

TEACH AITHEMES

A PERSONAL VIEW OF ARTIFICIAL INTELLIGENCE

Aaron Sloman

School of Cognitive and Cognitive Sciences (COGS), University of Sussex
At University of Birmingham since 1991

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

This file is <http://tinyurl.com/BhamCog/personal-ai-sloman-1988.html>

It is also available as PDF: <http://tinyurl.com/BhamCog/personal-ai-sloman-1988.pdf>

This was originally published as the Preface to
Computers and Thought: A Practical Introduction to Artificial Intelligence,
(Explorations in Cognitive Science)

By Mike Sharples, David Hogg, Chris Hutchinson, Steve Torrance, David Young
MIT Press, 20 Oct 1989 - 433 pages

Available without diagrams online here, both browsable and as a zip package:

<http://www.cs.bham.ac.uk/research/projects/poplog/computers-and-thought>

<http://www.cs.bham.ac.uk/research/projects/poplog/computers-and-thought.zip>

Related teaching material for use with Poplog/Pop11

http://www.cs.bham.ac.uk/research/projects/poplog/contrib/pop11/ct_book

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

This preface has also been available since about 1988 as 'TEACH' file in
the Poplog system: TEACH AITHEMES

See also

<http://tinyurl.com/thinky-ex>

Thinky programming and other kinds

<http://tinyurl.com/thinkyprog>

Tips on how to teach thinky programming:

<http://tinyurl.com/PopVidTut>

Video tutorials on some of this material

CONTENTS

- Introduction
- What then is AI?
- Goals of AI: the trinity of science
- But what is intelligence? Three key features:
- Intentionality
- Flexibility
- Productive laziness
- Sub areas of AI
- A simple architecture
- Sketch of a not very intelligent system
- Limitations of the model
- Less ambitious projects
- Key ideas in AI models
- Computers vs brains
- "Non-cognitive" (?) states and processes
- Conceptual analysis
- Tools for AI
- An example of the expressive power of an AI language
- Horses for courses: multi-language, multi-paradigm systems
- Conclusion

-- Bibliography

-- Introduction

There are many books, newspaper reports and conferences providing information and making claims about Artificial Intelligence and its lusty baby the field of Expert Systems. Reactions range from one lunatic view that all our intellectual capabilities will be exceeded by computers in a few years time to the slightly more defensible opposite extreme view that computers are merely lumps of machinery that simply do what they are programmed to do and therefore cannot conceivably emulate human thought, creativity or feeling. As an antidote for these extremes, I'll try to sketch a sane middle-of-the-road view.

In the long-term AI will have enormously important consequences for science and engineering and our view of what we are. But it would be rash to speculate in detail about this.

In the short to medium term there are extremely difficult problems. The main initial practical impact of AI will arise not so much from intelligent machines as from the use of AI techniques to build 'intelligence amplifiers' for human beings. Even if machines have not advanced enough to be capable of designing complex systems, discovering new concepts and theories, understanding speech at cocktail parties and taking all our important economic, political and military decisions for us, AI systems may nevertheless be able to help people to learn, plan, take decisions, solve problems, absorb information, find information, design things, communicate with one another or even just brain-storm when confronted with a new problem.

Besides helping human thought processes, AI languages, development tools and techniques can also be used for improving and extending existing types of automation, for instance: cataloguing, checking software, checking consistency of data, checking plans or configurations, formatting documents, analysing images, and many kinds of monitoring and controlling activities.

But there is no sharp boundary between such AI applications and computer science generally. Indeed the boundary is not only fuzzy but shifts with time, for established AI techniques and solved AI problems are simply absorbed into mainstream computer science. A striking example is compiling: once only human beings could understand algebraic expressions, and making a machine do likewise was a problem in AI. Now any humdrum compiler for a programming language can do it (apart from some quirky languages, like simpler versions of the most widely used AI language, namely LISP!).

-- What then is AI?

Some people give it a very narrow definition as an applied sub-field of computer science. I prefer a definition that reflects the range of work reported at AI conferences, in AI journals, and the interests and activities of some of the leading practitioners, including founders of the subject. From this viewpoint AI is a very general investigation of the nature of intelligence and the principles and mechanisms required for understanding or replicating it. Like all scientific disciplines it has three main types of goal, theoretical, empirical, and practical.

-- Goals of AI: the trinity of science

The long term goals of AI include: finding out what the world is like,

understanding it, and changing it, or, in other words:

- (a) empirical study and modelling of existing intelligent systems (mainly human beings);
- (b) theoretical analysis and exploration of possible intelligent systems and possible mechanisms, architectures or representations usable by such systems;
- (c) solving practical problems in the light of (a) and (b), namely:
 - (c.1) attempting to deal with problems of existing intelligent systems (e.g. problems of human learning or emotional difficulties) and
 - (c.2) designing new useful intelligent or semi-intelligent machines.

In the course of these activities AI generates new sub-problems, and these lead to new concepts, new formalisms, and new techniques.

Some people restrict the term 'Artificial Intelligence' to a subset of this wide-ranging discipline. For example those who think of it as essentially a branch of engineering restrict it to (c.2). This does not do justice to the full range of work done in the name of AI.

In any case, it is folly to try to produce engineering solutions without either studying general underlying principles or investigating the existing intelligent systems on which the new machines are to be modelled or with which they will have to interact. Trying to build intelligent systems without trying to understand general principles would be like trying to build an aeroplane without understanding principles of mechanics or aerodynamics. Trying to build them without studying how people or other animals work would be like trying to build machines without ever studying the properties of any naturally occurring object.

The need to study general principles of thought, and the ways in which human beings perceive, think, understand language, etc. means that AI work has to be done in close collaboration with work in psychology, linguistics, and even philosophy, the discipline that examines some of the most general presuppositions of our thought and language.

This is why, at some Universities, AI has not been restricted to an engineering department. In fact it is now often to be found in several different areas of a University. E.g. at Sussex University it is in several different Schools including the School of Cognitive Sciences.

The term 'Cognitive Science' can also be used to cover the full range of goals specified above, though it too is ambiguous, and some of its more narrow-minded practitioners tend to restrict it to (a) and (c.1).

-- But what is intelligence? Three key features:

The goals of AI have been defined in terms of the notion of intelligence. I don't pretend to be able to offer a definition of 'intelligence'. However, most, if not all, the important work in AI arises out of the attempt to understand three key characteristics of the kind of intelligence found in people and, to different degrees, other animals. The features are intentionality, flexibility, and productive laziness.

-- Intentionality

This is the ability to have internal states that refer to or are ABOUT entities or situations more or less remote in space or time, or even non-existent or wholly abstract things.

So intentional states include contemplating clouds, dreaming you are a duke, exploring equations, pondering a possible action, seeing a snake or wanting to win someone's favours. These are all cases of awareness or consciousness of something, including hypothetical or impossible objects or situations. A sophisticated mind may also have thoughts or desires about its own state - various forms of SELF consciousness are also cases of intentionality.

Particular categories of intentional states include:

- perceiving something
- believing or knowing something
- wanting something, or having something as a goal
- considering or imagining a possibility
- asking a question about something
- having a plan or strategy

All intentional states seem to require the existence of some kind of REPRESENTATION of the content of the state: some representation of whatever is believed, perceived, desired, imagined, etc. A major theme in AI is therefore investigation of different kinds of representations and their implementation and uses. This is a very tricky topic, since there are many different kinds of representational forms: sentences, logical symbols, computer data-bases, maps, diagrams, arrays, images, etc. It is very likely that there are still important forms of representation waiting to be discovered.

Moreover, many representations are themselves abstractions that are not necessarily explicitly or directly embodied in physical structures, for example a very large sparse array that is encoded in a compact form. It is therefore useful to talk about 'virtual representations' as opposed to physical representations.

A particularly important case involves the use of inference procedures. If new conclusions can be drawn from what is represented, then besides the information stored explicitly there is additional information that can be DERIVED when needed. Thus we all have knowledge of arithmetic that goes beyond the tables we have learnt explicitly, since we know how to derive new facts from them. A different example is using an old map to work out a new route. Different kinds of representations require different kinds of inference mechanisms.

One reason why computers are powerful tools for exploring intentional systems is that they can very rapidly construct or change virtual representations, whereas mechanical construction would often be too slow to deal with a world that waits for no man or machine. Brains also seem to have this ability, though exactly how they do it remains largely unexplained. Perhaps new kinds of machines will one day exhibit new kinds of rapid structural variability enabling new kinds of intelligence to be automated.

-- Flexibility

This has to do with the breadth and variety of intentional contents, for instance the variety of types of goals, objects, problems, plans, actions, environments etc. with which an individual can cope, including the ability to deal with new situations using old resources combined and transformed in new ways.

Flexibility in this sense is required for understanding a sentence you have never heard before, seeing a familiar object from a new point of view, coping with an old problem in a new situation, dealing with unexpected obstacles to a plan. A kind of flexibility important in human intelligence involves the ability to raise a wide range of questions.

A desirable kind of flexibility often missing in computer programs is 'graceful degradation'. Often if the input to a computer deviates at all from what is expected the result is simply an error message and abort, or worse in some cases. Graceful degradation on the other hand would imply being able to try to cope with the unexpected by re-interpreting it, or modifying one's strategies, or asking for help, or monitoring actions more carefully. Instead of total failure, degradation might include taking longer to solve a problem, reducing the accuracy of the solution, reducing the frequency of success, and so on.

One of the factors determining the degree of flexibility will be the range of representations available. A system that can merely represent things using a vector of numerical measures, for example, will have a narrower range of possible intentional states than a system that can build linguistic descriptions of unlimited complexity, like:

the man
the old man
the old man in the corner
the old man sitting on a chair in the corner
the sad old man sitting on a chair with a broken leg in the corner
etc.

So flexible control systems of the future will have to go far beyond using numerical measures, and will have to be able to represent goals or functions, and relationships between structures, resources, processes, constraints, and so on.

Another requirement for flexibility is non-rigid control structures. In most machines behaviour is pre-determined by structure. Computer programs with conditional instructions allow more flexibility. Even greater flexibility is achieved by turning the whole program into a set of condition-action rules, as is done in some AI programming languages known as 'production systems'. Then, instead of the programmer having to determine in advance a good order in which tests should be made and actions attempted, the rule interpreter can examine the applicable rules and decide in the light of the context at 'run time'. If the program can change the set of rules yet more flexibility is available.

However, an excess of flexibility can cause its own problems, notably a lack of control. That leads to the idea of a layered process architecture where some kind of higher level supervisor program watches over the actions of lower level programs and decides when they need to be suspended, modified, or aborted. This kind of flexibility is not much in evidence in AI programs yet, but will become increasingly feasible as computer power becomes cheaper and more readily available.

Different kinds of flexibility are to be found in different organisms. For example, birds that can build only one sort of nest may nevertheless be very flexible and adaptive in relation to availability of materials

and sites for such nests. Many aspects of human intelligence range over a potentially infinite variety of structures - for instance infinitely many sentences, dance movements, algebraic equations, or social situations. To account for this we need to study the generative power of the underlying mechanisms and representations, as well as mechanisms that allow major changes of direction in the light of new information.

-- Productive laziness

It is not enough to achieve results: intelligence is partly a matter of HOW they are achieved. Productive laziness involves avoiding unnecessary work.

A calculator blindly follows the rules for multiplication or addition. It cannot notice short cuts. If you tell it to work out 200 factorial minus 200 factorial, it will do a lot of unnecessary computation, and perhaps produce an overflow error. The intelligent solution is a far more lazy one. A chess champion who wins by working through all the possible sequences of moves several steps ahead and choosing the optimal one is not as intelligent as the player who avoids explicitly examining so many cases because he notices some higher level pattern that points directly to the best move.

The implications of this kind of laziness are profound. In particular, noticing short cuts often requires using a far more complex conceptual structure, such as might be needed to discern high level symmetries in the problem space. Compare trying to answer the question 'Is there a prime number bigger than a billion?' by searching for one, with Euclid's lazy approach of proving in a few lines that there is no largest prime number.

Why is laziness important? Given any solvable task for which a finite solution is recognizable, it is possible in principle to find a solution by enumerating all possible actions (or all possible computer programs) and checking them exhaustively until the right one turns up. In practice this is useless because the set of possibilities is too great.

This is the 'combinatorial explosion'. Any construction involving many choices from a set of options has a potentially huge array of possible constructs to choose from. If you have four choices each with two options the total set of options is sixteen. If you have twenty choices each with six options, the total shoots up to 3,656,158,440,062,976. Clearly exhaustive enumeration is not a general solution. The tree of possible moves in chess is larger than the number of electrons in the Universe (if we are to believe the physicists). So lazy short cuts have to be found.

For example a magic square is an array of numbers all of whose rows columns and diagonals add up to the same total. Here is a 3 by 3 magic square made of the digits 1 to 9.

```
672
159
834
```

If you try to construct an N by N magic square by trying all possible ways of assigning the NxN numbers to the locations in the square then the number of possible combinations is the factorial of NxN. In the case of the 3x3 square that makes 362,880 combinations. Trying them all would not be intelligent. A sensible procedure would involve testing partial combinations to see whether they can possibly be extended satisfactorily, and, if not, rejecting at one blow all the combinations with that initial sequence.

It is also sensible to look for symmetries in the problem. Having found that you can't have the number 5 in the top left corner, reject all combinations that involve 5 in any corner.

Yet more subtle arguments can be used to prune the possibilities drastically. For example, since eight different triples with the same total are needed, it is easy to show that large and small numbers must be spread evenly over the triples, and that they must in fact add up to 15. So the central number has to be in four different triples adding up to 15, the corner numbers in three triples each, and the mid-side numbers in two each. For each number we can work out how many different triples it can occur in, and this immediately restricts the locations to which they can be assigned. E.g. 1 and 9 must go into locations in the middle of a side, and the only candidate for the central square is 5. In fact, a high level symmetry shows that you need bother to do this analysis only for the numbers 1 to 4. You can then construct the square in a few moves, without any trial and error. What about a two by two magic square containing the numbers 1, 2, 3 and 4? Think about it!

These examples show that the ability to detect short cuts requires the ability to DESCRIBE the symmetries, relationships, and implications in the structure of the task. It also requires the ability to NOTICE them and perceive their relevance, even though they are not mentioned in the statement of the task. This kind of productive laziness therefore depends on intentionality and flexibility, but motivates their application. Discovering relevant relationships not mentioned in the task specification (e.g. "location X occurs in fewer triples than location Y") requires the use of a generative conceptual system and notation.

An intelligent problem solver therefore requires a rich enough representation language to express the constraints and describe relevant features, and a powerful inference system to work out the implications for choices. Being lazy in this way is often harder than doing the stupid exhaustive search. But it may be very much faster. This points to a need for an analysis of the notion of intellectual difficulty.

Productive laziness often means applying previously acquired knowledge about the problem or some general class of problems. So it requires learning: the ability to form new concepts and to acquire and store new knowledge for future application. Sometimes it involves creating a new form of representation, as has happened often in the history of science and mathematics.

Laziness motivates a desire for generality -- finding one solution for a wide range of cases can save the effort of generating new solutions. This is one of the major motivations for all kinds of scientific research. It can also lead to errors of over-generalisation, prejudice, and the like. A more complete survey would discuss the differences between avoiding mental work (saving computational resources) and avoiding physical work.

-- Sub areas of AI

So far I have given a very general characterisation of intelligence and the goals of AI. Most work in the field necessarily focuses on a sub-area, and each area has its own literature growing too fast for anyone to keep up with.

The topic can be divided up in a number of ways. One form of division reflects the supposed architecture of an autonomous intelligent system.

Thus people study components like vision, language understanding, memory, planning, learning, motor control, and so on. These include empirical studies of people and other animals as well as exploratory engineering designs.

There are also attempts to address what appear to be general issues, for instance about suitable representational formalisms, inference strategies, search algorithms, or suitable hardware mechanisms to support intelligent systems. A second order debate concerns whether there are any generally useful formalisms or inference engines. Some who oppose the notion argue that different kinds of expertise require their own representations and algorithms, and indeed early attempts to produce general problem solvers showed that they often had a tendency to get bogged down in combinatorial searching.

Until recently computer power has been expensive and scarce, so hardly anybody has been able to do anything about assembling integrated systems. Increasingly, however, we can expect to see attempts to produce robots with a collection of computers working together. This will lead to investigations of different kinds of global architectures for intelligent systems. In particular, whereas most AI systems in the past have been based on a single sequential process, it will increasingly be appropriate for different subsystems to work asynchronously in parallel.

-- A simple architecture

Initially it is to be expected that systems will be designed with the following main components:

(a) Perceptual mechanisms

These mechanisms analyse (e.g. parse) and interpret information taken in by the 'senses' and store the interpretations in a database.

(b) A database of information.

This is not just as a store of facts, for a database can also store procedural information, about how to do things, in a form accessible by planning procedures. It may include both particular facts provided by the senses and generalisations formed over a period of time.

(c) Analysis and interpretation procedures

These are procedures which examine the data provided by the senses, break them up into meaningful chunks, build descriptions, match the descriptions, etc. Analysis involves describing what is presented in the data. Interpretation involves describing something else, possibly lying behind the data, for instance constructing a 3-D description on the basis of 2-D images, or inferring someone's intentions from his actions.

(d) Reasoning procedures.

These use information in the database to derive further information which can also be stored in the database. For instance if a lot of information about lines is in the database, inference procedures can work out where there are junctions. If you know that Socrates is a man, and that all men are mortal, you can infer something new about Socrates.

(e) A database of goals.

These just represent possible situations which it is intended should be made ACTUAL. There may also be policies, preferences ideals, and the like.

In the case of real-time intelligent systems further complexities arise from the need to be able to deal with new information and new goals by interrupting, modifying, temporarily suspending, or aborting current processes. I believe that these are the kinds of requirements that explain some kinds of emotional states in human beings, and we can expect similar states in intelligent machines.

It is possible that full replication and understanding of the types of intelligence found in people (and other animals) will require the development of new physical designs for computers. Already there is work investigating highly parallel "connectionist" architectures loosely modelled on current theories about the brain as an assembly of richly interconnected neurons that compute by exciting and inhibiting one another. Such machines might be specially useful for long term associative memory stores, and for low level sensory processing. However, the hardest problem will be knowing how to 'program' such machines.

It may also turn out that we need to discover entirely new kinds of formalisms or representations. For example, at present it is very hard to give machines a good grasp of spatial structures and relationships of kinds that we meet in everyday natural environments. It isn't too difficult for a computer to represent a shape bounded entirely by plane or simply curved surfaces. But we, and other animals, have visual systems without that restriction. Similar comments apply to the representation of motion, e.g. in a ballet, or the non-rigid transformations of a woollen jumper as you take it out of a drawer and put it on.

-- Less ambitious projects

Much AI work is concerned with subsystems of an intelligent system, rather than trying to design a complete autonomous intelligent robot.

In most cases the hardest problems involve identifying the knowledge that is required to perform a task, and finding good ways to represent it. As already hinted, in vision there is a largely unsolved problem of representing shapes and motion in sufficient generality to accommodate the range of objects we all perceive effortlessly. In designing speech understanding systems a key question is what features in the acoustic signal are significant in identifying the meaningful units in utterances. In designing fault diagnosis systems it is often extremely difficult to identify the clues actually used by an expert, the inference strategies used in drawing conclusions from the clues, and the control strategies used in deciding what to do next when the problem is difficult. The difficulties are compounded when the expert needs to be able to combine different sorts of knowledge in a new way, for example knowledge about electrical properties of components, the mechanical and spatial properties, the thermal properties, and the functional design of the system.

One reason these tasks are so difficult is that much human expertise is below the level of consciousness. People are quite unable simply to write down the grammatical rules they use in generating and understanding their native language, despite many years of use. The same applies to most areas of human expertise, though paradoxically it is the most advanced and specialised forms, usually learnt late in life, that are easiest to articulate. This is often partly because they are less rich and complex than more common and superficially impressive abilities shared by all and sundry. This has led to techniques for 'knowledge elicitation', a process that often has much in common with methods by which philosophers probe hidden assumptions underlying our conceptual

systems. (See below.)

For those who wish to apply AI in such a way as to avoid these difficult research issues, it is generally advisable to tackle much simpler problems, for example fault-diagnosis problems where there is already a lot of clearly articulated reliable information on how to track down the causes of malfunctions.

-- Key ideas in AI models

Several important concepts and techniques keep cropping up in work in AI, including the following:

- (a) Structural description (e.g. list, database). This generally depends on analysis of a structure, e.g. Segmenting and recognising parts, properties and relationships, which may then be described.
- (b) Matching (e.g. see the TEACH *MATCHES and *SCHEMATA files.)
- (c) Canonical form (to simplify matching, searching in database, etc.). An example is trying to represent seen objects in terms of their internal structure rather than in terms of their appearance from one viewpoint.
- (d) Domain (a class of structures, with its laws of well-formedness). E.g sentences of English form a domain, logical proofs form a domain, three-D polyhedra form a domain. 2-D line drawings form a domain. Domains can overlap, and one can include another.
- (f) Interpretation of a structure (building another structure which it is taken to represent). For instance interpreting a 2-D image by building a description of the 3-D scene depicted.
- (g) A search space. The structure of a class of problems and possible solutions to those problems is often thought of geometrically.
- (h) Search strategy (controlling search).
- (i) Inference. (Deduction, reasoning.)
- (j) Alternative representations of the same thing (e.g. turtle picture vs database).
- (k) Indexing and addressing. E.g. how do you recognise and complete the following so quickly 'A ---- in time saves ----', 'Birds do it, bees do it, even -----', etc., when you have hundreds of thousands, probably millions of items of stored information in your mind. It can't be that you search LINEARLY through the lot.
- (l) Structure sharing. This is a very important and general notion which can be found in recognition processes, problem-solving and planning processes, inference processes, etc. The basic idea is that if different alternatives have something in common, you should not have to repeat the exploration of the common parts. This can considerably reduce the amount of backtracking required in a search process, for instance. (TEACH VIEWS describes a package that uses structure sharing.)
- (m) Heuristic evaluation and search-pruning procedures.
- (n) The transition from matching to inference. A search for a good match can often be CONTROLLED in part by restrictions on the

variables, e.g. pattern elements like:

??X:NP

where the procedure NP checks that what is matched against X is a noun-phrase. (LIB GRAMMAR uses MATCH in this way). In general, a process which we would ordinarily call matching, for instance matching a 3-D scene against a 2-D image may include a great deal of inference, in addition to checks for correspondences between parts. An extreme case would be the notion of matching a GOAL against a PLAN to achieve the goal. The notion of 'match' here has been considerably stretched. How does one check that a plan will or can achieve a goal? One of the current debates in AI concerning the importance of what are called 'SCRIPTS' or 'FRAMES' can be interpreted as being concerned with the issue that inference can be kept to a minimum during much of the matching required for perception, understanding, planning.

- (o) Trade-offs. Closely connected with several of the previously mentioned ideas is the idea of a trade-off. By doing more work at the time you build up a structure you may be able to use it later with less effort: e.g. a trade-off between compile time and execution time. Converting descriptions into a 'canonical' form to simplify matching and recognition is an example. A more familiar trade-off concerns time against space. Another is generality or flexibility against efficiency.

Does the transition from Roman to Arabic numerals involve a trade-off, or is it pure gain? What about using a new symbol for every word, versus building words out of simpler symbols?

-- Computers vs brains

Whether or not the model sketched above is accurate, concepts like these, which have proved essential for exploring the model, may also be essential for developing correct theories about how the mind works. This may be so even if the human mind is embodied in a physical system whose fundamental computational architecture is very different from a modern digital computer: e.g. it seems to be more like a huge network of communicating computers each connected to thousands of others in the net.

Computer models like this are sometimes called "connectionist" models.

-- "Non-cognitive" (?) states and processes

One of the standard objections to AI is that although it may say something useful about COGNITIVE processes, such as perception, inference and planning, it says nothing about other aspects of mind such as motivation and emotions. In particular, AI programs tend to be given a single 'top-level' goal, and everything they do is subservient to this, whereas people have a large number of different wishes, likes, dislikes, hopes, fears, principles, ambitions, all of which can interact with the processes of deciding and planning, and even such processes as seeing physical objects or understanding a sentence. This is correct and important.

There are ways of extending the model so as to begin to cope with this sort of complexity, without leaving a computational framework. For example, what sorts of processes can produce new motives? How would motives be represented? What sorts of processes could select motives for action? How would one motive (e.g. a fear or preference) interact with

the process of trying to achieve another? In order to answer these questions we must clarify what we understand by the key terms. This requires conceptual analysis.

-- Conceptual analysis

This involves taking familiar concepts, like 'knowledge', 'belief', 'explanation', 'anger', and exploring their structure. What sorts of things can they be applied to, how are they related to other concepts, and what is their role in our thinking and communication? To meet the above criticism of AI in full, it is necessary to engage in extensive analysis of many concepts which refer to mental states and processes of kinds which AI work does not at present say much about, concepts like 'want', 'like', 'enjoy', 'prefer', 'intend', 'afraid', 'sad', 'pleasure', 'pain', 'embarrassed', 'disgusted', 'exultation', and the like.

This is not an easy task, since we are largely unconscious of how our own concepts work. However, by showing how motives of many kinds might co-exist in a single system, generating many different kinds of processes, some of which disturb or disrupt others, we may begin to see how, for example, emotional states might be accounted for. This would require considerable extension of the model outlined above, and would make use of concepts used not so much in AI work as in computer science, especially in the design of operating systems, for instance concepts like 'interrupt', 'priority' and 'communication between concurrent processes'. But such modelling is still some way off.

-- Tools for AI

Anyone who has spent much time programming will appreciate that getting computers to perform AI tasks is not easy. Moreover, most of the widely used programming languages were not designed for this sort of purpose, and the programming support tools, such as editors, compilers and debuggers, are not adequate for projects that are not concerned with implementing well-understood algorithms worked out in advance on the basis of mathematical analysis.

AI development work requires languages that support a wide range of representations including things like verbal descriptions, logical rules of inference, plans, definitions of concepts, images and speech wave-forms. This requires the use of languages that make it easy to build and manipulate non-numerical as well as numerical structures. Examples of such highly expressive languages are LISP, the oldest AI language, Prolog, a language based on logical inference, and POP-11, developed first at Edinburgh University (as POP-2) then at Sussex. POP-11 has the power of LISP but a far more readable syntax and a range of additional features.

Moreover, since the process of building a program is often a tentative exploratory task, part of whose goal is to find out precisely what the constraints and requirements for the program are, it is necessary to provide languages and compilers that support 'rapid prototyping' and very flexible experimentation. Compilers for conventional languages such as C, Ada, Fortran, Pascal, for example, do not allow you to define new experimental procedures or modify old ones, without re-linking the whole system, which can be very slow and wasteful of human and computer time if the system is already big. So AI development tools include interpreters and incremental compilers and editors that are linked in with the compilers so that there is no need for continual switching between the two. The best development environments for LISP, Prolog and POP-11 provide such integrated support tools.

-- An example of the expressive power of an AI language

I'll give one example to illustrate the kind of thing that AI languages provide to simplify programming tasks. Suppose you have to store lists of lists of words and for some reason need a program to find a sublist containing a pair of given words and produce a list of the words in between. For example given the pair of words "cat" "horse" and the list of lists:

```
[[book cat chair spoon][ape cat dog flea horse shark][castle house tower]]
```

it should produce the list: [dog flea]. Writing a program like this in a language like C or PASCAL would require the use of three nested loops and rather complicated constructs for back-tracking if you find a false clue like "cat" in the first list. The POP-11 a pattern matcher enables you to write a single line instruction:

```
list_of_lists --> [== [== cat ??wanted horse ==] ==]
```

(or a more general form replacing "cat" and "horse" with variables), to solve this problem.

Having expressive constructs tailored to the requirements of the task enables programmers to get things right first time far more often. This is one reason why many AI systems include "macro" facilities for extending the syntax of the language to suit new applications. Similarly it is often useful to try one method to solve a task and if that fails try others, where each method itself involves trial and error strategies. Programming this back-tracking control structure yourself is tedious, and you may not do it efficiently, whereas Prolog provides a very general form of it built in to the language.

-- Horses for courses: multi-language, multi-paradigm systems

Which language is best for AI? This is a misguided question. Different languages are needed for different problems or different sub-problems, and for that reason a good AI development environment should make a range of languages available in such a way as to make it easy to integrate programs written in different styles. Also, even if one language is ideal for a particular project, it may be that there is software readily available in another language. Duplicating the development could be very wasteful. So a system that makes it easy to link in a program written in another language is desirable.

POPLOG attempts to meet this requirement. It includes all three of the languages mentioned above, all incrementally compiled into a common portable "virtual machine", which runs on a range of computers and operating systems (in 1986 these are: VMS, UNIX System V, Berkeley UNIX 4.2, on VAX, DEC 8000 series, Hewlett-Packard 9000/200 and 900/300, SUN-2, SUN-3, Bleasdale, GEC-63, Apollo Domain - and probably more later). It also allows programs written in conventional languages to be linked in and unlinked dynamically, and provides facilities for developing new special-purpose sub-languages suited to particular sub-tasks. (The detailed mechanisms are described in REF *SYSCOMPIL and REF *VMCODE. The Alvey Real-time Expert Systems Club, for example made good use of this language-extension facility, which is also used to implement all the POPLOG Languages.

It is very likely that other systems will become available offering some or all of the POPLOG features. Already there are some LISP systems that include a PROLOG subset. POPLOG itself is being used in many countries

including the UK, the USA, Scandinavia, Europe, India, Japan and Australia. E.g. it the core teaching system in a Masters degree in the University of New South Wales.

-- Conclusion

This is by no means a complete overview of AI and its tools. At best I hope I have whetted the appetites of those for whom it is a new topic. The bibliography includes pointers to books and papers that extend the points made in this article.

As readers may have discerned, my own interests are mainly in the use of AI to explore philosophical and psychological problems about the nature of the human mind, by designing and testing models of human abilities, analysing the architectures, representations and inferences required, and so on. These are long term problems.

In the short run, my guess is that the most important practical applications will be in the design of relatively simple expert systems, and in the use of AI tools for non-AI programming, since the advantages of such tools are not restricted to AI projects. In principle, AI languages and tools could also have a profound effect on teaching by making new kinds of powerful teaching and learning environments available, giving pupils a chance to explore a very wide range of subjects by playing with or building appropriate programs. But since our culture does not attach much importance to education as an end in itself, I fear that this potential will not be realised. Instead millions will be spent on military applications of AI.

-- Bibliography

- R. Barrett, A. Ramsay and A. Sloman POP-11: A Practical Language for AI, Ellis Horwood and John Wiley, 1985, reprinted 1986.
- Margaret Boden, Artificial Intelligence and Natural Man, Harvester press, 1977.
- E. Charniak and D. McDermott, Introduction to Artificial Intelligence, Addison Wesley, 1985.
- William S. Clocksin and C.S. Mellish, Programming in Prolog, Springer-Verlag, 1981
- John Gibson, 'POP-11: an AI Programming Language' in Yazdani 1984.
- David Marr, Vision, Freeman 1982.
- Tim O'Shea and Marc Eisenstadt, editors: Artificial Intelligence: Tools Techniques Applications, Harper and Row, 1984.
- Allan Ramsay and Rosalind Barrett, AI in practice: examples in POP-11 Ellis Horwood and John Wiley, forthcoming 1987.
- Elaine Rich, Artificial Intelligence, McGraw Hill, 1983.
- A. Sloman The Computer Revolution in Philosophy, Humanities Press and Harvester Press, 1978.
- A. Sloman, 'Why we need many knowledge representation formalisms', in Research and Development in Expert Systems, ed M. Bramer, Cambridge University Press, 1985.
- A. Sloman, 'Real-time multiple-motive expert systems' in Martin Merry (ed), Expert Systems 85 Cambridge University Press, 1985
- A. Sloman and Graham Thwaites, 'POPLOG: a unique collaboration' in Alvey News, June 1986.
- G J Sussman, A Computational Model of Skill Acquisition, American Elsevier, 1975
- P.H. Winston, and B.K. Horn, LISP, Addison-Wesley, 1981.
- Terry Winograd, Language as a cognitive process: syntax,

Addison Wesley, 1983.

Patrick H. Winston, *Artificial Intelligence*,
Second Edition, Addison-Wesley, 1984.

Masoud Yazdani, editor, *New Horizons in Educational Computing*,
Ellis Horwood and John Wiley, 1984.