

Panel, Mon 4th Nov 2002

The architecture of an enduring, perceiving, learning, cognitive agent.

HOW TO THINK ABOUT ARCHITECTURES:

**Requirements, Designs,
Niche space, design space,
Trade-offs, Trajectories**

**Architecture-based conceptions of
emotions and other affective states**

Aaron Sloman

School of Computer Science,
The University of Birmingham, UK
<http://www.cs.bham.ac.uk/~axs/>

(Last changed November 13, 2002)

Why I don't like the life-long intelligent assistant project

- Designing a “grown up” (ready to use) intelligent agent can probably be done by producing a large collection of ad-hocery “tied together with string”. E.g.
 - there need be no deep integration between different modules that use an understanding of spatial structure and time, such as choosing an itinerary and designing an office layout.
 - There need be no understanding of what is in common between explaining why a political strategy won't work, why a proposed itinerary is not good, why a software design is poor, etc. (What are explanations for? How do they work?)
 - there need be no deep relation between understanding emotions in others and being anxious about a risky plan, and no deep understanding of why reassuring utterances can reduce anxiety, since a large enough collections of shallow rules will suffice.
 - By starting with a design for some sort of “complete” agent with a wide (but limited) range of diverse competences integrated as in a young child, we may be able to design a common platform that can be extended to support many different kinds of intelligent agents including the “personal assistant”.
 - What happens in humans is an existence proof that that is possible.
 - However, very little is understood about the architecture of a young child's mind, and what it is that makes various later developments possible.
 - One way to make progress is to aim for something more basic and general, and then see how to produce the more specific application.
- Of course both approaches could be pursued in parallel, as long as researchers talk to one another about what they have learnt.**

Alternative proposal - two interacting 3 to 5 year olds

In the final session of the workshop Raj Reddy asked me what sort of project I would propose if I had many millions of dollars to spend.

I gave him a short and unclear answer, referring to some ideas being developed partly in collaboration with Marvin Minsky and Push Singh.

A much longer answer can be found in a draft, still evolving, slide presentation here: <http://www.cs.bham.ac.uk/research/cogaff/manip/>

It outlines a project to design a couple of child-like robots that can do various things and teach one another to do things, or do things better.

Key hypothesis:

in humans by about the age of about five most of the "infrastructure" for a huge variety of types of subsequent learning has been developed.

- We don't know much about how that infrastructure works, though we know quite a lot about what it can and cannot do.**
- Aiming to replicate it (at some level of abstraction) in the design for a working robot could put us in a much better position to do all sorts of other things later on.**
- Think of all the different skills and knowledge humans pick up, at school, from reading, exploring, playing with toys, etc.**
- There are reasons for NOT trying to start from a neonate (or an embryo!) that boot-straps itself.**

Deliverables and milestones: stage 0

- **Analysis of problems**

I claim there are many problems we can *label* at a high level of abstraction, but whose detailed nature we do not understand.

Most people will say that perception is a problem, but they mischaracterise the problem.

E.g. they may think that perception is segmentation and recognition, whereas those are merely **fragments** of the task. Marr described the purpose of vision as providing information about **geometric and physical properties** of objects.

Perception of **affordances** is more important and more general. (See my white paper)

- **Identification of possible solution space**

People propose solutions based on their favourite architecture, or reasoner or representation formalism.

We need a generative taxonomy of possible solution types, and a generic mode of evaluating possible solutions against various classes of problems.

(We need to understand mappings between design-space and niche-space.)

- **Some solution-components:**

We need taxonomies of:

- Mechanisms
- Formalisms (**object and meta**)
- Architectures (**combining mechanisms using various formalisms**)
- Modes of reasoning (including reasoning linking requirements and designs)
- Example applications that could demonstrate use of the above

First draft analyses should be available within a year, and will be repeatedly refined thereafter, if the programme is well managed.

Why look for biological inspiration?

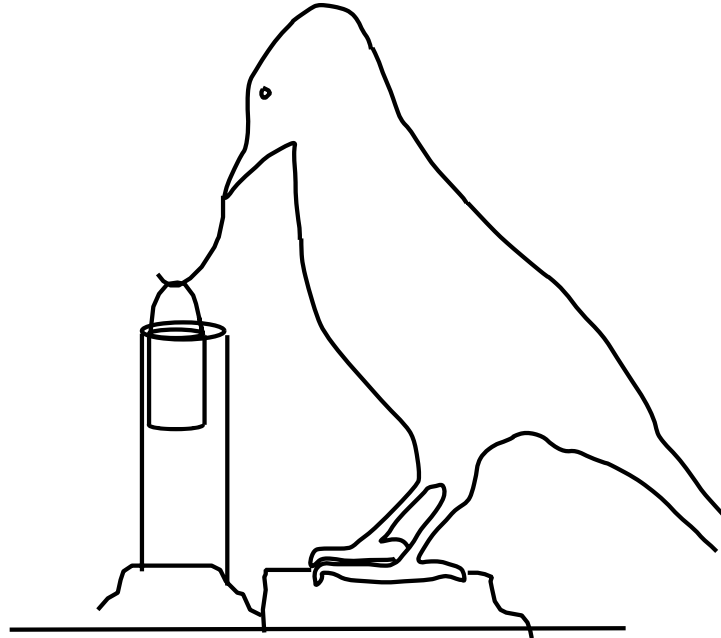
A partial answer

- One way to gain a deeper understanding of the problems (not necessarily the solutions) is to look at some of the capabilities of biological systems, including young children and many animals.
- We should examine those capabilities as engineers trying to replicate the systems. That will help us understand what the animals do. But it can be very difficult to find deep ways of characterising the tasks.
- Doing that may be irrelevant if we want to make machines do things no animal has ever been able to do: e.g. adding up large collections of numbers at lightning speed, or various kinds of data-mining.
- However if the kinds of tasks we want machines to perform are analogous to things adult humans can do and involve considerable generality, then there may be an 'unobvious infrastructure' that in humans is a product of millions of years of evolution and learning, and which we may not easily be able to reinvent from scratch.

An example follows:

Betty Crow: Cognitive Agent and Hook-maker

Two crows, Betty and Abel, learnt to use bent wire to fish a bucket of food out of the vertical tube (as in the picture). Then Abel flew off with the hook.



See the video here: <http://news.bbc.co.uk/1/hi/sci/tech/2178920.stm>

To find more, give google: **betty crow hook**

- Betty tried using a straight piece of wire for a while, and failed.
- She then pushed one end of the wire into the tape holding the tube and moved the other round with her beak, making a hook, which she used to lift the bucket.
- She did this 9 times out of 10. **Reported in Nature and shown on BBC TV (August 2002).**

HOW CAN WE FIND OUT WHAT BETTY WAS DOING?

What sort of architecture could do what Betty did?

Many explanations are compatible with **any** observed performance, e.g.:

- **Pure chance?**
- **An innate behaviour** triggered by some mixture of internal and external state?
 - What mixture?
 - How did the genes get the information? Why was it selected?
- **A learnt adaptation** in a trainable (altricial) reactive system?
 - What sort of boot-strapping could achieve this?
 - How is the learnt information acquired, represented, stored, activated, used?
- **Was it a deliberative** (e.g. problem-solving) process?
 - Using what sort of ontology for possible goals, states, actions?
 - Using what general knowledge?
 - Invoked how?
 - Acquired how? (Using an architecture built in infancy?)
 - Using what planning mechanisms? (Using what representations, what search mechanisms?)
- **Did it involve self-knowledge?** (Reflection/meta-management)
 - Did Betty understand what she was doing, or did she, like many AI deliberative systems, lack reflection/meta-management? (Can a crow teach another crow to do this?)

The questions are deep and important because understanding of spatio-temporal processes can be re-used in many contexts.

E.g. doing mathematics, designing architectures, thinking about anything complex.

Vision and affordances

Vision is not just about

- Object recognition
- Perception of geometrical and physical structure and motion
- Building cognitive maps for route-planning

There's something deeper, not yet properly characterised, which can be called **perception of affordances**.

Affordances are not “objective” properties intrinsic to physical configurations. They are **relational** features dependent on the perceiver's

- Common or likely goals and needs
- Capabilities for action (physical design + software)
- Constraints and preferences (avoid stress, injury)

Affordances in a complex scene can be construed as

- **sets of sets** of counterfactual conditionals,
- **spatially indexed**: different sets attached to different parts of objects.

How should affordances be perceived, represented, used, explained to others?

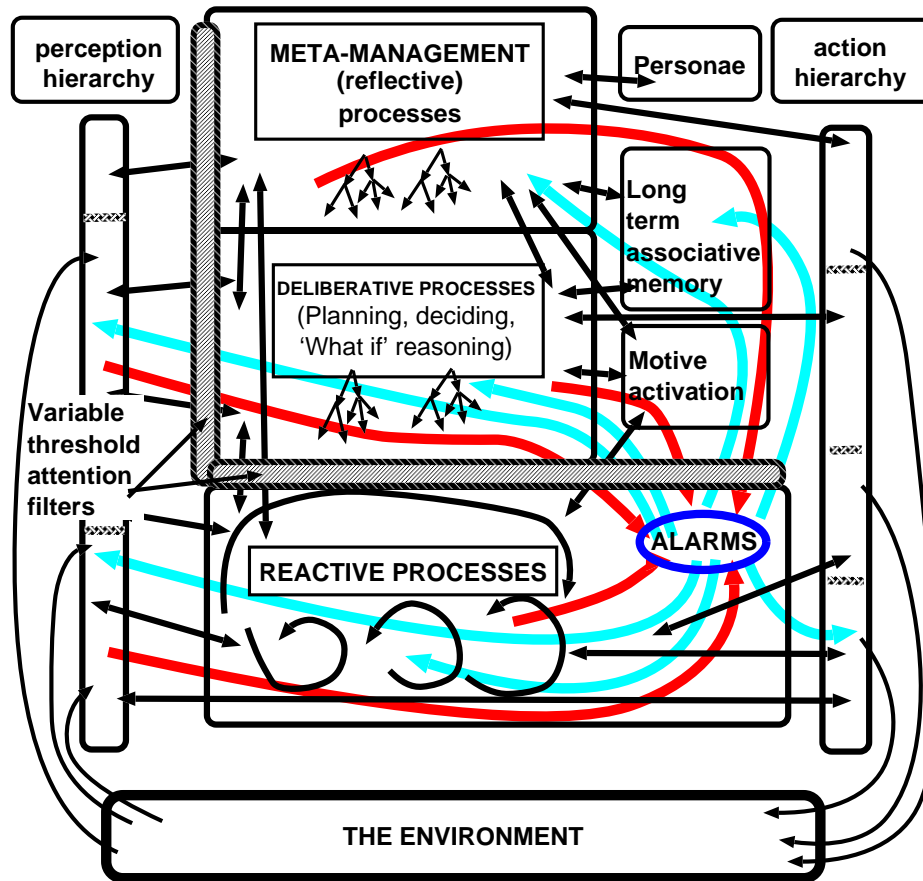
Different representations and mechanisms handle affordances in different architectural layers - e.g. skilled behaviour is mostly reactively controlled.

Probably not using modal logics???

A hypothetical Human-like architecture:

H-CogAff (See <http://www.cs.bham.ac.uk/research/cogaff/>)

This partly overlaps with Minsky's *Emotion machine* architecture.



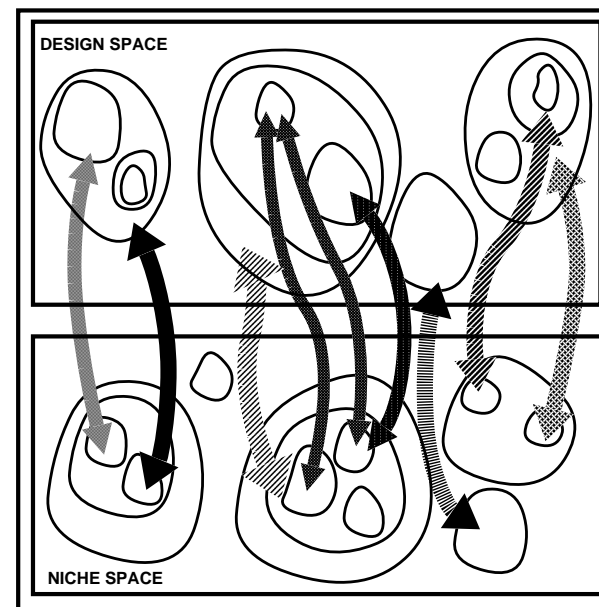
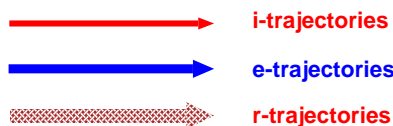
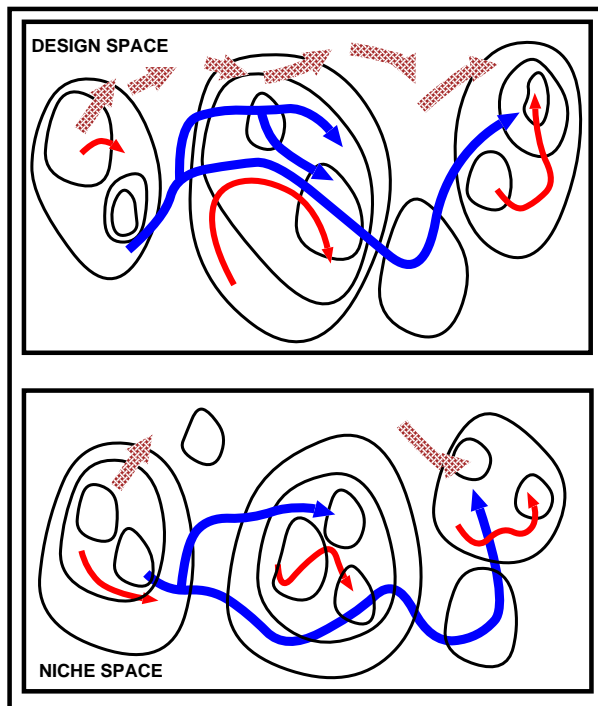
Where could it come from?

Various trajectories: evolutionary, developmental, adaptive, learning...

Two spaces – and trajectories

Mappings and trajectories in design space and niche space.

Don't expect fitness functions to characterise the relations between design space and niche space. Different fitness **relationships** link different regions of design space and niche space. (See the white paper, and Cogaff project papers, for more on this, including different kinds of trajectories: individual, evolutionary, repair trajectories.)



I-trajectories divide into various kinds of **development** and **learning**.

There are also

- **c-trajectories:**

E-trajectories where evolution is mediated by cognitive processes, e.g. mate selection.

- **s-trajectories:**

Trajectories of social systems.

One way to think about classes of architectures

The CogAff (draft) schema

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

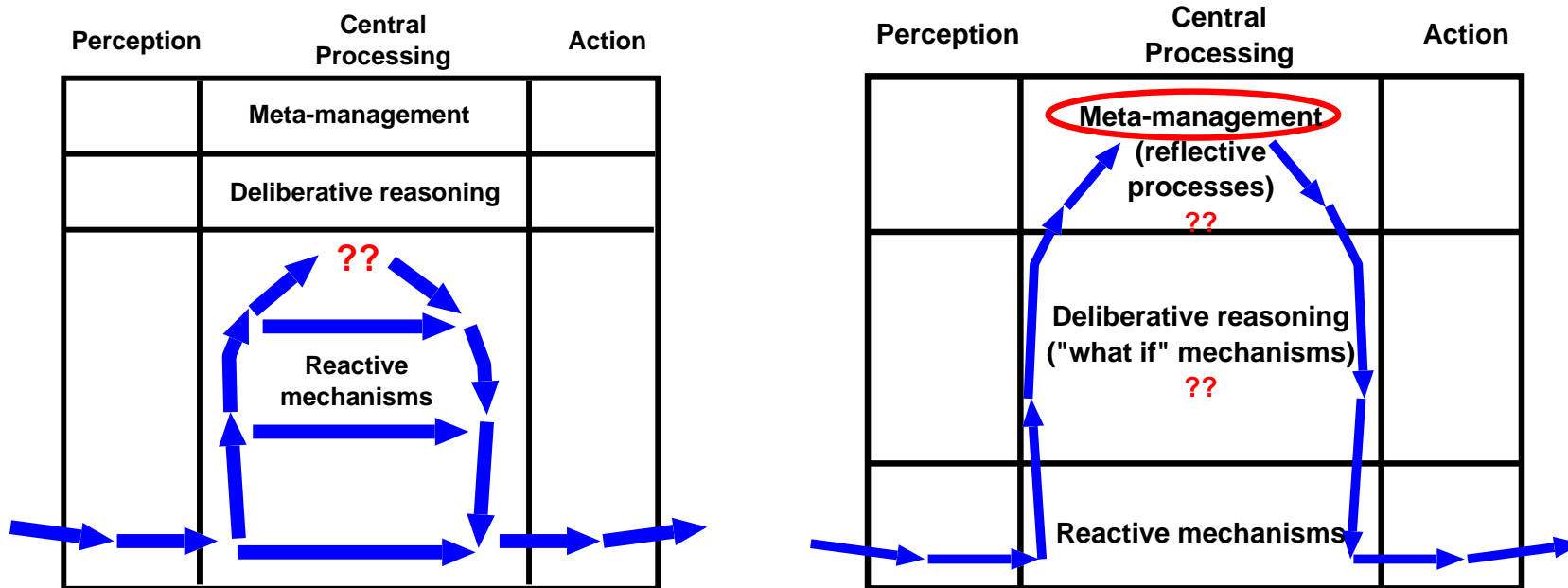
Perception and action both involve different concurrent levels of abstraction:
i.e. “multi-window” perception and action

In describing an architecture we can ask

- which boxes contain mechanisms,
- what forms of representation are used in the different boxes,
- what communication links exist between mechanisms or boxes,
- whether the mechanisms are all there initially or whether they grow after birth or hatching, etc. (“precocial” vs “altricial” designs.)

CogAff accommodates many architectures

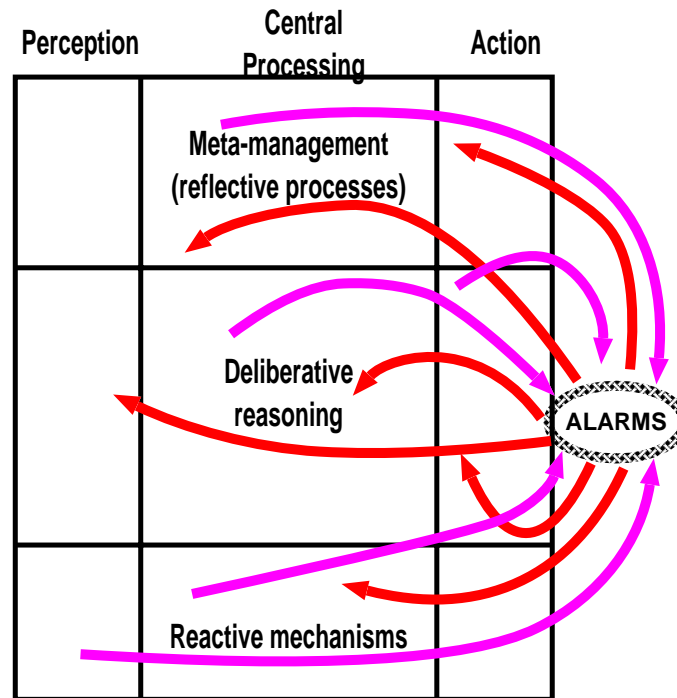
Reactive subsumption vs Contention-Scheduling



The second case is an Omega Ω architecture.
Both of these have peep-hole (not multi-window) perception and action

Many other architectures are accommodated by the CogAff framework.

Cogaff enhanced with the concept of “alarm”



The alarm mechanism should be drawn within the reactive box, but is shown separately for clarity.

There is a very general requirement for alarms. Why?

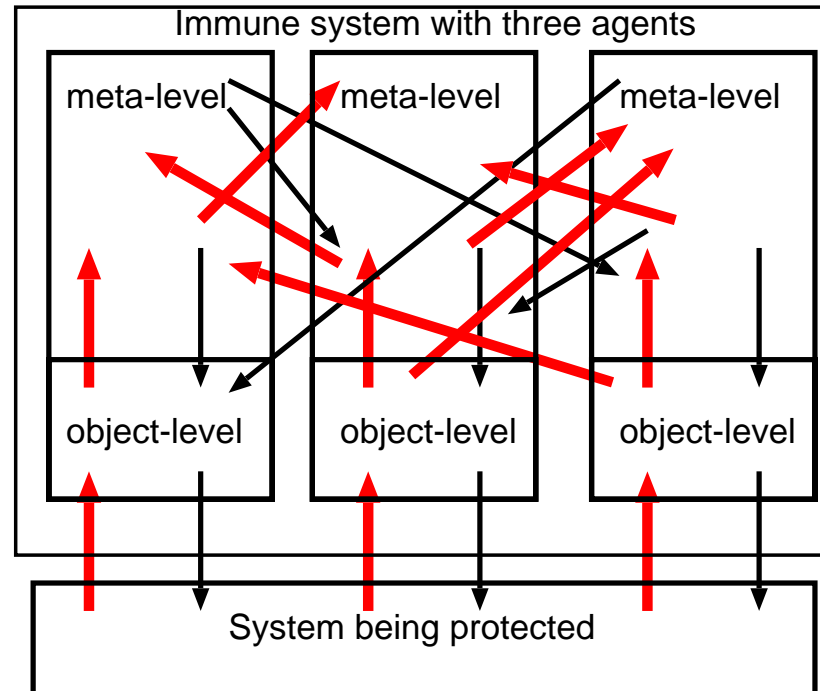
How many different sorts are there?



(Explaining varieties of emotions)

Catriona Kennedy's Mutual meta-management

This was an unexpected development of the CogAff schema.

Towards robust immune-system-like intruder-monitoring



 red thick upward arrows: sensing
 black thin downward arrows: acting
(Not all possible arrows shown)

See <http://www.cs.bham.ac.uk/~cmk>

Beware of published theories of emotions

- Most emotion theorists don't know how to explain states in terms of what's going on in an information-processing architecture.
- So they try to characterise emotions in terms of observable/measurable phenomena (blood pressure, galvanic skin response, weeping, smiling, posture, tone of voice, etc.)
- Instead we should develop an affective ontology based on varieties of control processes within an architecture, e.g. generation of motives, detection and resolution of conflicts of motives, priority changes, alarm mechanisms and interrupts of various sorts, switches of attention and modes of processing, etc.
- In various papers I've shown how at least three major categories of emotions are associated with the three types of architectural layers (primary, secondary, tertiary emotions), but that's a grossly over-simplified taxonomy.

A full theory would start from our common-sense taxonomy of desires, preferences, pleasures, pains, values, ideals, attitudes, concerns, interests, moods, emotions, intentions, etc. and then produce a richer, more precise, architecture-based taxonomy of affective control states.

Compare: H.A. Simon 'Motivational and emotional controls of cognition', 1967.

Some things we still have to do

Understanding most kinds of human competence is much harder than understanding Betty's, especially the very **general** collection of competences shared by all humans in all cultures from childhood (e.g. four or five years).

We need to develop

- A good **ontology** for describing contexts, percepts, goals, tasks, solutions, actions. (including: internal, external, communicative,etc.)?
- A way of using that ontology to specify **requirements** for a human-like (child-like) system.
- A good **formalism** (set of formalisms?) for specifying **requirements** or **desirable capabilities**?
- A good ontology for **design-components**, e.g. involving
 - Forms of representations
 - Types of mechanisms
 - Modes of composition
- Methods to **derive the types of match/mismatch** between a design and a set of requirements. (Compare formal verification).

We don't yet know whether we can design and build complete systems, or whether we'll have to evolve them, or whether we'll have to design altricial infants that bootstrap themselves.

See this draft project proposal <http://www.cs.bham.ac.uk/research/cogaff/manip/>

A note on tools

For exploring designs for complex agents we shall need powerful tools.

There should be a requirement for software tools to be platform-independent so that different researchers can share results, and so that they work on machines and operating systems of the future. Preferably the tools should all be open source so that limitations can be overcome through collaborative effort.

Projects are often stalled waiting for a supplier to make changes.

Some tools start from a specification of an architecture and help designers produce instances of that architecture (e.g. SOAR, PRS).

Do we know what architecture will be needed five years from now?

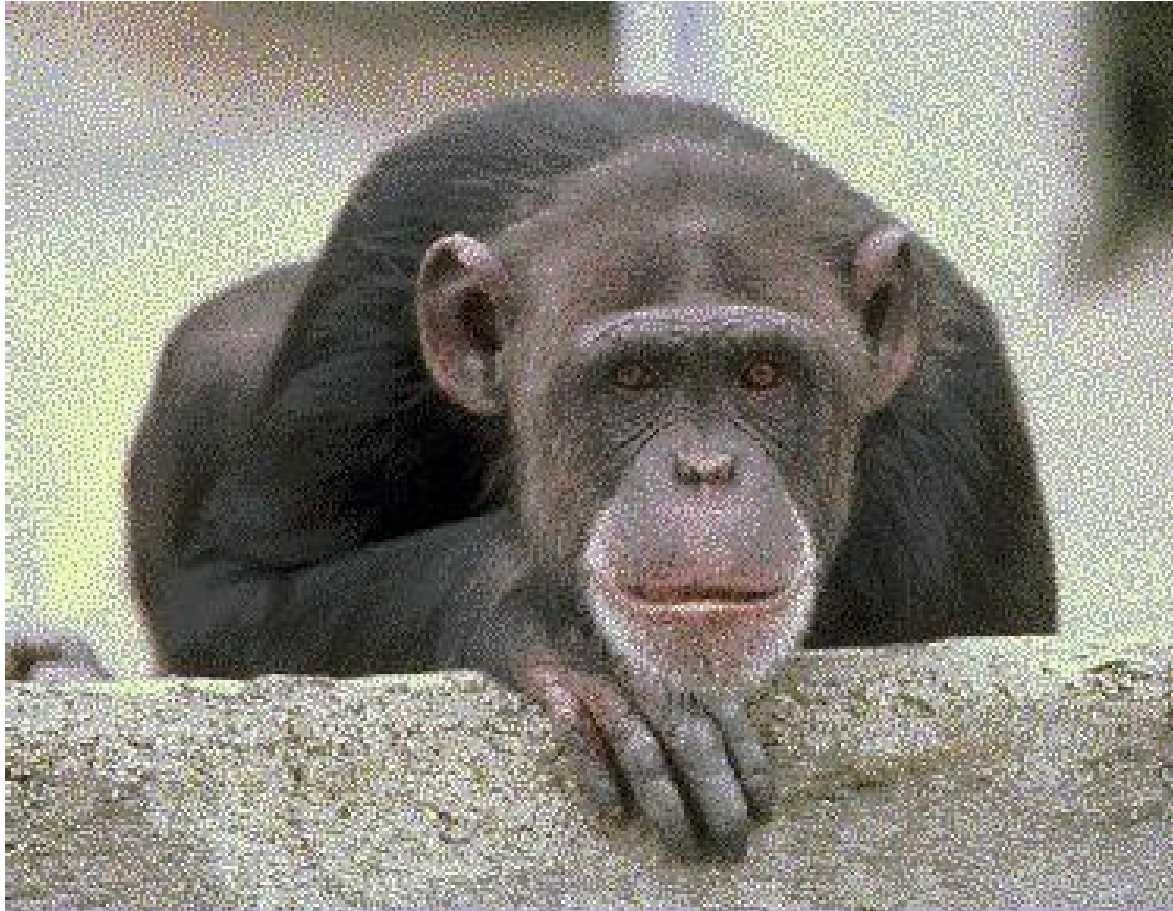
Bham tools: Our own work assumes that we don't yet know what sort of architecture will suffice, so our tools (the SimAgent toolkit) allow exploration of many sorts of architectures, possibly containing many components that work in parallel, possibly at different speeds, possibly using different forms of representation, etc. (E.g. rule interpreters, theorem provers, neural nets). We have also built in many features to support self-monitoring and self-modification (reflection).

For more on SimAgent, see <http://www.cs.bham.ac.uk/axs/cogaff/simagent.html>

This is implemented in Pop11 but could easily be re-implemented in Common Lisp.

(Probably no other existing language has the flexibility required, apart from these two.)

How should we describe his behaviour?



**Perhaps he is trying to make sense of yours?
Will your intelligent assistant be able to?**

Courtesy of <http://www.zoosociety.com/animals/chimp-pics.htm>