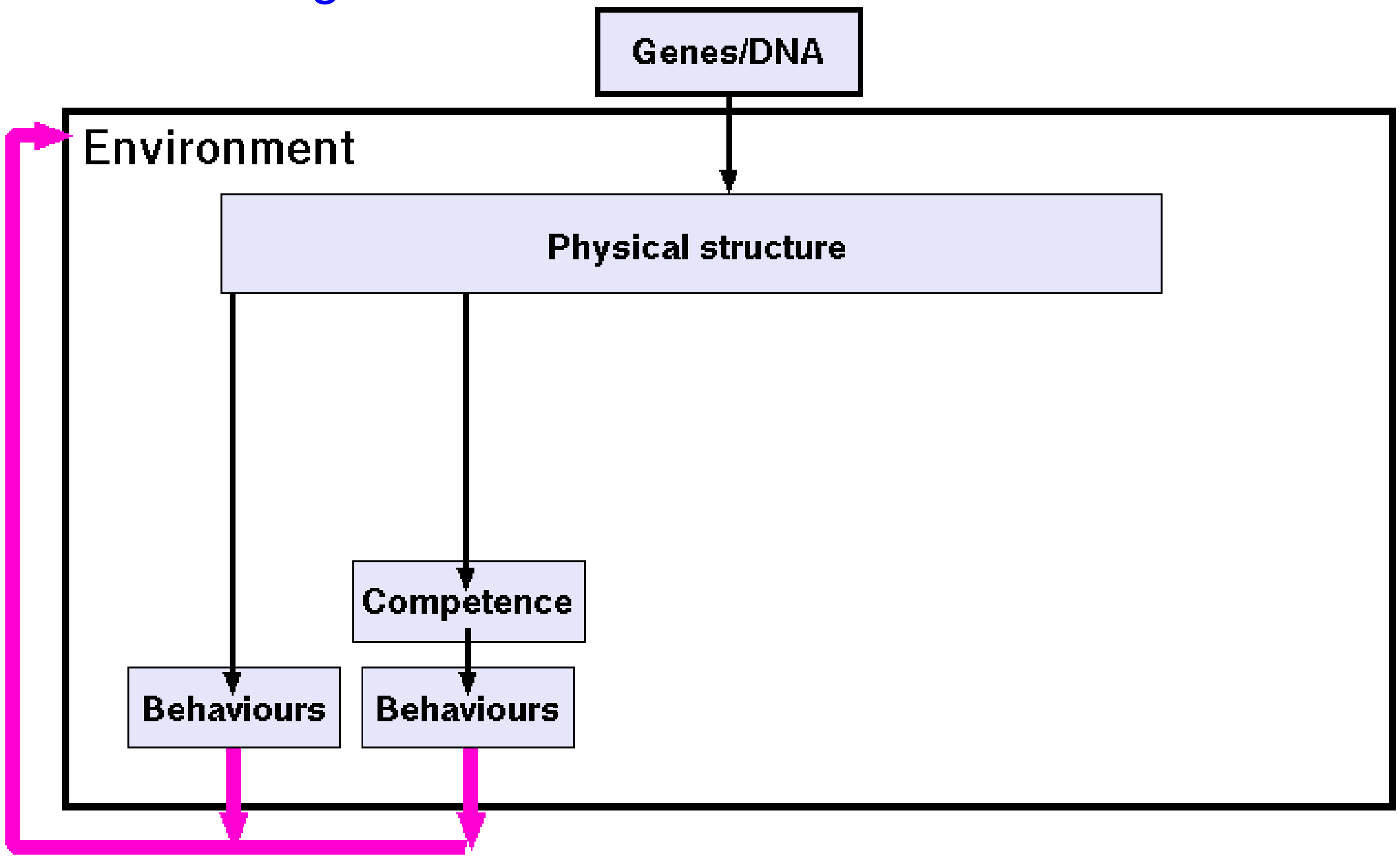


The vast majority of organisms (including micro-organisms) are like this.

Many don't live long enough to learn much.

Individual trajectories

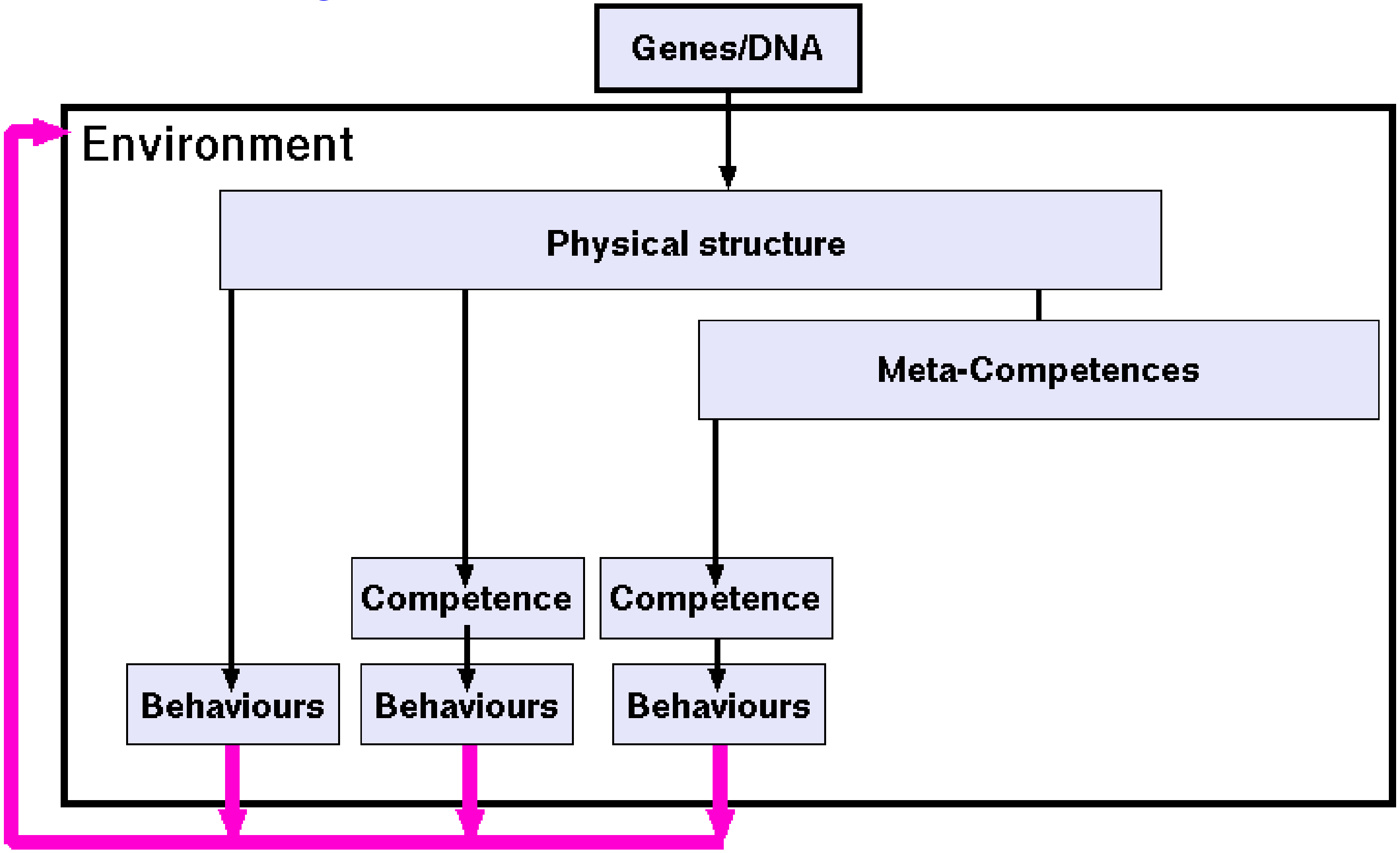
Routes from genome to behaviour



Some organisms instead of having only fixed behaviours triggered by events, have competences: packages of behaviours with conditional tests, or context-supplied parameters, that allow variations in the environment to be accommodated.
(There are nearly always intermediate cases in Biology!)

Individual trajectories

Routes from genome to behaviour

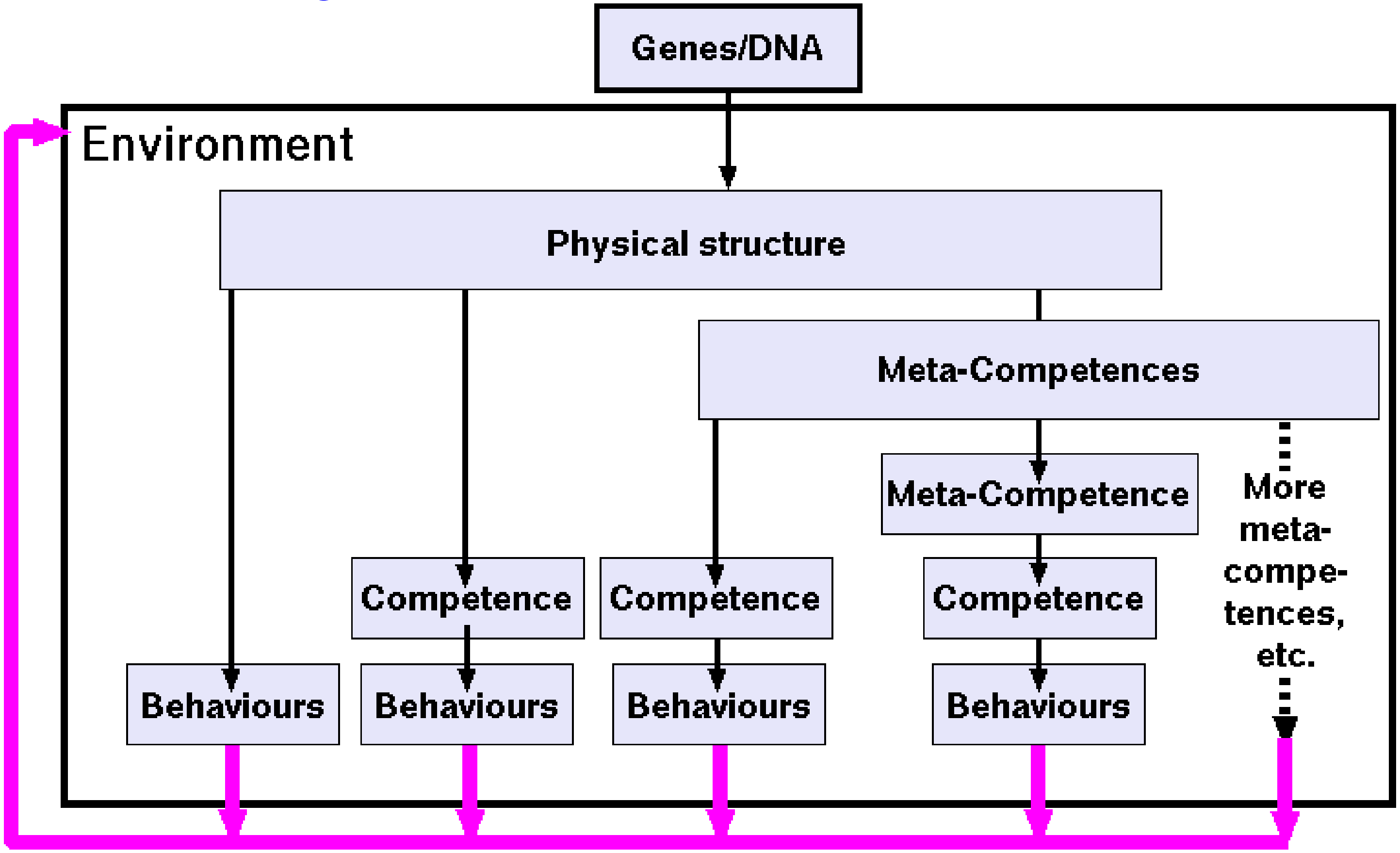


Competences may be meta-configured, i.e. developed by a learning process.

The learning processes may use a lot of genetically provided information about the environment, including the sorts of things that can be learnt and the kinds of things worth doing in order to learn them.

Individual trajectories

Routes from genome to behaviour



Meta-competences may also be meta-configured, i.e. developed by a learning process, using a meta-meta-competence, which itself may have been preconfigured or grown.

Humans go on learning how to learn through and beyond university.

The last few slides

The ideas in the last few slides were developed with Jackie Chappell, and reported in

Jackie Chappell and Aaron Sloman, 2007,
Natural and artificial meta-configured altricial information-processing systems,
International Journal of Unconventional Computing, 3, 3, pp. 211–239,
`http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0609`,

Chris Miall suggested the diagram structure.

All the competences and the specific contents of percepts, goals, plans, action-specifications, hypotheses need to be expressed in some form(s) of representation, produced by biological evolution, supporting structural variability, compositional semantics, and manipulability.

Such forms of representation, used internally (i.e. not for communication) must be available to pre-verbal children and many other species. See

`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#glang`
Evolution of minds and languages. What evolved first and develops first in children: Languages for communicating, or languages for thinking (Generalised Languages: GLs)

There is much more to be done: turning these ideas into designs for working systems, and in the process bugs in the ideas will surely show up and the ideas will change.

Compare Lakatos on how scientific research programmes develop.
Some turn out to be progressive and some degenerative:
it is often impossible to decide in advance which will be progressive.

One way in which meta-competences are extended

A feature of human learning/development that has been noticed in connection with language development, but seems to be more general is

moving from a large collection of empirically learnt examples, to a "framework theory" unifying the information in something like a deductive (not necessarily logical) system that allows new cases to be derived and old cases to be explained.

This allows the individual to deal with novel situations with complete certainty instead of having to do empirical research – e.g. being sure that a long thin rigid object of unfamiliar shape can be used as a lever to transfer a force.

The well known example of this is moving from example-based and pattern-based use and understanding of language to use of a generative syntax.

This seems to be a specialisation of the more general ability to base a new unified system of knowledge on a prior fragmentary collection of examples.

A difference is that in human languages there are many exceptions to the rules/laws so the architecture has to be extended to cope with them – in production or understanding, a non-trivial extension. (Hence "U-shaped" language learning.)

The ability to acquire knowledge by "working things out" tends to be ignored in studies of learning in children and other animals, in contrast with:

- Learning empirically
- Learning by being taught
- Learning by imitation
- Having innate knowledge.
  (See `http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#toddler`

We need a better understanding of the problems evolution solved – in order to investigate possible solutions

Alas, it's hard: many problems are invisible.

So people focus on their favourite problem(s).

Then wonder why their machines seem to be so far behind what they are aiming for.

Sometimes part of the problem is assuming that their favourite forms of representation, or favourite programming paradigms are suitable for developing explanatory models.

I've been to many seminars and conferences where speakers have just assumed

- that a robot's visual system has to solve the problem of computing distances using some familiar unit of measurement,
- that locations and orientations of objects need to be represented in some global coordinate system using some familiar units of length and angle,
- that objects have to be recognized and labelled as being in some category in order to be seen
- that only what exists can be seen – as opposed to perception of what is possible but not happening, and constraints on such possibilities.

Is it possible that initially animals have to start by building up networks of partial orderings, e.g. being able to represent "X is further/larger/heavier/more compressible/more viscous than Y", but without assuming that the relation forms a total order.

Vision is the hardest problem – to see and to solve.

Vision and mathematics

politics – art of the possible

mathematics – science of possible and impossible.

vision (especially perception) also about what's possible.

Too many people think vision is about recognition.

Compare J.J.Gibson: perception involves getting information about affordances.

Generalise this: information about structures, including fragments of surfaces, structural relationships and possibilities for change + constraints on change among the relationships: i.e. "proto-affordances".

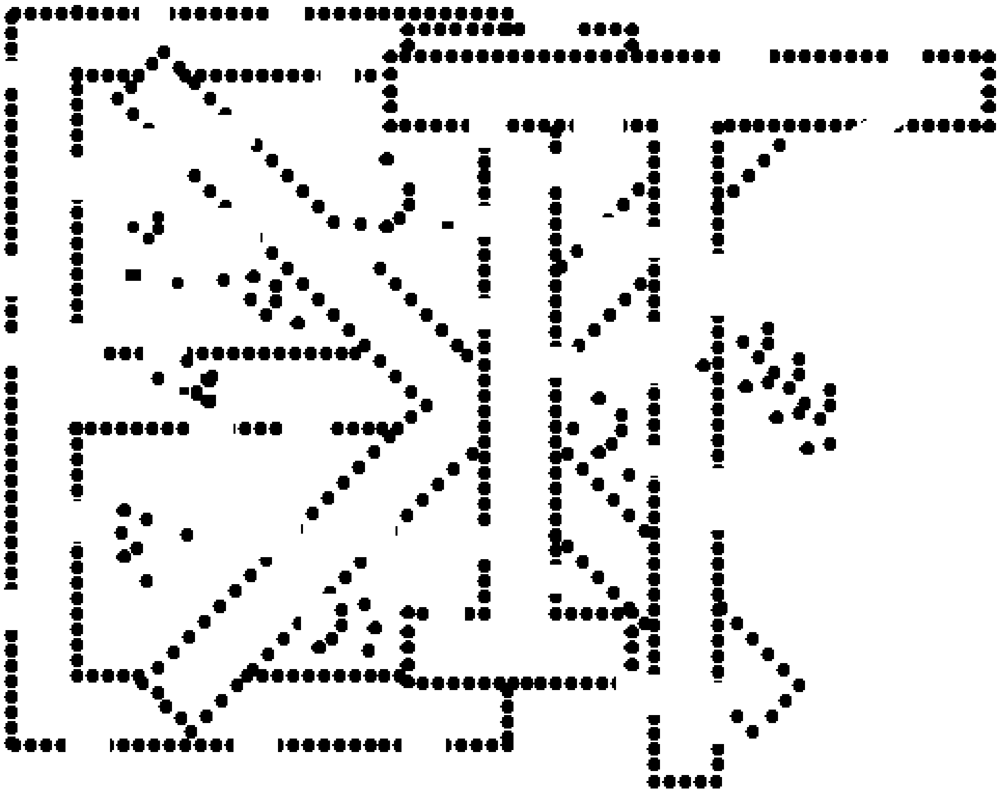
Despite the (very) low resolution and the noise you can see a great deal of structure in these images, and you can see many proto-affordances – things that could change, including available information.

What processes using two fingers of one hand could transform one of the configurations into the other?

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

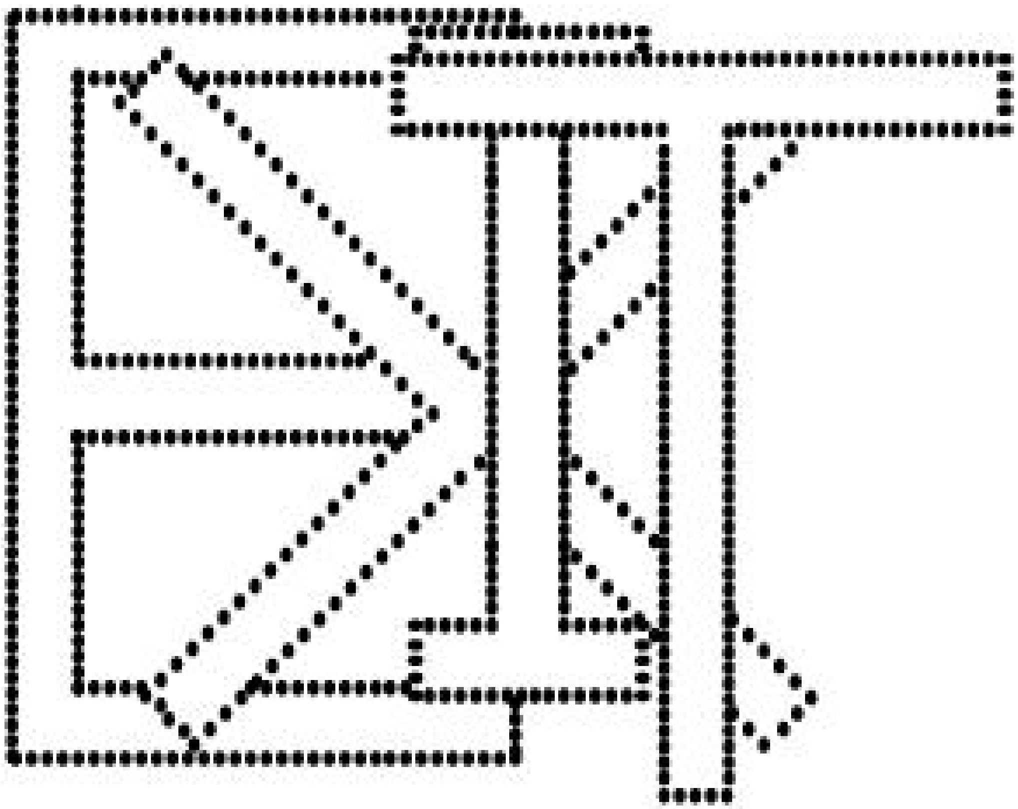
Work on Popeye around 1975: 1

The Popeye program could find several levels of structure in a picture like this:



The perception at different levels was done collaboratively and incrementally by concurrently active modules (simulated by interleaving)

Work on Popeye around 1975: 2



In case you did not see the structure in the previous image

Work on Popeye around 1975: 3

This shows the various layers of interpretation making use of several different ontologies, in different “domains”, as we called them, following Max Clowes.

Inferences (hypotheses) were propagated up and down the layers, across ontology boundaries.

(f) Words at the top level

(e) Letter sequences, where the letters were suggested by the stroke collections and by partly recognised words.

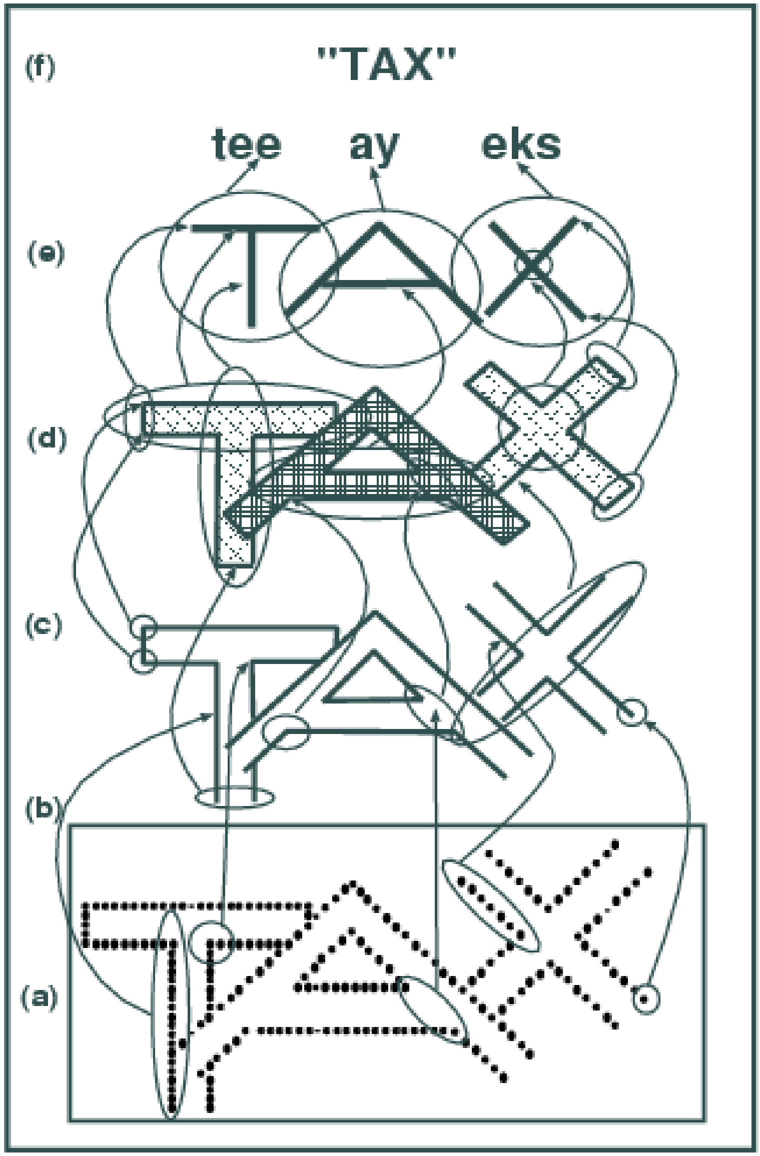
(d) collections of strokes (abstract linear entities)

(c) Collections of bars and bar-junctions made of opaque laminas, hallucinated onto the lines and line junctions

(b) Collections of lines and junctions hallucinated onto the dots

(a) Binary pictures (we called them ”dotty pictures”).
  Scanned initially looking for evidence of line-fragments grouped using the hough transform, suggesting which were good clues.

For more see Chapter 9 of *The Computer Revolution in Philosophy* (1978)



http://www.cs.bham.ac.uk/research/projects/cogaff/crp/chap9.html

Structures vs Processes and Possibilities

It took me many years to realise that

- for biological organisms, perception of static scenes and structures must be a special case of perception of processes, and
- just as perception of structures requires an understanding of relationships and constraints on relationships, perception of processes requires understanding of changing relationships and constraints on changing relationships.

This is deeply connected with the nature of mathematics: discovering what is and is not possible, and why some necessary connections are necessary, and not merely highly probable associations.

That goes beyond J.J.Gibson's understanding of affordances as merely empirical discoveries.

If the constraints are also understood they can be used to work out what is and is not possible in novel situations: as young children and many animals can do – but not yet robots.

Toddler Theorems – and Bugs

There are lots of “toddler theorems” discovered by very young children, which adults do not notice, because they have assuming them to be trivially obvious for many years.

Conjecture:

That ability later grows into the basis of mathematical competence.

Additional steps required include

- development of meta-cognitive extensions to the architecture (able to formulate observations and hypotheses about the modes of reasoning used);
- development of new communication competences allowing individuals to accelerate their learning of previously discovered results.

I am trying to use that as a basis for defending a (slightly modified) version of Kant’s philosophy of mathematics.

`http://www.cs.bham.ac.uk/research/projects/cogaff/10.html#1001`
If Learning Maths Requires a Teacher, Where did the First Teachers Come From?

The main modification, compared with Kant’s theory, is to allow that the mechanisms involved in mathematical discovery and proof are not infallible: bugs can occur, but can also be found, diagnosed and sometimes removed.

For that we have to combine ideas from Sussman’s Hacker, and Lakatos’ *Proofs and Refutations*

Sample of views on the nature of mathematical knowledge

- David Hume:
  All knowledge is either empirical or relations between ideas (analytic) or “sophistry and illusion”.
- J.S. Mill: Mathematical knowledge is empirical (that’s how much of it starts)
- Immanuel Kant:
  Mathematical knowledge is synthetic (non-analytic), non-empirical and necessary.
  But he allows that discoveries are triggered by experience.
- Logicism of Russell and Frege:
  All mathematical knowledge is just logical knowledge
  (Frege - arithmetic Yes, but geometry No)
- Bertrand Russell (1918): Mathematics may be defined as the subject in which we never know what we are talking about, nor whether what we are saying is true.
- Hilbert: Mathematical knowledge (of the infinite?) is about formal systems.
- Brouwer, Heyting, & others: intuitionists, finitists.
- Wittgenstein (RFM) (a form of conventionalism? Anthropologism?)
- Lakatos: Mathematics is Quasi-empirical (only half right?).
- Richard Feynman: the language nature speaks in. (Yes. Sort of – see below).

These are shallow and inadequate summaries, to illustrate the variety of philosophies of mathematics.

A new proposal: Robotics-based philosophy of mathematics

Kant (*Critique of Pure Reason*(1781)) claims that mathematical knowledge starts from experience of **how things happen to be** and then somehow becomes non-empirical knowledge about **what has to be.**

He did not put it as baldly as that.

I have been collecting examples of things learnt empirically by infants and toddlers (and future robots) that later change their epistemological status.

However, it is not yet clear what exactly changes, nor what forms of representation, mechanisms and architectures are involved in making this happen.

Providing a theory that shows how future human-like robots could go through such transitions will provide a new vindication for Immanuel Kant's view of mathematical knowledge as (a) synthetic, (b) necessarily true, (c) non-empirical (but NOT innate).

But things are more complex and varied than he realised, as I'll try to show.

I suspect that if Kant had known about AI, he would have used it, to explore and explain philosophical theories about how minds work.

This is the topic that first got me into AI.

What is empirical information?

Empirical information (not just beliefs), e.g.

- about specific situations,
- about general truths,
- about associations,
- about how to do something,
- about what will happen in certain circumstances,

can be **acquired** in various ways, e.g. by:

- observing something that happens, possibly as a result of performing some observation, introspection, measurement or manipulation (in some cases repeatedly),
- being informed by another individual, newspapers, a book, a dream, ...
- derivation from other information, using logic, mathematics, statistics, analogy, etc.
- guess-work, hunches
- from the genome (Genetic inheritance is the main source of general information in most species.)

What makes many kinds of information empirical (as Popper noted) is not **the source**, but **the possibility of observations and experiments contradicting the information**:

If a perceived counter-example could turn up, a prediction could fail, a situation could be observed to have been previously misperceived, or a method could sometimes not work, etc. (as often happens in science) the information must then be empirical. (Finding exactly which portions are empirical may be tricky.)

I. Lakatos in *Proofs and Refutations*: showed that mathematics can be “quasi empirical”.
The counter-examples in mathematics can occur in thought, not just perceived phenomena.

Theories about what is **possible** (e.g. formation of neutrinos) can only be **confirmed** empirically, not **refuted**. (This needs more explanation: See Chapter 2 of *The Computer Revolution in Philosophy*)

What came first: mathematics teachers or pupils?

It is normally assumed that mathematical knowledge has to be acquired with the help of adults who already know mathematics.

If that were true, mathematical knowledge could never have developed on earth, for there were no mathematics teachers in our early evolution.

These slides develop (using a spiral of theory, experiments, challenges and evidence), a conjecture about how mathematical competences originally evolved under pressure from biological needs.

Such pressures led to mechanisms, forms of representation and architectures that supported two very different forms of knowledge acquisition

- **Empirical learning**, including use of statistical information, correlations, etc.
- **Working things out**, including designing novel structures and mechanisms, and also making predictions, forming explanations, using not statistical but structural information – e.g. geometry, topology, properties of matter... etc.

Later on, mathematical knowledge became formalised, ritualised, and transformed into an object of study for its own sake, and culturally transmitted: through mathematics teaching.

But that teaching works only because the mechanisms produced by evolution are there in every normal child, though there may be differences leading to different speeds of learning and different preferred directions of learning.

Some informal experiments

The next few slides present a domain for you to think about, consisting of the configurations and transformations possible with an elastic band and a collection of pins.

This is one of many relatively small, restricted, domains in which one can interact with, play with, explore and learn from a subset of the environment.

Very young humans, and some other animals, over many months, interact with, play with, explore and learn from very much richer, more varied, more complex environments.

That process of conceptual, ontological, and "scientific" development is, I believe, hardly driven at all by biological physical needs for food, drink, warmth, physical comfort, etc. – though they have their role in other processes.

I have also deliberately abstracted from the role of social interaction during play since it is clear that in some species a great deal of exploration and learning can happen in periods of play **without** social interaction.

(I also think the role of imitation has been over-rated: you cannot imitate something done by another if you do not have the prior ability to do it yourself. If you are not so equipped, no amount of demonstration by another will teach you – e.g. to play a violin, place a puzzle piece in its recess, or to talk!)

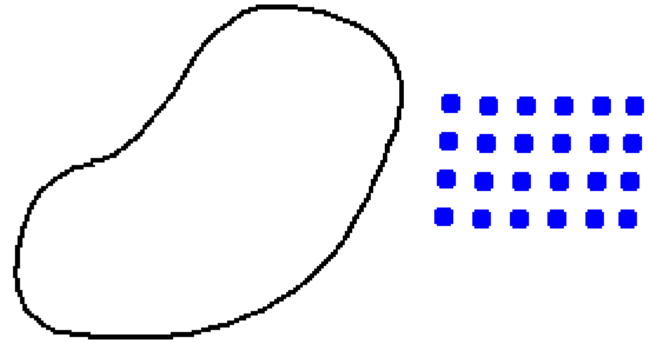
I hope that your experience as an adult, playing in an accelerated and simplified way, with a partly familiar and partly unfamiliar domain of structures and processes involving pins and rubber bands, will help you to appreciate some of what happens during infancy and childhood – including some of the long forgotten things that happened during yours.

Try to be a child 1

A child can be given one or more rubber bands and a pile of pins, and asked to use the pins to hold the band in place to form a particular shape

Ideas for many games and explorations for children, using string, elastic, pins, stones, pencil, paper, etc. can be found in this excellent little book (Penguin Education series):

Jean Sauvy and Simonne Suavy,
The Child's Discovery of Space: From hopscotch to mazes – an introduction to intuitive topology, 1974



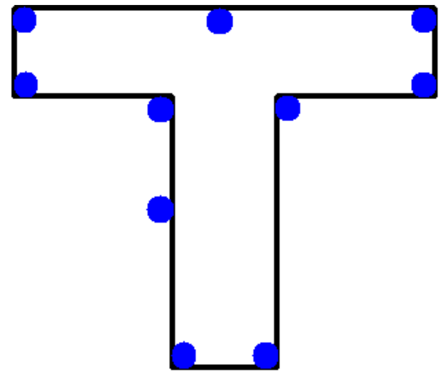
- You can easily work out how many pins you need to form the rubber band into **a square** (The pins need not go through the elastic! Only sideways pressure is needed.)
- How many pins will you need to make **a six pointed star**?

  You can probably work that out by reasoning about stars, possibly imagining or drawing one.
  A young child might find an empirical answer by exploring ways of pinning the rubber band into shape.
  A younger child may be able only to discover the shape by chance or by copying someone else.
- How many pins will you need to form **an outline capital "T"**?
  - What's the minimum number required?
  - Are you sure? (A child may only be able to report results of actual trials.)
  - How can you be sure that you have found the minimum number required?
    (Assume all the corners of the outline "T" are right angles (90^o), ruling out a "T" made from three long thin triangles.)
  - Does the minimum number required change if the "T" is upside down? How do you know?

Try to be a child 2

Stretching a rubber band to form an outline capital “T” is easy.

Are any of the pins shown forming the “T” on the right redundant?
How do you know? Why don’t you have to try removing them to see what difference that would make? (Like a very young child?)
Can you see another way to make a capital “T” – using only five pins? How?

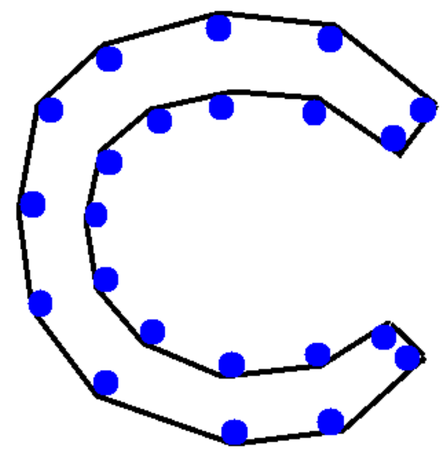


A capital “C” is possible, but not so easy.

How many pins would you need to make the outline of the “C” completely smooth: is it possible?

A child thinking about that might discover deep facts about continuity and curvature.

What’s the minimum number of pins required to form the “C”?
Is that a well-defined question?



Which of the outline capital letters can, and which cannot, be formed using pins and a single rubber band?

In particular, which of these: “A”, “B”, “D”, “E”, “F”, “G”, “H”, “O”,

A young child might have to do a lot of experimenting in order to answer these questions.

You can probably answer them simply by thinking about them, visualising various possibilities. What goes on in a brain, or mind, doing that?

How did you acquire such competences, and what forms of representation, mechanisms, and architectures make those competences possible? As far as I know, nobody has answers.

Rubber band (toddler?) theorems

A child playing with shapes that can be made using a rubber band and pins (or pegs on a peg-board – a more constraining environment)

- may discover, by observing what happens, many ways of rearranging the configurations, and
- explore consequences of various combinations of arrangements and rearrangements
- and then make some new discoveries about the effects of those sequences.

For example, things to be learnt could include:

- There is an area inside the band and an area outside the band
- The possible effects of moving a pin that is inside the band towards or further away from other pins inside the band. (The effects can depend on whether the band is already stretched.)
- The possible effects of moving a pin that is outside the band towards or further away from other pins inside the band.
- The possible effects of adding a new pin, inside or outside the band, with or without pushing the band sideways with the pin first.
- The possible effects of removing a pin, from a position inside or outside the band.
- Patterns of motion/change that can occur and how they affect local and global shape (e.g. introducing a concavity or convexity, introducing or removing symmetry, increasing or decreasing the area enclosed).
- It's possible to cause the band to cross over itself. **(NB: Is an odd number of crossings possible?)**
- How adding a second, or third band can enrich the space of structures, processes and effects of processes.

You can play with a virtual rubber band here, `http://nlvm.usu.edu/en/nav/frames_asid_172_g_2_t_3.html`
and here but only making convex shapes: `http://nrich.maths.org/public/viewer.php?obj_id=4901`

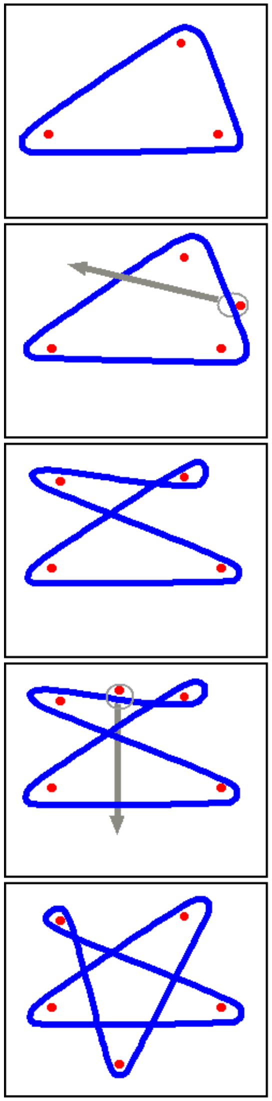
Producing, detecting, fixing bugs in mathematical proofs

Definition: a star is a closed polygonal figure with alternating convex and concave corners.

Is the following a theorem?

> The minimum number of pins required to hold a single rubber band in the shape of a star is six?

Zeyn Saigol produced the counter-example on the right:



Is it a counter-example?

Compare Imre Lakatos: Proofs and Refutations

What about making a star (as defined above) with exactly four pins?

It's not just rubber bands

The questions presented above about what can be learnt by playing with rubber bands and pins are not unique.

Similar questions can be asked and answered as a result of playing with and exploring

- many materials of different kinds

  (including string, wood, paper, water, sand, cloth, crayons....),

- structures involving those materials

  (including marks on surfaces),

  and

- processes in which structures and relationships change.

As adults, we can think explicitly about many of these processes, though we do not have access to the mechanisms and representations we use in doing that (though some people may think they do).

There are deep continuities and deep differences between adult processes and infant and toddler processes, and unanswered questions about the nature of the progression to adult competences.

Theories of kinds of stuff, bits of stuff, bits of process

Experiments with pins and rubber bands involve interactions between bits of stuff of different kinds – the interactions are bits of process.

The bits of stuff include:

- portions of the elastic band – all flexible and stretchable
- rigid, pointed, graspable pins, with a pointed end and a blunt end, and other portions
- the surface on which the band and pins lie, and into which the pins can be pushed
- the materials composing the experimenter's hands and fingers.

The bits of process involve changes described in previous slides, including both local changes (e.g. pin pushes piece of rubber sideways), and more global changes

(e.g. a straight line becomes bent, or *vice versa*, the rubber becomes more taught and more resistant to further pushing, an asymmetric shape becomes symmetric, or *vice versa*).

Someone playing with pins and bands could acquire a moderately complex, but finite, "micro-theory" about many local structures and processes that can occur.

The theory can generate predictions about results of indefinitely many (but not all) different combinations of actions involving the rubber band and pins

- The predictions will use topological and geometrical reasoning, not just observed correlations.
- This is a form of intuitive child science, that might later be fully explained by adult official science (physics, materials science, plus mathematics).
- Compare the work by Pat Hayes on "The naive physics manifesto".
- The theory will not predict exactly when the band will break if stretched a long way. (Why not?)

Alternatives to empirical knowledge

Empirical beliefs normally cannot be proved to be correct.

Not even in science, contrary to popular misconceptions
See `http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#talk18`
"What is science?"

Contrast empirical information with:

- Necessary truths – some derivable using only logical or mathematical methods.

  Examples – nothing could make these false:

  P or (not P), **not(P and (not P))**, **Containment is transitive**, $3 + 5 = 8$

  If P is true, and (P logically implies Q) is true, then Q is true.

  If Q is false, and (P logically implies Q) is true, then P is false.

- Intermediate case (Semi-necessary/Relatively-necessary truths):

  Truths logically/mathematically derivable from the best available deep, general, well-established theory about how some important part of the world works. E.g.

  Any portion of pure water contains twice as many hydrogen atoms as oxygen atoms.
  Electric currents flowing through a metal produce heat.
  A rubber-band star needs an even number of pins. (Or is that necessary?)

- "Framework Theories" (in one sense of the term)

  Unavoidable/indispensable (implicit) theories required for perceiving, thinking, learning, acting, etc.

  Explaining precisely what such a theory is would require us to build something like a working robot using one: too hard, at present!

  So instead I'll merely offer sketchy examples, below.

Varieties of semi-necessary knowledge

Semi-necessary information includes deep, general, well-tested theories about how some important parts of the world work.

It also includes whatever is derivable from the theories, using derivations that depend on the form in which the information is encoded.

There are different types of semi-necessary knowledge, including the following:

- Official-adult science

  Whatever is in, or is derivable from, the best established general theories about the nature of space, time, and the things that can occupy space and time.

  That would include modern physics, chemistry, and biology, for example. (What else?)

  This is the most explicitly formalised type of theory, and many of the derivations use logical and mathematical forms of inference.

- Unofficial-adult science

  This includes a vast amount of information about kinds of things that can exist, kinds of process that can occur, and constraints on what can occur. (I.e. a rich ontology, much of it unarticulated.)

  There is a core **shared** among all adults, and also much that **varies** according to culture, professions, and individual interests and histories: e.g. rubber-band science must be relatively new.

  The contents of this sort of science are not usually articulated explicitly: much is represented internally implicitly in a usable form, and often used, though rarely communicated – e.g. craft knowledge.

  Everything derivable (how?) from the contents of this science is also part of this unofficial science.

- Baby-toddler science

  This includes precursors of adult-unofficial science, developed in early life – like rubber-band science.

Baby-toddler semi-necessary knowledge

Official-adult science has received most academic attention.
Baby-toddler science has only been investigated in narrow fields, in a shallow way, depending on changing fashions in psychology.
Unofficial adult science is somewhere in between (some of it studied in anthropology and ethnography).

A deep study of Baby-toddler science requires investigation of environment-driven requirements for the design of **working** animals, potentially testable in robots that develop as many animals do, e.g. through play and exploration in the environment.

Baby-toddler science includes deep and widely applicable theories developed in infancy and childhood that make sense of the environment, including:

- theories about kinds of stuff that can exist

  e.g. various kinds of solids, fluids, sticky stuff, etc., with various kinds of flexibility, elasticity, etc. Some of this will be mentioned later.

- theories about kinds of relationship that can hold between bits of stuff of various kinds

  e.g. relationships between rigid fixed pins and elastic bands in various shapes.

- theories about kinds of process (changing properties and relationships) that can occur – e.g. what happens if a pin is removed in the capital "T"? What can happen if you pull a piece of string?

An incomplete presentation on the ontology developed in baby-toddler science is here:

`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#babystuff`
DRAFT: Assembling bits of stuff and bits of process, in a baby robot's world: A Kantian approach to robotics and developmental psychology. (Help needed extending this.)

Ways to think about evolution and development.

The CogAff Schema for Architectures

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

A crude taxonomy of sub-functions of possible architectures

divided (vertically) between (overlapping)
- sensory/perceptual functions
- more general (and more central) functions
- action/effector functions,
  (Compare Nilsson’s “three towers”.)

and divided (horizontally) between
- evolutionarily oldest
  (completely reactive – e.g. microbe-like or insect-like)
- more flexible deliberative functionality
  (representing, reasoning about merely possible or unperceivable situations and processes)
- meta-semantic functionality
  (able to represent, reason about, and possibly control things that represent, including self and others).

Architectures can differ according to which sorts of sub-functions they have (in the various boxes), how they are connected, whether the components are all fixed, or grown during development.

The latter is ruled out by some definitions of “architecture” as unchangeable.

A special subset of the CogAff Schema

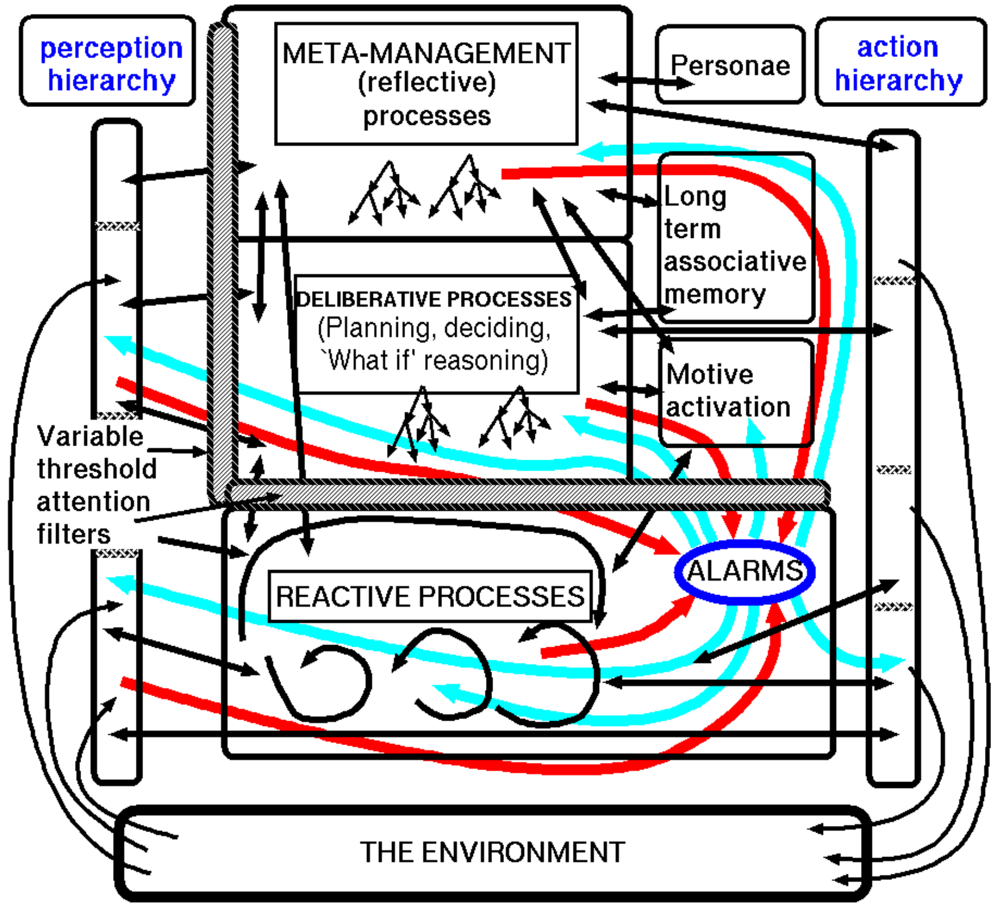
The conjectured H-Cogaff (Human-Cogaff) architecture

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

A special subset of the CogAff schema, containing many kinds of concurrently active mutually interacting components.

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

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



Compare Minsky's **Emotion Machine** architecture – with different subdivisions highlighted.

Embodiment is important but not in the way most embodiment theorists think.

The physical environment is a deeper driver of the evolution and development of animal intelligence than most people have noticed

Evolution seems to have produced some overlapping cognitive functions in animals with very different morphologies, e.g. primates, elephants, corvids, octopuses: convergent evolution of cognition.

The deep explanatory similarities in these competences will be closely related to features of a complex 3-D environment with various mixtures of persistence and change, including possibilities, opportunities, threats, dangers, routes, obstructions, food, usable objects, etc. (proto-affordances and affordances).

McCarthy:

"Instead of building babies as Cartesian philosophers taking nothing but their sensations for granted, evolution produced babies with innate prejudices that correspond to facts about the world and babies' positions in it. Learning starts from these prejudices. What is the world like, and what are these instinctive prejudices?" (op. cit.)

There will be some species differences, and possibly also differences within a species (e.g. humans).

Social drivers came much later, building on the results

Human language has evolutionary and developmental precursors

CONJECTURE: Both pre-verbal humans and also many non-human species make use of internal forms of representation, used for:

- perception
- reasoning, planning
- goal formation
- motive selection
- question formation
- plan execution

Such forms of representation will need to provide features of human languages

- variable complexity
- structural variability
- compositional semantics
- manipulability

But they need not be restricted to linear forms or discrete elements. Some will use analogical rather than Fregean forms of representation.(Compare human sign languages.)

We can generalise the notion of "language" (Generalised Language, GL) to include these forms of representation used for internal purposes, rather than for communication.

`http://www.cs.bham.ac.uk/research/cogaff/04.html#200407` (1971)
`http://www.cs.bham.ac.uk/research/projects/cogaff/81-95.html#43` (1979)
`http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#glang` (2008)

Why Is Progress in AI/Robotics Slow?

Usual (wrong or incomplete) answers

- Wrong answer 1: It's not really slow – look at all the new results

  Visual recognition, visual tracking, 3-D reconstruction, DARPA challenges, robot soccer, impressive robots, e.g. BigDog .... but these are all very limited

  e.g. doing but not thinking about doing:
  real-time control but not reflection on what was and wasn't done or why, or what could have been done.

- Wrong answer 2: Progress is slow because the problems are really really hard, and in order to solve them we need the following new approaches ...
  - Biologically inspired mechanisms (neural nets, evolutionary computation....)
  - Biologically inspired morphology for robots (emphasis on "embodiment")
  - New architectures
  - Development of quantum computers, or some other new form of computation
  - .....

These answers miss the main point.

An alternative answer - **Progress is slow because:**

- We have not understood the problems
- Most of the problems come from the nature of the environment

  Which has changed in the course of evolution:

  There's not much scope for intelligence if you are a micro-organism living in an amorphous chemical soup

- We need to understand the cognitive problems and opportunities provided by the environment.
  - These arise from the fact that the environment is structured in many ways – some of which, at least, can be discovered and used.
  - This requires something deeper than using statistics to learn correlations and probabilities (e.g. Bayesian learning)
  - Some species, though not all, can do this.
  - Humans go further, and discover what they have learnt, formalise it, and transmit it across generations.
  - And they use it to do increasingly complex things with little intellectual effort: support for productive laziness is a key feature of mathematical competences

My task for the next few hundred years: understanding requirements and possible designs.

There's lots more

E.g.

On the Cognition and Affect web site

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

In my talks directory

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

On the CoSy project web site

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

Messy overview of my doings:

`http://www.cs.bham.ac.uk/~axs/my-doings.html`