Alan Creak
9 December 1993

THOUGHTS ON BUILDING AUTONOMOUS ADAPTING MACHINES

- somewhat after Cliff, Harvey, and Husbands, in any order.

NOTE that "in any order" applies both to the authors and to the thoughts. This is a rather tidier version of an earlier informal note, presented in this form so that I don't lose it.

This long and rambling document started its life as an attempt to digest the work of D. Cliff, I. Harvey, and P. Husbands (hereinafter CHH) on the construction of intelligent robots, and other bits and pieces, and to set it into some sort of perspective against which sensible proposals for future work can be evaluated. It's grown a lot, so that the work of CHH isn't particularly prominent any more, but that's good. I've used my ideas on classifying by learning and action (which get extended in the process) as a framework; I think they helped.

(Throughout this note, various systems are identified as "System <tag>". The tags are taken up again in the not-quite-final section.)

LEARNING AND ACTION.

This bit is extracted from a note I wrote while we were exploring possibilities for Paul's research. It's here because I use the learning-action idea later on. It remains in note form because on attempting to express it in English it became clear that to do so would take too long.

Computers have been used to control machines for many years now, but the results, though impressive and industrially productive, have not yet produced the "intelligent factories" of which people have dreamed from time to time. It has proved far harder to generate anything that we would describe as intelligent behaviour than was expected in the early euphoric days of artificial intelligence research.

PROGRAMMING A COMPUTER TO ACT.

Traditional computer control systems. Good for routine control, assembly lines, simple robots – pick-and-place. Does anything you want, provided you can programme it before the event.

Very useful – most current control systems. But inflexible. Requirements change, new machines on assembly lines, new routines for robots. Don't cope with faults at all well (unless preprogrammed).

PROGRAMMING A COMPUTER TO LEARN TO ACT.

Robot "teaching". Adaptive control systems. Rule-based systems. Easily extensible in principle. "Shallow knowledge".

Programme the computer to accept new rules for action.

PROGRAMMING A COMPUTER TO LEARN TO LEARN TO ACT.

Model-based systems – diagnosis, problem solving, remember solutions. "Deep knowledge" produces more "shallow knowledge".

Programme the computer to develop its own new rules for action by observing what happens.

PROGRAMMING A COMPUTER TO LEARN TO LEARN ?

By analogy : programme the computer to accept new algorithms for learning.

PROGRAMMING A COMPUTER TO LEARN TO LEARN TO LEARN ?

By analogy : programme the computer to develop its own new algorithms for learning by observing what happens.

This is where we want to be – because we don't know how to produce coded algorithms effectively to react to new cirumstances and requirements. Also seems to come closer to human abilities.

Not necessarily as alarming as it sounds. Example : an autonomous vehicle which can explore an environment using a television camera as sensor to find its way about may be moved to a new environment in which satisfactory paths are marked by white lines on the floor. It should be able easily to adapt its learning methods to incorporate this new source of information. We don't expect miracles; if we transferred a human under similar conditions, we might point out the significance of the white lines, so that's fair with the robot – but we don't want to do any reprogramming.

METHODS.

Symbolic : always seen as an advantage of Lisp – programmes are the same form as data, so can be manipulated.

Eurisko : Grew from AM, which was at the "learn to learn" stage, and demonstrated that it wasn't good enough. Lisp – tried to make generalisations, grow new algorithms.

Genetic algorithms : mechanisms for "evolution". Neural networks : versatile learning devices – flexible, adaptable.

OTHER ROBOTIC APPROACHES.

Winograd : "classical" : interesting, but never got very far – too inflexible.

Brooks : subsumption : can generate complex behaviour, but all designed in.

Kuperstein : neural networks : restricted to very simple observations ?

Waxman : Mavin : cognitive model : very complex, all preprogrammed ?

CHH : evolutionary robotics : genetic algorithms : closest to this project, only example of $learn^3$? only neurons and genetic algorithms.

ADAPTING.

When CHH speak of "adaptive systems", they don't mean that their control systems will adapt to circumstances – they have almost gone out of their way to make sure that their control systems can't adapt to anything at all. They mean that, over *N* generations, their evolutionary system will evolve something appropriate to the pertaining conditions (System CHH). That isn't the sort of adaptive system I want in *my* robot.

What I want is a system which will learn to cope with new circumstances in something like the way I do – by deploying what it knows and what it discovers by experiment to perform some task under whatever conditions present themselves, and to adjust its behaviour as the ambient conditions change (System Me). The task could be as simple as "keep near the middle of the space bounded by this circular wall", or it could be as complicated as "survive". (It isn't necessarily a robotic system; Kim's work(1) fits into this rather general description (System Kim).) People, and a lot of animals, can learn new tricks; it's a lot faster than evolution. It means that they can learn from their experience within their lifetimes.

Of course, the *really* clever bit is the way people manage to learn from *other people's* experience, by communication. This makes it possible for every individual to benefit, at least in principle, from all the experience of each. It's a *lot* faster than evolution, and much more effective than going it alone. How would you go about building a learning, evolving, self-improving system based on communication ? (If you were desperate to fit my Ph.D. students into the act, you could say that Tim's "processing elements" do something quite like that, but it's at a very low level (System Tim).)

Such a system would presumably resemble the CHH system in relying on a population of similar, but not identical, individuals, each working on some task, or perhaps on a set of similar tasks, and exchanging information on how they are getting on. To make it practicable, that almost certainly means simulation, as it does for CHH.

LEARNING IN CHH SYSTEMS.

Just what is it that their systems learn ? At the beginning of the evolution, they have an ignorant machine, and an administrative system which knows about breeding. At the end of the evolution, they have a machine which knows how to get to the middle of a circle, and the breeding machine. The *system* has learnt the solution to a problem.

It has also had a lot of experience, but that is all gone. Both parts of the system – the robot and the breeder – acquire experience as the experiment proceeds, but the system is unable to do anything with it.

At the robot level, one can argue that the activities necessary for control and intelligence are essentially similar, and use the same machinery. If no intelligence is evolved, then, it is because there is no selection pressure to encourage evolution of means to use the experience. No credit is given for intelligence, there are no variables in the environment which the robot must deal with, there is nothing for an individual robot to learn because the environment is always the same and all its behaviour can therefore be hard-wired in. Perhaps more challenging environments would stimulate the evolution of intelligence (System CHH+Int).

An alternative view is that if we know that intelligence might be useful, then it is sensible to provide basic intelligent organs, just as the CHH robot is provided with touch sensors and photoreceptors (System Intelligent). In this view, it is an accident that the most common example of the machinery of intelligence has something in common with corresponding example of the machinery of control, and not necessarily fruitful to expect them both to evolve together. (If that's so, it might be less constraining to permit them to evolve more or less separately.)

By including the breeder level, that argument can be taken a step further. An intelligent robot will be able to profit from its own experience by learning to perform its task more effectively; an intelligent breeder should be able to profit from *its* experience by learning to breed more effectively (System Breeder). How could that be managed ? The obvious way in the CHH context is to have a large population of breeders and use an appropriate genetic algorithm on them (System Breeder^2). (This becomes reminiscent of the Grand Canonical Ensemble !) I don't think we need any more levels, though, provided that we can make our breeders sufficiently general so that they can breed anything – including themselves. That isn't as simple as it seems, though; see the discussion of System Breeder^2 below for more comment. (This will be all right until they evolve consciences, and turn against eugenics, but all being well, or ill, that will take some time.)

While this suggestion unifies the breeding and robot functions in the abstract, there are still *two* machines (the robot and the breeder) and *three* activities (control, intelligence, and breeding). In us, all these are combined (System Us), though we also have consciences and therefore restrict our eugenics to covert methods. (I except Hitler, Pol Pot, and certain people from Yugoslavia.) Is it reasonable to aim at some such unified structure for a robot population (System Unified) ? One could imagine a single engine completely controlled by a chromosome (biological details not guaranteed) and covering selection, intelligence, and behaviour.

The value of such a unified approach depends on the aim of the exercise. There are intrinsically two levels of activity in the complete system. Control and intelligence are local activities which are carried on by individual robots; breeding is a population-wide activity. People are not a good model here, because we complicate matters by breeding, quite locally, in pairs. A single breeding engine which can select and juggle "genes" from the whole population is surely a much more efficient means of selection. On the other hand, as a population grows a local method of breeding scales a lot better.

If there's a conclusion, it's probably in favour of retaining the separation between local development (breeding or otherwise) and global action, unless the aim is to work towards machines with as great a degree as possible of real autonomy; but that's a long way from keeping in the middle of a circle. I conclude that the question isn't urgent in practice for CHH, but that the principle may be important in trying to design a new system.

NEURAL NETWORKS.

The obvious competition to the CHH ideas comes from more conventional neural network techniques and from symbolic artificial intelligence. How might one attempt to tackle the CHH problem from these positions ?

First, then, what's a "conventional" neural network analogue of the CHH approach ? I take it that a conventional neural network is one in which many individual processing elements are connected with no topological restrictions by synapses of variable strength, each of which conveys the output of one neuron to the input of another, and that there is some learning mechanism by which the strengths of the synapses can be changed systematically in response to the network's behaviour.

There are **similarities** to the CHH machines : the networks can adapt, grow, shrink (usually implemented by a separate pruning algorithm, but there's no obvious reason why it shouldn't be automatic).

There are **differences** from the CHH machines : the functions of adaptation, growth, and shrinkage can all happen within a single network as it develops in isolation, while the corresponding phenomena in the CHH context can only happen at the breeding stage. The network can learn, so that experience is not destroyed, and there is some point in "rewards" and "punishments". Genetic algorithms are much more brutal to individuals. All individuals die anyway, no matter how well they have performed; the good ones will have progeny in the next generation, while the bad ones won't. (Compare *Brave New World* ?)

Genetic algorithms have a characteristic which may be quite valuable : they keep designing new models, not just accreting new bits onto old ones. This property endows the technique with the advantages of a (rather abstract) sort of sexual reproduction – in effect, the evolution is guided by the experience of all members of the population, so it is possible to cover much more ground in a generation. On the other hand, the children have to start from scratch every time.

SYMBOLIC ARTIFICIAL INTELLIGENCE.

There has been a lot of work on learning systems using traditional artificial intelligence methods. I don't propose to review it; it's textbook material, and I don't think I could do a good job on it anyway. Generally, though, learning is associated not only with acquiring new information (which is easy), but with fitting the information into a structure of some sort. That's because we suppose that learning itself isn't much use; we have to be able to use the material learnt in some way, and unless it is useful simply to regurgitate it as it came to us, that means it has to be linked with something else. (For example, outside the world of Scrabble it isn't much use just knowing that *mixen* is a word; you have to link it to its meaning – a dunghill – before it makes sense, and we need some sort of words ↔ meanings table so that we can find it when we need it.)

Without a prepared structure, learning is hard – which is why there's a dictum to the effect that the easiest thing to learn is something you almost know already. That seems to work for people, too. Most learning work has therefore been on topics for which we can prepare structures.

I think that there's been rather little on exploratory learning, where the task is not only to learn but to find out what there is to learn. Even these can only get one or two steps back from built-in structure; simply because computer programmes operate in terms of structures, you can't avoid providing a structure for *something*. If your provided structure is a structure which helps you to build new structures, you may be getting somewhere. I suppose one would hope to end up with a structure that could build itself, which might then be self-sustaining in some sense, though I'm not very clear just what sense that would be !

Lenat's programmes, AM and Eurisko, are the best examples I know, but I'm out of date.

The most strikingly distinctive feature of symbolic methods is that they're comprehensible. This is hardly surprising; they are undoubtedly modelled on their originators' views of how they (the originators) go about thinking. As compared with neural networks and genetic algorithms, they operate at a much higher conceptual level (a judgment which itself begs questions), and can therefore do some things impressively well. On the other hand, they are complicated, and *very* brittle, so can't easily change into something better.

It's interesting that Lisp was invented as a high-level computer language in which programmes and data were indistinguishable, so that programmes could easily write new programmes, or even change themselves; the aim was to provide a programming environment in which artificial intelligence would flourish. It's my impression that this has proved about as successful as corresponding claims that digital computers were good because programmes could change their own code. To some degree, Eurisko exploits this property of Lisp (in a rather genetic-algorithmic way), but I don't recall any other specific examples.

ANALYSIS.

I shall try to categorise the areas I've mentioned in terms of *learning* and *action*, introduced earlier.

System CHH : The CHH system's robots are programmed (by the breeder) to act. The breeder is also programmed to act. Neither is programmed to learn – and indeed neither does learn – but because of the actions taken by the breeder the system as a whole learns to act. (So much for the idea that a computer can only do what it's programmed to do ! – and what about the Chinese Room argument ?)

System Me : The (unspecified) "system" is programmed to learn to act. Functionally, it's like System CHH; I prefer it because it sounds quicker. The similarity between the systems is to a large degree a consequence of the loose specification for System Me.

System Kim : System Kim, as far as it's defined at present, is likely to be a conventional neural network, programmed to learn a set of codes to be translated, programmed to act by translating the codes, and programmed to learn to act in that it must amend its actions to track changes to those codes with time. Conventional networks should be enough to do the trick(2); the result is complex, because everything has to be programmed in.

System Tim : Tim's TME network(3) is programmed to learn to act. It can be programmed to do a variety of things by giving it appropriate examples, but essentially it always learns the coefficients of the neural network in its "programming elements" which is best suited to the job to be done. It is interesting that Tim's network resembles the CHH system by having a local and a global level, though the global level is not made obvious. Locally, the network performs some transformation from an image onto a set of fields corresponding to required classes of data; globally, the experience of all the networks is collected and shared by a rather abstract supervisor, which is where the communication comes in.

System CHH+Int : The breeder is still programmed to act, and the robot is still programmed to act. So where's the "intelligence" ? Hm. I suppose there still isn't any – though trying to identify intelligence by considering how it's done isn't the best way to find it. What one might perhaps get is a *fixed* system – because there's nothing else available in the architecture – which could cope with different conditions. That's what the current versions do, in effect, when they switch from one mode of operation to another(4). Could it ever even conceivably do better ? To do so, I think it would have to be able to work with an architecture which contains some element of variability, so that it could invent new components for itself in response to new experiences. It can almost certainly do that by evolving large numbers of various sorts of switch and logic unit, and wiring up an adaptive machine of some sort. This could take a while ! It begins to sound a bit like a Turing machine (System Turing).

I've avoided the question, though. The CHH+Int system certainly exhibits learning even if it isn't programmed in anywhere. We presumably regard this as an emergent property, and one of the characteristics of evolution. We could say that the breeder is programmed to act, and the robot is programmed to act, and the system can learn to act. But questions are begged yet again. Emergent property or not, the learning is no accident, and somebody put it there. Even though there's no specific line of code in which learning identifiably happens (perhaps the closest is in the choice of which members of the population to carry forward for breeding), the intention is there somewhere, and I should say that the breeder is programmed to ....

Ah. (That paragraph ending is not contrived for dramatic effect – that's how it happened. Real-time inspiration again.) But the breeder *doesn't* learn, and neither do the robots. The only entity that does learn is the "system", whatever that is. (That's where I was in System CHH.) Or, perhaps, the gene pool. That's better – at least it's a sort of database. Ah, again. So the breeder is "really" just an old-fashioned artificial intelligence programme using generate-and-test ! The only oddity is that the test part is done externally. I conclude, with some relief, that the breeder, including its database, the gene pool, does learn after all. But it learns to act indirectly; its own action – building new robots – isn't the action which is judged. Now I've got here, I don't know how to fit that into my pretty classification. Perhaps it won't. Not to worry.

System Intelligent : By definition, this is the previous system but with intelligence hardwired into the robot : the breeder is programmed to act, and the robot is still programmed to act – and to think. (It was at this point that I went back and expanded the CHH+Int discussion past the first paragraph.) Better, the breeder is still programmed to learn by generate-and-test, but with a more optimistic evaluation function. That's not better – it's rubbish.

It's rubbish because I've missed the point somewhere. The object of the exercise is to build better robots. Surely, then, building a cleverer robot should be judged by some criterion other than how well it contributes to the effectiveness of evolution. For all my obsession with the breeder and the system, the robot is the important component, and should be at the centre of this discussion.

All right. So where does that get me ? The CHH robot was programmed to act; the Intelligent robot is programmed to act and to learn to act. But what about the breeder ? That hasn't changed – it is still programmed to learn to act. I wonder, though, whether it might be worth separating out its particular sort of action as specially significant. It certainly has a peculiar relationship to the system's performance which other actions don't, as it operates in a sense within the system, not outside. And it isn't a new action; it's programming. In these terms, the breeder is programmed to learn to programme.

Now I have three significant ideas in my attempt at classification – action, programming, and learning. I am not quite sure just what I mean by "programming" here; I suspect that I really mean something like "action internal to the system", but "programming" will do for the time being.

System Breeder : This is derived from System Intelligent by making the breeder learn as well as the robots. (In fact, the two advances in intelligence are probably independent, so there's another model with a clever breeder and silly robots. The details are left as an exercise for the reader.) Now the robot is programmed to act and to learn to act; the breeder is programmed to learn to learn to programme.

System Breeder^2 : This system is derived from System Breeder by adding a new level of breeder which breeds breeders. Call the new breeder the breeder^2. Applying the description of System Breeder in the obvious way, the breeder^2 is programmed to learn to learn to programme to learn to learn to programme. (Some of the "learn"s can be cut out by not having intelligent breeders. I think that's a detail.) The question is whether the breeder and breeder^2 are the same – and, if not, for what value of n, if any, is breeder^n = breeder^{n+1} ?

This is not meant as a frivolous question. If we can find a finite value for n, then a self-contained and in some sense autonomous system is in principle achievable. (Just what sort of autonomy can be ascribed to an abstraction which lives only in a computer is open to question.) If not, then we shall always have to contrive something new for the next level, and the whole thing becomes rather trivial after the first level or two.

I therefore ask : what's the difference between breeder and breeder^2 ? I answer : I don't know, as we haven't built breeder^2. I therefore further ask : can we use breeder for breeder^2 ? I answer : perhaps – but two conditions must be satisfied. First, the breeding mechanisms must be the same. That isn't obviously an insuperable problem; just as one can describe the properties of the robot in generic terms, so one can presumably describe the properties of a breeder. Second, though, there must be additional functions in the environment, of sorts which I haven't yet considered. I can think of two : an *arbiter*, and a *womb*.

The *arbiter* is needed to assess the members of the population so that the breeder can breed selectively. Someone has to say which traits are desirable in the population; the arbiter is it. We can only manage a breeder^2 if we can find a universal arbiter at some level.

The *womb* is needed to express the genetic material by building the corresponding machine. As with the arbiter, we must have a universal womb if there is a breeder$^\infty$.

System Unified : This system is proposed to address some of the comments above. All changes are brought under the same umbrella, and governed by the same mechanism. Its list of components should, by the discussion above, be extended to include womb and arbiter, but provided that it is a breeder$^\infty$, that doesn't matter a lot. Is it feasible ? I suspect that the major difficulty (where "major" is a matter of scale, with even the "minor" difficulties well beyond our solution for some time to come) is likely to be in the arbiter. How can one judge the performance of all the levels of activity ? In natural evolution, the arbiter is survival, but that's just a bit crude, and it's only managed to produce one improvement in the breeder (from asexual to sexual reproduction) in 10^9 years or so. I don't think we can wait that long. (The Ph.D. limit is six years.)

System Turing : Presumably, if the Church-Turing thesis is accepted, this represents the limit. Are the devices doing anything that Turing machines can't ? (It seems highly unlikely.) Or vice versa ? (Probably not.) So what's new ?

System Us : This comes after System Turing just in case we are cleverer than Turing machines. (That "System Me" turned up earlier is not to be interpreted in an analogous manner.) Whether you believe that we're accidents or something more, though, we have a lot of machinery which does, more or less, combine some of those functions into one. We act, we learn to act, perhaps we learn to learn. We programme (not just computers – they exist only *because* we found that programming was a good idea), we learn to programme, we programme to act, we programme to learn. We do all right; do our robots (or whatever) need to do any better ?

They wouldn't need to do any better (learn to learn to learn ..., etc.) if they could do it as well as we can. We are very good at learning; but we have a lot of machinery for doing it. The CHH thesis is that, as we can't provide our robots with the right machinery (in particular, even if we can provide enough, we don't know how to programme it), they need the "higher order" functions in order to evolve the machinery they need.

Our main achievement, so far as evolution goes, is to try to subvert it. We have done so by changing the arbiter. Survival now depends not just on basic survival skills but on what sort of country you live in. We are rather good at keeping alive people who in earlier years would have died from their congenital conditions; we have consciences, some of which tell us (again, in some sorts of country) that everyone who meets the basic requirement of being alive has the same rights as everyone else – which includes the right to breed. On the other hand, our consciences are highly malleable, and can easily be trained to abhor those things we find inconvenient, such as unfortunate pregnancies and Iraqis. Even without a planned eugenic policy, all this will presumably mount up in time.

The point of that diatribe is that the consequences of intelligence in an autonomous evolving population may not be simple. The potential for messing things up comes in when the intelligence can be applied to the selection procedure – but perhaps that's one of the things we need if we are to improve our systems quickly. I don't know whether these conclusions, such as they are, apply to developing systems which don't depend on evolutionary mechanisms, but I suspect that there will be parallels.

SO WHAT'S THE RIGHT WAY ?

What sort of machinery would be appropriate ? The obvious starting point is something which could learn rules of action. Just how best to use this might be a matter to leave to the system.

(Note that this is beginning to look like stuff done by early cyberneticists.)

The communication metaphor.

There is one feature of System Us which I've mentioned before, but I think it's very significant. We begin with a lot of machinery for intelligence – but we don't have to evolve new strains of people who are born knowing how to drive cars or use computers or even go to the middle of a circle. We are born with abilities to learn, and to communicate, and that's how we get much (most ?) of our variety. Not only do we communicate instructions – which is to say, programmes – to each other; we share experience, so that many people can learn from one event. By writing down our experiences and programmes, we can make them available to the whole population.

Can we use this model, rather than the evolutionary model, as a basis for building self-improving machines ? ("Improving" here means something quite like "adapting", but that didn't seem quite right.) Given that we can communicate effectively, there may be no need to reproduce. Indeed, one can argue that reproduction is just a nuisance, as its only effect is to set an organism back to zero with no personal experience to guide it in using the global experience of the species. On the other hand, one can also argue that restarting is a means of avoiding obsessions; an immortal organism may become so bound by its own interpretations of events that it continues to pursue fruitless lines of development, whereas a new assessment of the evidence may reject these lines as unprofitable.

How could such a model be transformed into a practical programme (there it is again) for action ? There are certain criteria which must be met, and I don't know what they are. One such is that it is important to retain some variety in the system, otherwise one might lose ideas just because one of them wins at the moment and everything behaves logically. We presumably don't want to insist that the system behaves illogically, but somehow we want to maintain distinct points of view. One possible solution, taken from the evolutionary model, is to have many individuals, working on different tasks. (The tasks probably have to be related, or there won't be anything to compare.) Each individual publishes its experience, and each must decide, on the basis of its own experience, whether or not to adopt the published methods. Here are two ways in which this exchange of experience could be organised – doubtless there are many other possibilities.

1 : Each individual compares published solutions with its own history of problems, asking whether the published solution would have worked in its own case. If so, it would add the solution to its own repertoire, maybe with some measure of performance which could be refined by experience.

2 : Each individual refers to the published list when it encounters a problem, estimates the likelihood of success of the listed solutions, and tries the method for the problem which looks best.

Any such method depends critically on the method of publication. I am not at all confident that I can even classify the possibilities. The "traditional" approaches are the symbolic knowledge representation techniques of artificial intelligence, based on linguistic or semantic net models. Just how well those might fit with, say, neural network machinery isn't clear. ART networks implement an alternative knowledge base, in a sense, and one could imagine an ART skeleton with plug-in neurons which could be acquired from some database to provide for specific sorts of memory. Alternatively, neural knowledge can – sometimes – be converted into rules, and the rules converted back again. One of my suggestions for Kim's work was to find a way to base a neural network on a symbolic table. Clearly, if one wished to follow this track, there would be many (very interesting) avenues to explore.

Communication with analysis.

Simply sharing knowledge is all very well, but it doesn't allow for any innovation. To generate new behaviour, which is certainly something we want, there must be something in the system which can analyse the performance of different methods (the arbiter), and can then *work out* new algorithms (a womb ?). Once again, one thinks of symbolic methods, though that may just be habit. One can certainly imagine neural-like approaches to such questions, though my imaginings also involve quite a bit of rather specific switching machinery. No, they are not yet ready to exhibit.

What part of the system should perform this task of analysis ? In a system without communication, it would have to be the active modules themselves, whatever forms they might take, but with communication it is possible to incorporate specialised innovators which can propose new techniques. It would remain the task of the active modules to accept or reject the techniques. (Perhaps the innovators correspond to research institutes ?)

CONCLUSION.

If there is one, it has something to do with the desirability of the communications metaphor as a model for development.

I'd still like to have a way to classify the different approaches. I think the { learning, programming, action } set might still be a promising basis for such a categorisation, but my struggles as recorded above in real time demonstrate that the categorisation is not always easy. Still, that isn't necessarily a fatal disease, provided that the result is illuminating. The process has certainly illuminated me.

REFERENCES.

1 : G.A. Creak : *Some notes on a proposed interface device*, unpublished working note AC78 (December, 1990).

2 : G.A. Creak, H.-W. Gellersen : *An adaptive machine access system for the handicapped based on neural networks*, Auckland University Computer Science Report #60 (Auckland University Computer Science Department, May, 1992).

3 : T.J. Stucke, G.A. Creak, G.G. Coghill, "The Minds Eye: Extraction of structure from images of objects with natural variability" *ACNN'93, Proceedings of the Fourth Australian Conference on Neural Networks*, February 1993.

4 : I. Harvey, P. Husbands, D. Cliff : *Genetic convergence in a species of evolved robot control architectures* (University of Sussex, Cognitive Science Research Papers CSRP 267, February 1993). (– or another report by the same authors.)