For BCS Conference on Expert Systems
December 1985

REAL TIME MULTIPLE-MOTIVE EXPERT SYSTEMS

Aaron Sloman [*]
Cognitive Studies Programme, University of Sussex,
Brighton, BN1 9QN, England

May 8, 2004

Abstract

Sooner or later attempts will be made to design systems capable of dealing with a steady flow of sensor data and messages, where actions have to be selected on the basis of multiple, not necessarily consistent, motives, and where new information may require substantial re-evaluation of plans and strategies, including suspension of current actions. Where the world is not always friendly, and events move quickly, decisions will often have to be made which are time-critical. The requirements for this sort of system are not clear, but it is clear that they will require global architectures very different from present expert systems or even most AI programs. This paper attempts to analyse some of the requirements, especially the role of macroscopic parallelism and the implications of interrupts. It is assumed that the problems of designing various components of such a system will be solved, e.g. visual perception, memory, inference, planning, language understanding, plan execution, etc. This paper is about some of the problems of putting them together, especially perception, decision-making, planning and plan-execution systems.

Introduction: designing intelligent actors

We already know something about how intelligent systems may be able to select or construct plans and execute them. Current AI planning systems can cope with a goal given by the user, together with a database of relevant constraints and factual information. A suitable plan may either be retrieved from a database or synthesised using plan formation techniques, whose details need not concern us now. (Standard textbooks on AI describe planning systems, e.g. Boden (1977) Winston (1984). See also Tate (1985) for a recent overview.)

Some systems can also execute plans, for instance in controlling a robot, managing a refinery or playing a game. It is not always possible simply to execute the plan, once it has been chosen or constructed: execution may have to be interleaved with further planning. It has long been known that a plan may have to be changed or gaps filled in during execution, for instance because not all the required information is available initially, or because the assumptions on which the plan was based may not remain true during execution.

There is nothing conceptually difficult about this. If suitable perceptual mechanisms are available, it is possible to insert into a plan at various points instructions to survey the scene and check assumptions or collect new information, after which the planning module may be invoked to modify or extend the current plan if necessary. Let us call this the ‘regular re-assessment’ strategy.

[*] Author now at the School of Computer Science, University of Birmingham, Birmingham, B15 2TT, UK, http://www.cs.bham.ac.uk/~axs

Quite apart from the difficulties of designing perceptual systems, planning mechanisms, inference mechanisms, managing databases of plans, controlling actions, etc., there are several problems with this as a general model of intelligent planning and action. The problems are concerned with the global organisation of the system: its macro-architecture. By analysing the requirements for intelligent action we shall see the need for a number of asynchronous parallel sub-processes with complex interactions between them. The major problems derive from incomplete knowledge, other resource limitations and the need for speed. Further complications arise out of the fact that intelligent systems may have multiple, not necessarily mutually consistent, independent sources of motivation.

The need for concurrent monitoring

The first problem arises because the environment does not buffer its information until we are ready to examine it. The regular re-assessment strategy assumes that new information can be collected at various discrete convenient pre-assigned steps during plan execution, whereas usually there is a continuous, or at least very rapid, flow of potentially relevant new information. If the information is ignored except at points where plans allow for new information to be considered, then information may be lost. This means that (at least conceptually) the perceptual process of absorbing new information and assessing its significance should go on in parallel with plan execution. In fact this is clearly how humans and animals work, as do interrupt-driven computer operating systems.

This will be called ‘concurrent monitoring’. It could be simulated on a fast enough single processor, using polling or interrupts, but that does not affect the conceptual point that logically there are distinct parallel processes: acting and perceiving.

The need to be able to interrupt execution

The perceptual process might simply produce an internal database of information which would be accessed from time to time by the plan execution process, if the plan specified occasions for considering whether relevant new information has arisen. In some cases, however, new information will indicate that it is not worth continuing the current sub-action, since its preconditions are already known to be violated, like finding that the train you planned to catch has been cancelled. Equally the new information may indicate that the whole current plan is misguided, for instance because the conference you were travelling to has been cancelled.

Either way, waiting for the next pre-programmed break in the current plan could involve wasted effort, risky delay, or even worse consequences.

To complicate matters further, in intelligent systems there may be many different co-existing goals, not just one goal provided by the user. For instance, one of the very high priority goals may be to preserve oneself, or other agents from fatal damage. New information acquired by continuous monitoring processes may indicate that there is an urgent need to abandon the current action and take steps to avoid some new danger or pursue some important new opportunity: for instance learning that the field you are about to cross is mined, or discovering that a movie you have long wanted to see is coming to town just after you had planned a trip.

This means that not only should the perceptual processes go on in parallel with current actions, but in addition there needs to be a process of comparing new information with existing long and short term goals and preferences. We could regard this ‘goal evaluation’ process as part of the concurrent monitoring process. However, goal evaluation may need arbitrarily complex reasoning, for instance to work out the implications of some new information for a current goal, and therefore either the goal evaluation needs to run concurrently with both the action and continued monitoring processes, or else, if it interferes with one of the other processes, then it must use rule-of-thumb heuristics which are guaranteed to give a quick (but potentially fallible) result.

We therefore have, at least conceptually, three parallel processes, action (plan execution), monitoring and goal evaluation. The result of the goal evaluation process must be capable of causing the current action to be interrupted (or modified) if necessary.

Systems in which one process is capable of interrupting another whether the second is ready for it or not are anathema to mathematical computer scientists, since it is very difficult to prove properties of such systems. An algorithm cannot be proved to produce a certain result if some portion of it it is capable of being interrupted or redirected at an arbitrary time. My view is that this flexibility is essential for intelligent systems and, if necessary, the analytical techniques will have to be extended rather than restricting the design of programming languages to rule this out. As we'll see, the feasibility of formal analysis of intelligent systems will be further undermined because of the need for speed, which dictates the use of rule-of-thumb heuristics at crucial stages.

Interrupting *vs* modifying

Sometimes new information can generate a need for existing actions not to be abandoned or suspended but to be modified. For example if you discover evidence that the bridge you are walking over is unsafe, you may start treading more lightly and watching out for evidence that it is about to collapse. While asking someone questions in an interview you discover he has recently had a bereavement: you may go on with the questions originally planned, but alter your manner and perhaps some of the phrasing.

Human beings are able to modify execution of a plan without interrupting it. There are many different sorts of modification, for instance speeding up or slowing down the action, changes of style, changes of route, attending to more of the details, choosing sub-actions with more care, using a different sort of monitoring process, moving more smoothly or evenly, using different materials or sub-actions.

Current plan formalisms and plan executing systems will need to be extended to be able to permit such variation in performance of the same action. This paper does not discuss modification of actions in any depth, as the main points we are interested in can be made in connection with suspending or aborting actions. A full investigation would require analysis of different forms of modification of an ongoing action, the circumstances under which they are useful, and the mechanisms which make them possible. (Sloman 1978 chapter 8 includes some examples.) To simplify discussion in the remainder of this paper I shall merely talk about 'interrupting', though in many cases 'interrupting or modifying' would be more appropriate.

Rule-of-thumb strategies

We have just seen an example of a general point. Many of the comparisons, or decisions, which are discussed in this paper are capable in principle of requiring indefinitely lengthy processes of analysis and inference, and possibly also new physical actions in order to collect new information or experiment with tools or plans before making a decision. It will often not be possible to permit such processes to continue at length: a decision or result has to be computed fast, or without disturbing some other ongoing activity. It may then be useful if rules or mechanisms are available which produce a decision quickly, even if it is not always the right decision. In the extreme case, where information is totally inadequate but a decision has to be taken urgently (e.g. whether to take the left or the right fork in order to escape from a dangerous pursuer), it may be better to choose at random than to think about the choice.

In less desperate situations there may be fragmentary evidence or quick calculations which provide some basis for choosing but may yield a wrong conclusion. An example would be a rule of thumb which says ‘If choosing X in other situations has proved successful, then choose X now’. This rule may be useful for a range of problems even though in some cases deeper investigation would show that Y was a better alternative than X. That is not of much use if doing the deeper investigation takes so long that you lose the opportunity to choose either X or Y, e.g. because the delay has enabled something to injure you, or has enabled both X and Y to flee. Even if an alternative more reliable method is quick, it is no use if the only way to apply it is to interrupt something else which would prove disastrous. For instance, the driver of a fast-moving heavy truck should guess whether to take the next turn rather than take her hands off the wheel to consult a road-map.

So the selection of a quick but unreliable method may be enforced by either resource constraints or time limits.

Where it is certain that delay or interruption of an action will be disastrous, it may be best to use a quick method which is not certain to produce the right result in all cases - as long as it sometimes produces a good result. This may be called the ‘Rule-of-thumb’ strategy. We shall see that it may need to be employed by several different components of an intelligent system with time constraints and limited knowledge or resources. This is, in principle, no different from the well-known need for heuristics in planning or problem-solving systems, i.e. rules of thumb which cut down search spaces or processing time, possibly with a loss of soundness or completeness.

Among the key issues in the design of intelligent systems using rule-of-thumb strategies from time to time, is where those strategies should come from, how they can be evaluated, and how their use may be controlled. In some cases, as indicated in the above example, they may be derived by simple inductive inferences from previous experiences. The mechanisms required for this sort of learning may be different from other sorts of learning mechanisms, e.g. mechanisms concerned with formation of new concepts, or creation of new explanatory theories, or development of problem solving skills. (There is a vast philosophical literature on whether any strategy for inductive learning can be rationally defended without circularity. For engineering purposes it may not be necessary to produce a totally general and provably correct solution. The same applies to explaining actual human and animal abilities.)

The need for concurrent priority comparisons

In its simplest form concurrent perception and goal evaluation, together with an interrupt mechanism, allows the receipt of new relevant information to cause the execution process to be interrupted so that new planning may be done, either to achieve a new high priority goal, or in order the better to achieve the original top-level goal.

Both the regular re-assessment model and this concurrent monitoring model assume that when new information shows that some re-planning is required, action can be suspended for as long as is necessary for the re-planning to be completed. In some cases, as we have seen, planning, or re-planning, may require arbitrarily complex and lengthy processes of inference, collecting new information, consulting experts, doing experiments, etc. Often it will not be possible to wait for such a re-planning process to be completed. Death or disaster may come first.

It may not be possible even to start the re-planning process, for instance if new information becomes available during the course of executing some extremely dangerous and difficult sub-operation which cannot temporarily be abandoned or suspended without risk to life and limb.

Alternatively, pausing to consider a new goal or to plan a different strategy may lose forever an opportunity to achieve the current goal, while the other could have waited. So, sometimes, instead of being done quickly using rule-of-thumb methods, a new activity may have to be totally suppressed.

Occasionally, the risk of ignoring the new information will be even greater than the risks of interfering with the current action, for example where the new information concerns some terrible unforeseen danger, or a marvellous unexpected new opportunity. Thus we see that in some cases the current action must not be interrupted on account of new information, whereas in other cases it is more urgent that the re-planning be done.

In principle it would be possible to continue with the current action whilst alternative plans are synthesised and evaluated in parallel, in the light of the new information. This may sometimes be feasible and at least in the case of human beings is often done, for instance if you continue driving along a route whilst considering whether you ought to revise your route in the light of new information about congestion ahead. Planning alternatives in parallel with plan execution makes it unnecessary to interrupt execution.

Sometimes it is not possible to do both in parallel, either because the computational load required for both activities cannot be supported, or because the new planning process requires physical activity in order to collect new information or experiment with new tools, etc. A physical system cannot be in two places at once, and eyes or cameras or hands cannot be directed in two directions at once. Thus computational resource limits and physical resource limits may make it impossible to complete the new planning without interrupting the old action. But, as already indicated, this could be disastrous.

So, although in general actions need to be interruptable and modifiable, some actions should not be disturbed except for more urgent tasks. This suggests that concurrent monitoring needs to be supplemented with a ‘priority comparison’ mechanism which decides whether new information is sufficiently important and urgent to warrant interrupting current plan execution. This could be part of the goal evaluation mechanism, e.g. if every new goal is compared with currently active goals to decide whether they are important enough to generate an interrupt. Alternatively, if new goals are put into some sort of queue, it may be possible for yet another independent process to compare them with existing goals (Croucher 1985).

The need for rule-of-thumb priority comparisons

A problem with any attempt to evaluate new information or to compare new goals with old ones is that it may not be immediately evident whether the new information is or is not more urgent, since assessing the implications of new information, or comparing the implications of two courses of action, may itself be an arbitrarily complex task. If the task requires getting new information, performing elaborate deductions, or in any way using resources required for the current activity, then deciding whether it would be desirable to interrupt the current activity may not be possible without actually interrupting it! This means that the priority comparison mechanism must be capable of running in parallel with the main plan execution, just as perceptual monitoring does, and secondly that it must be capable of taking resource-limited decisions. I.e. where comparing alternatives in detail would interrupt a high priority process, or where concurrently working out which priority is higher would take a long time, there must be a way of doing the comparison quickly, with the inevitable consequent loss of reliability.

An implication of this rule-of-thumb strategy for comparing priorities, is that sometimes the main action may be disturbed by this interrupt mechanism when it should not be. Elsewhere it has been argued that this is one source of emotional states in intelligent systems. (Sloman &

Croucher 1981, Croucher 1985.)

The need for ‘reflex’ responses

I have talked about several processes that may be arbitrarily complex and time-consuming which go on in parallel since none has the right to dominate the rest. Several different forms of decision making are involved. But they are all relatively slow, involving explicit, conscious or unconscious deciding, or planning, or inference. In some cases the need for speed may be met by making the processes use the rule-of-thumb strategy. However, in other cases even this will be too slow. If you put your hand down on a red-hot plate, then by the time you have worked out whether having it burnt is more serious than taking some weight off your legs, your hand will be badly damaged. Avoiding that sort of possibility requires certain sorts of new goals to trigger action directly, without going going through any decision making process. These may be called reflexes.

Reflex triggering can be done in several different ways, including fairly direct electro-mechanical circuits which cause a reaction without any high level processes intervening, or the use of software driven interrupts. We could call these ‘hard’ and ‘soft’ reflexes.

A measuring device may be able to trigger an action as soon as a threshold is exceeded. The human body seems to have many such “hard” reflexes. Other reflexes may be triggered by the addition of some new assertion to a store of beliefs, or even by the formation of a certain description at a low level in the perceptual process. These “soft” reflexes may be the result of learning, or, more precisely, training, like the boxer’s automatic responses to some of his opponent’s movements, or the expert pianist’s sight-reading capabilities, which seem to involve temporarily switching on a set of sophisticated reflexes. This sort of thing could be implemented using a ‘software demon’ mechanism.

The design of a system which allows reflex actions triggered by new information rapidly to take control, poses many problems about how to deal with sudden interruptions of a very large number of different processes at different levels. In existing computing systems it is possible to cause a process to abort completely or to freeze and go into some kind of ‘break’ mode for debugging purposes. But where the interrupt requires a complete new action to be initiated very rapidly, possibly including the control of several physical mechanisms guided by ongoing analysis of sensory input, then significant advances in AI and Computing Science will be needed.

Internal *vs* external actions

The discussion of the need for parallel monitoring and goal evaluation, and the possibility of interrupts and reflex responses, is applicable both to internal and external actions. For example, someone who is confronted with making a difficult choice may have a plan for thinking the problem through and may carry out that plan without performing any external action. Playing chess or trying to remember where you might have lost your bag are examples of internal actions. Monitoring of new information may proceed in parallel with an internal action too: for instance watching the clock while trying to decide. And the internal process may have to be interrupted if some new information turns up, like the discovery that making a decision is urgent whether or not the analysis has been completed. Alternatively, some far more important goal may be triggered by the arrival of new information, such as that your house is on fire.

Internal processing may have to be monitored too. For example, the mathematician thinking about an equation may have to be on the lookout for transformations in which she divides both sides by zero; and a plan synthesiser may have to monitor the consistency of sub-plans or the relations between the plan being constructed and the constraints, such as expense or resources available. If checks are made only whenever a complete plan has been formulated, the search for a good plan is likely to be defeated by a combinatorial explosion.

Meta-level motivators

There is a great deal more to be said about where goals come from and how they are compared. Some are subgoals of a prior goal and are generated by the planning process. Some are responses to new information, such as wanting to know what is happening outside as a result of hearing a loud noise. A full discussion would need to describe various kinds of motive generators, motive generator generators, etc.

Similarly, since not all the goals generated may be mutually consistent, and even the consistent ones may not be simultaneously attainable, various sorts of motive comparators will be required for deciding which of two incompatible goals should be rejected, or which of two compatible goals should be pursued first, etc. Motive comparators may themselves be subject to modification, or new ones may be created in the light of experience by ‘motive comparator generators’.

Research is needed on how to design generally useful motive generators, motive comparators, and higher-level generators which can be thought of as one type of learning system (Sloman & Croucher 1981). Research will also be needed on types of global architecture which will enable such things to function in resource limited systems which need to be able to store and process enormous amounts of information.

Two sorts of parallelism

The kind of parallelism required for the reasons given here may be called macro-parallelism, and distinguished from micro-parallelism of the type supposed to characterise fifth generation computers. Macro parallelism concerns the division into relatively few, possibly quite complex sub-systems which compute in parallel and send one another messages, perhaps interrupting or modifying one another’s behaviour. Each sub-system might be very like a single conventional computer, or it might be a highly parallel system with a micro-parallel architecture. Micro parallelism concerns the construction of very large numbers of processors amongst which any sort of programming task may be shared, for example under a data-flow regime. There is plenty of evidence that the brain uses micro-parallelism, though the processing units seem quite unlike the units employed in computers, including highly parallel computers. What may not be so obvious is that for the reasons we have been analysing, there is a need for macro-parallelism in animal brains too.

My own view is that for the design of truly intelligent systems it is more important for us to understand forms of macro-parallelism than micro-parallelism.

Conclusion

The space of possible computing systems is enormous and very varied, and we have barely begun to explore it. There has been some impressive progress, but overall the achievements of many animals and very young children outstrip even the most sophisticated computing systems of our time. Natural intelligence still shows many sorts of creativity and flexibility unmatched by artefacts. I have tried to draw attention to a type of flexibility which has not received much attention in the AI and Expert Systems literature, and I have pointed towards mechanisms which might help achieve it.

Current research is mostly concerned with the design of relatively simple systems which perform one sort of task – and quite rightly so, since we must learn to walk before we run, or fly. Sooner or later, however, the components will have to be assembled. There may be many different ways of putting things together for different purposes. We have noted some of the constraints relevant to assembling systems which are resource-limited, yet driven by multiple and changing goals in a fast-moving and not always friendly environment.

Some of those constraints point to the need for macro-parallelism and the ability of some sub-processes to interrupt or modify others. Some of the constraints imply that not all important decisions (e.g. whether a current action should be interrupted) can be left to the highest level intelligent processes: at the time such a decision should be taken, the high-level process may have been single-mindedly pursuing a task which does not expose it to the need for reconsideration.

The need for speed or the need to avoid diverting resources from important ongoing activities may enforce a rule-of-thumb strategy for many of the components, so it cannot be expected that such systems will always behave optimally. The experience of AI researchers hitherto suggests that when large knowledge stores and powerful inference mechanisms are available it will always be necessary to use fallible heuristics to enable decisions to be taken quickly. A system with so many fallible components in key positions may be expected to be almost human. The inevitability of familiar types of fallibility should be a matter of concern to those who hope that important decisions can be taken very rapidly by machines in the not too distant future.

Acknowledgements

This work is supported by a fellowship from the GEC Research Laboratories, and a grant from the Rennaisance Trust. The ideas have benefited from discussions over several years with Monica Croucher, and reading her draft thesis.

BIBLIOGRAPHY

Boden, M.A. (1977). *Artificial Intelligence and Natural Man,* Harvester Press and Basic Books.

Croucher, Monica. (1985). *A Computational Approach to Emotions*, Draft Thesis, University of Sussex, 1985

Simon, H.A. (1967). ‘Motivational and Emotional Controls of Cognition’, reprinted in *Models of Thought*, Yale University Press, 1979.

Sloman, A. (1978). *2 The Computer Revolution in Philosophy: Philosophy Science and Models of Mind*, Harvester Press and The Humanities Press.

Sloman, A. and M. Croucher. (1981) ‘Why robots will have emotions’ in *Proc. 7th IJCAI* Vancouver, p. 197-202.

Tate, A. (1985). ‘A review of knowledge-based planning techniques’ in *The Knowledge Review*, vol 1 no 2, British Computer Society.

Winston, P.H. (1984). *Artificial Intelligence,* Second Edition, Addison Wesley.