

To appear in *Communications of the Association of Computing Machinery* 1999

BUILDING COGNITIVELY RICH AGENTS USING THE SIM_AGENT TOOLKIT

Aaron Sloman and Brian Logan

School of Computer Science, University of Birmingham
Birmingham B15 2TT UK

{a.sloman, b.s.logan}@cs.bham.ac.uk

Synthetic agents with varying degrees of intelligence and autonomy are being designed in many research laboratories. The motivations include military training simulations, games and entertainments, educational software, digital personal assistants, software agents managing Internet transactions or purely scientific curiosity.

Different approaches are being explored, including, at one extreme, research on the interactions *between* agents, and at the other extreme research on processes *within* agents.

The first approach focuses on forms of communication, requirements for consistent collaboration, planning of coordinated behaviours to achieve collaborative goals, extensions to logics of action and belief for multiple agents, and types of emergent phenomena when many agents interact, for instance taking routing decisions on a telecommunications network.

The second approach focuses on the *internal* architecture of individual agents required for social interaction, collaborative behaviours, complex decision making, learning, and emergent phenomena within complex agents. Agents with complex internal structure may, for example, combine perception, motive generation, planning, plan execution, execution monitoring, and even emotional reactions.

We expect the second approach to become increasingly important for large multi-agent systems deployed in networked environments, as the level of intelligence required of individual agents increases. This is particularly relevant to work on agents which must cooperate to perform tasks requiring planning, problem solving, learning, opportunistic redirection of plans, and fine judgement, in a partially unpredictable environment. In such contexts, important new information about something other than the current goal can arrive at unexpected times or be found in unexpected contexts, and there is often insufficient time for deliberation. This requires reactive mechanisms. However some tasks involve achieving new types of goals or acting in novel contexts, which may require deliberative mechanisms. Dealing with conflicting goals, or adapting to changing opportunities and cultures may require sophisticated motivational mechanisms.

Motivations for such research include: an interest in modelling human mental functioning (e.g., emotions), a desire for more interesting synthetic agents ('believable agents') in games and computer entertainments, and the need for intelligent agents capable of performing more complex tasks than hitherto.

Many researchers are investigating relatively simple agents performing restricted tasks, such as sorting a manager's incoming email, or giving a novice user information about a software system. Some believe that, in future, a manager in a large company might be replaced by a team of software agents, some attempting to obtain and summarise information about other companies, some monitoring and evaluating activities in subsidiary companies, and others taking strategic decisions.

We believe that cognitively rich internal mechanisms and structures are necessary to model complex communication and planning with (and between) agents. However, these may have unexpected side-effects. It has been argued ([10]) that intelligent robots will need mechanisms for coping with resource limits and that interactions within these mechanisms will sometimes produce emotional states involving partial loss of control of attention, as already happens in humans.

Motivational and emotional processes may also be involved in attempts to persuade a character in an interactive story to adopt a particular belief, or in discussing the relative importance of various tasks with a personal assistant agent. Similar mechanisms may be involved when communication operates at several different levels, for example, when it involves humour, irony or metaphor. Related claims about motivational and emotional requirements for intelligence can be found in [4, 7].

Different kinds of architectures

The internal architecture of an agent can take different forms, with varying degrees of intelligibility and modularity. At one extreme there may be very large numbers of low level components (e.g. neurons) all interacting in such a way that useful global behaviour emerges, without any explicitly programmed intervening structure to account for the behaviour or any architecture we can comprehend. At another extreme agents may have a clearly defined modular architecture with a hierarchical decomposition into separable components performing different tasks, using different though strongly interacting mechanisms to perform those tasks. In between are architectures with varying degrees of functional decomposability and intelligibility.

Another dimension of variation concerns the extent to which the internal architecture is purely 'reactive', where detection of internal or external conditions immediately generates new internal or external responses, which in turn could trigger new reactions (see Figure 1). The internal and external feedback loops and changes of state which alter a reactive agent's responses to new sensory inputs can combine to achieve quite sophisticated behaviour. Examples could include software agents monitoring and reacting to sensor readings in a chemical plant, and controlling complex and varied processes. More sophisticated reactive systems may require processing at different levels of abstraction in the sensory and motor subsystems, e.g. detecting not only low level signals but also more abstract patterns in the data, and triggering complex behaviours involving co-ordination of several effectors. However, this may prove inadequate in some contexts.

Deliberative architectures achieve even greater flexibility by allowing sets of possible actions to be assembled and evaluated without necessarily being selected for execution. A deliberative architecture (which might be implemented using internal reactive mechanisms) requires a large amount of stored knowledge, including explicit knowledge about which actions are possible and relevant in various circumstances, and what the effects of various

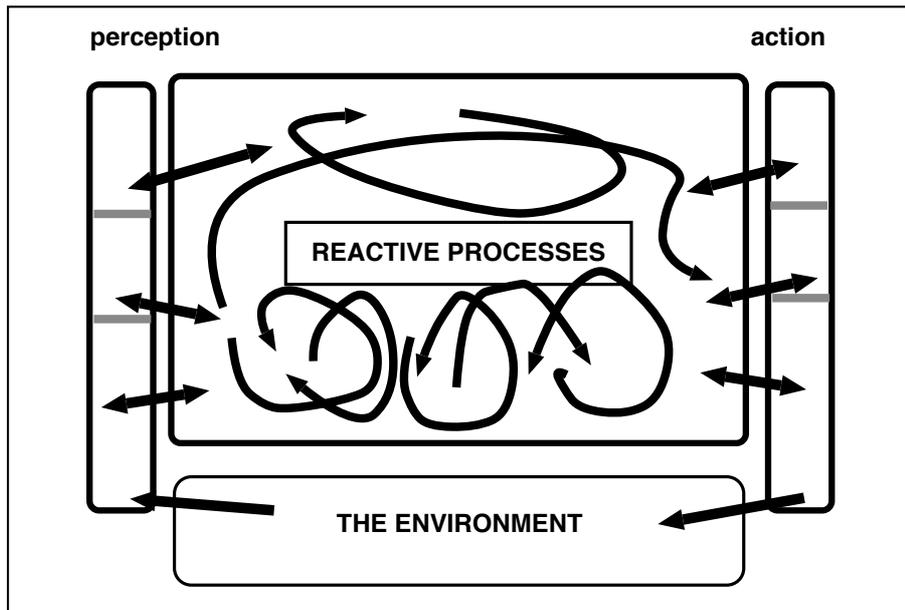


Figure 1: The reactive layer.

actions are in those circumstances. It also requires a re-usable short term memory for building structures representing possible action sequences in order to evaluate their consequences. The reuse of memory, the inability of the long term store to answer many questions in parallel, and the sequential nature of plan construction will typically make a deliberative system much slower than a reactive one. These resource limits might necessitate an attention filter to suppress low priority interruptions. New levels of abstraction in the perceptual and action subsystems may be required to match the more abstract processing in the deliberative system.

Such a deliberative architecture might be required for a personal assistant or a plant control mechanism able to cope in novel circumstances where routine reactions are sometimes inappropriate and where trial and error behaviour is too dangerous or too time consuming.

Any externally observable behaviour that can be produced by a deliberative architecture can, in principle, be produced by a purely reactive architecture, as long as an exhaustive collection of condition-action rules has been assembled in advance by a designer or perhaps by an evolutionary process. Producing a comprehensive set of reactive behaviours in advance may take an inordinately long time and may require excessive amounts of storage.

In deciding between reactive and deliberative architectures in an environment with a high degree of complexity and variability, there are complex trade-offs involving the time and effort required to pre-compile all the potentially relevant rules, the space required to store them, and the mechanisms required for efficiently finding and selecting between potential contenders at run time. Our conjecture is that evolution has solved this problem in natural systems with a *hybrid architecture* involving closely integrated concurrently active deliberative and reactive sub-architectures (see Figure 2).

Such a hybrid architecture may also include mechanisms for transferring some of the results of the deliberative layer to the reactive layer (e.g., responses to stereotypical situations) to improve speed of performance or release the deliberative layer to attend to other things. Many learnt human skills are like this, and similar types of skill acquisition may be needed in software agents and robots.

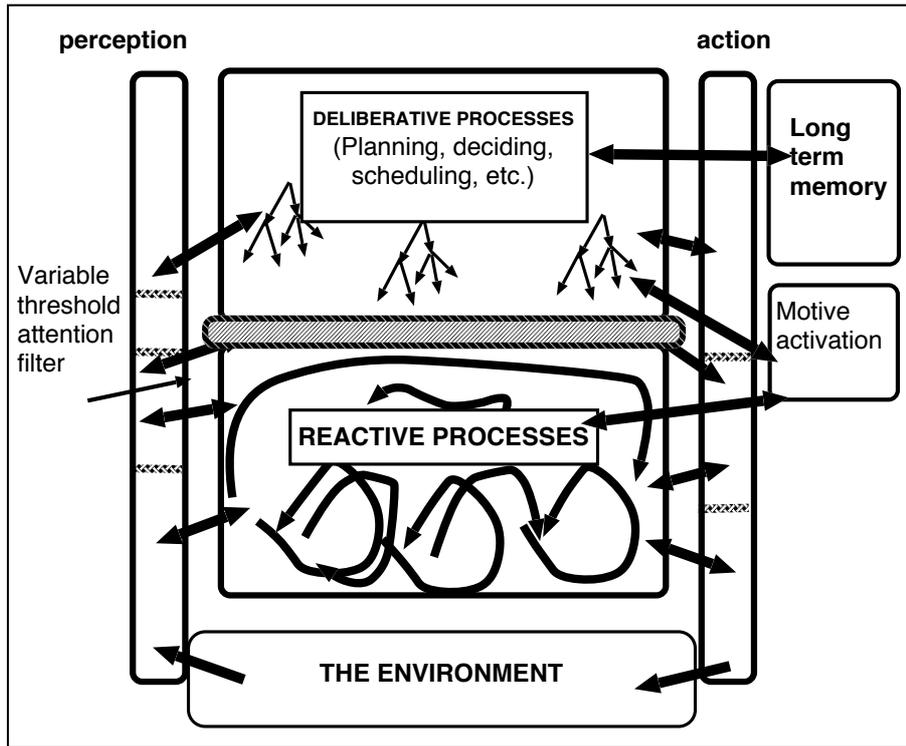


Figure 2: Reactive and deliberative layers.

Self monitoring

Work on multi-agent systems often assumes that interacting agents are perfectly rational or have a fixed collection of motives (i.e., goals, preferences and standards), requiring only simple cognitive mechanisms such as payoff matrices or utility functions. While this may be appropriate for systems consisting of simple agents with ant-like cooperation, it is inadequate for societies of intelligent human-like software agents or autonomous robots. In such situations the set of motives is not fixed, as new motives can be generated in new contexts, for instance the goal of helping a friend who is in trouble, or having to deal with unexpected conflicts between high level objectives and standards or preferences. Realistic multi-agent systems will have to take account of these internal within-agent conflicts as well as between-agent conflicts and negotiations. For example, a plant management software team may encounter conflicts between safety of human workers and economy of plant operation. Simulated battlefield commanders or simulated anti-terrorist strategists may have to detect and handle conflicts between protecting civilians and capturing opponents.

In these contexts an agent may benefit from an additional architectural layer, with the ability to monitor, evaluate, and possibly modify internal processes of various kinds (see Figure 3). This could be based on a mixture of internal reactive and deliberative processes directed at the agent's internal sensory buffers, its goals and preferences, deliberative strategies, records of recent decisions, and the like. Such a system might discover conflicts of motivation and devise a strategy for dealing with them. Or it may notice that certain problem solving processes are taking too long, so that a less cautious but more speedy strategy is required.

We expect that some synthetic agents, including agents working in groups or teams, will

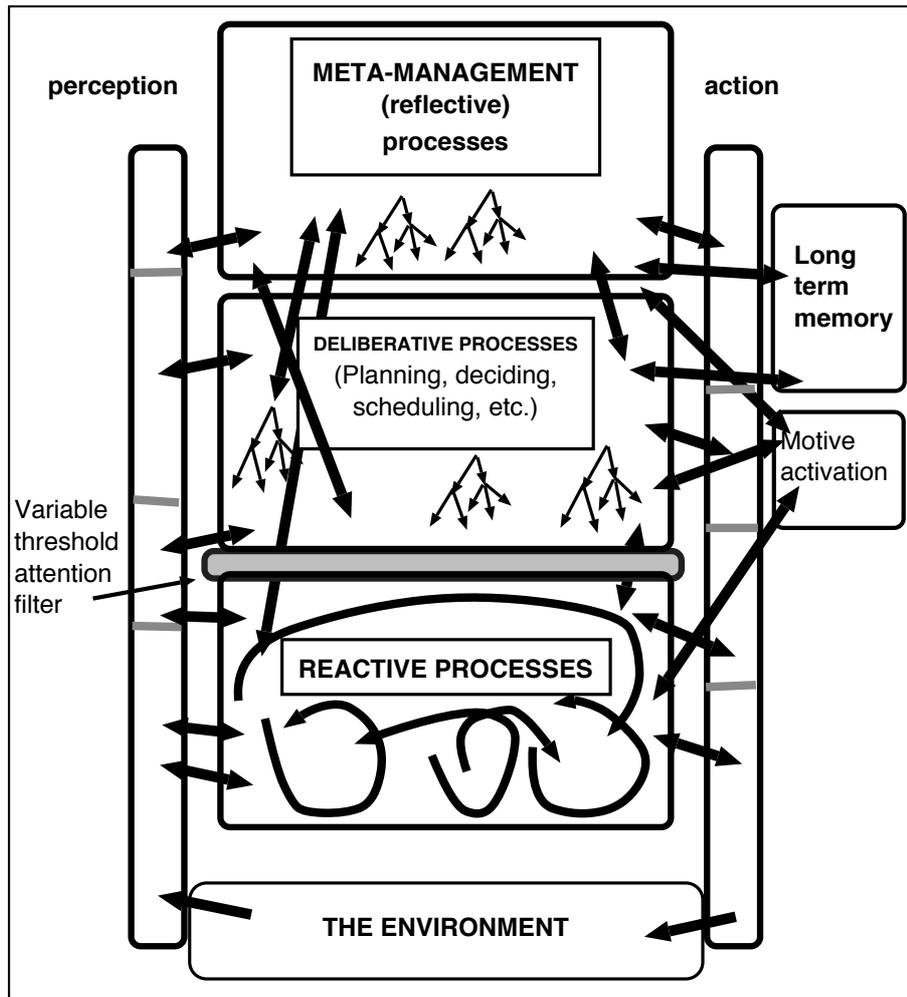


Figure 3: The meta-management layer.

require a similar variety of internal processing. For instance meta-management is required to decide when to stop trying to solve a problem and instead ask for help.

In humans this ‘meta-management’ capability seems to be used for a variety of purposes including social control via inculcation of ethical and other standards of self-assessment, and also some kinds of learning in which deliberative processes are evaluated, found wanting and improved. It also seems to be impaired during times of stress or in certain emotional disturbances.

Some of the internal processes, including both deliberative processes and reactive processes where chains of associations are needed to solve a problem, may be too slow for an environment where opportunities and dangers requiring immediate action can turn up unexpectedly. A *global alarm system* using fast pattern-recognition mechanisms able to trigger stereotyped internal and external responses could help in dealing with such situations. The need for rapid reactions could make such a mechanism sometimes produce erroneous responses. This trade-off between speed and intelligence can be found in some human emotions and may also turn up in synthetic agents. A trainable alarm system might reduce the frequency of mistakes.

Multi-processing within agents

The previous discussion leads to the conclusion that sophisticated agents will include many concurrent processes performing various kinds of tasks such as:

- analysis and interpretation of sensory data at different levels of abstraction,
- monitoring for ‘alarm’ conditions sometimes producing rapid global re-direction of processing,
- generation of new motives,
- assessment of motives and deciding whether to adopt them,
- construction of plans for achieving motives (either after deciding to adopt them or as part of the process of deciding whether to),
- deciding which currently adopted plans should be executed now, which postponed, which abandoned, etc.,
- learning new patterns and categories, evaluating past performance,
- attempting to interpret the observed behaviour of other agents,
- understanding explicit (and implicit) communications from others,
- deciding what to say to others and how to say it,

and much besides.

In addition to relatively high level processes, a robot with a complex physical body, like an animal, may also require a large number of reactive processes monitoring and controlling various aspects of bodily function, posture, gait, fine control of movement of hands, etc. Software agents dealing with high volume information streams requiring fast decisions may also need a well-trained reactive layer.

Multiple programming paradigms: the SIM_AGENT toolkit

For some years the Cognition and Affect project at the University of Birmingham has been exploring the problems sketched above [2, 9]. To support this research we designed and implemented a software toolkit, SIM_AGENT, to allow exploration of ideas through implementation using rapid prototyping [8, 11]. We found it necessary to provide support for a variety of programming paradigms, in addition to widely used object oriented techniques.

For instance, the need for flexible reactive internal mechanisms within agents is supported by a rule-based programming style, where diverse mechanisms within an agent are implemented in different condition-action rulesets.

To allow a wide range of conditions to trigger internal processes and in order to support sophisticated internal reasoning, a list-processing paradigm is used to implement the internal databases and communication channels between parts of an agent and between agents.

To support ‘sub-symbolic’ processing the rule-based system has interfaces to procedural programs and both conditions and actions can be linked to mechanisms like neural nets where appropriate.

The need for fast low-level processes is met by using compiled code. More flexible processes subject to self-monitoring can use interpreted internal languages. This also allows us to vary the relative speeds of different components, to explore hypotheses about resource limited systems. The presence of an incremental compiler allows rapid development and testing, as well as supporting easy experimentation by students in a teaching environment. Use of indirect references allows rules to be redefined easily or procedures to be recompiled during a pause in an experimental run of the system. Automatic store management and garbage collection prevents memory leaks.

The implementation language is Pop-11, a sophisticated extendable lisp-like language with a Pascal-like syntax, in the Poplog environment. (Pop-11 is also at the heart of Clementine, a commercially successful data-mining product.)

The toolkit allows construction of sets of agents where each agent has a multi-component architecture, in which different components operate in parallel, using different kinds of mechanisms. These can include neural nets or other reactive mechanisms alongside AI symbol manipulating mechanisms, used for solving complex planning problems, or for certain meta-management tasks. For example, a system may use a neural net to obtain evidence of fraud in credit transactions based on subtle combinations of data, together with deliberative mechanisms to analyse suspect cases in greater detail. A factory automation system could use a set of reactive procedures to handle routine management and a deliberative planner to handle novel problems, such as the production of a new product. A meta-management layer could decide whether to change the criteria being used by the planner, or whether to suspend planning and begin acting because of shortage of time or because of the need to collect more information by acting on a partial plan.

Many agent toolkits are based on a commitment to a particular architecture, e.g. SOAR [5]. Because SIM_AGENT has no such commitment it has proved attractive to researchers and students wishing to explore alternative designs in a variety of fields, including simple computer games, battlefield simulations in training software (at DERA Malvern, UK), modelling aspects of human cognition and emotions [3, 12], agents planning in a dynamic domain [6] and telecommunications. The ease of development in an AI environment makes it particularly useful for teaching, using simple demonstrations which students can modify.

Planned extensions, arising out of our recent work on meta-management tasks, include improved facilities for self-monitoring and self-modification by agents. Recent work on requirements for agent toolkits is reported in [1].

Conclusion

The trade-offs between different types of architectures are not clear and much research is still required. Although more efficient toolkits dedicated to particular types of agent architectures will be more appropriate in tasks where the architectural requirements are narrowly specified in advance, we expect that increasingly toolkits with the flexibility of SIM_AGENT will be needed as research focuses more strongly on agents with more sophisticated multi-functional architectures, which raise many of the hardest unsolved problems in AI. Currently these are ignored in much multi-agent systems research, but that cannot last.

Further information and availability

The toolkit and additional information about it are freely available via FTP and WWW. A brief overview with some MPEG movies can be found at:

http://www.cs.bham.ac.uk/~axs/cog_affect/sim_agent.html

The Cognition and Affect FTP repository of theoretical papers is at

ftp://ftp.cs.bham.ac.uk/pub/groups/cog_affect/

The Birmingham Poplog directory is at the following URL, where the README file lists information about SIM_AGENT and its supporting libraries.

<ftp://ftp.cs.bham.ac.uk/pub/dist/poplog>

SIM_AGENT requires Poplog, available from Integral Solutions Ltd.

<http://www.isl.co.uk/>

References

- [1] Jeremy Baxter and Brian Logan, editors. *Proceedings AAAI-98 Workshop on Software Tools for Developing Agents*, Menlo Park, California, 1998. American Association for Artificial Intelligence.
- [2] L.P. Beaudoin and A. Sloman. A study of motive processing and attention. In A. Sloman, D. Hogg, G. Humphreys, D. Partridge, and A. Ramsay, editors, *Prospects for Artificial Intelligence*, pages 229–238. IOS Press, Amsterdam, 1993.
- [3] Darryl N Davis. Reactive and motivational agents: Towards a collective minder. In J.P. Mueller, M.J. Wooldridge, and N.R. Jennings, editors, *Intelligent Agents III – Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag, 1996.
- [4] Daniel Goleman. *Emotional Intelligence: Why It Can Matter More than IQ*. Bloomsbury Publishing, London, 1996.
- [5] J. E. Laird, A. Newell, and P. S. Rosenbloom. SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64, 1987.
- [6] B.S. Logan and N. Alechina. A* with bounded costs. In *Proceedings of the 15th National Conference on Artificial Intelligence–AAAI-98*, 1998. Also Birmingham School of Computer Science technical report CSRP-98-09.
- [7] Rosalind Picard. *Affective Computing*. MIT Press, Cambridge, Mass, London, England, 1997.
- [8] R. Poli, M. Brayshaw, and A. Sloman. A hybrid rule-based system with rule-refinement mechanisms,. In *Proceedings Expert Systems'95 Conference, Cambridge UK*. British Computer Society Expert Systems Group, 1995.

- [9] A. Sloman. What sort of control system is able to have a personality. In Robert Trappl and Paolo Petta, editors, *Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents*, pages 166–208. Springer (Lecture Notes in AI), Berlin, 1997.
- [10] A. Sloman and M. Croucher. Why robots will have emotions. In *Proc 7th Int. Joint Conference on AI*, pages 197–202, Vancouver, 1981.
- [11] A. Sloman and R. Poli. Sim_agent: A toolkit for exploring agent designs. In Mike Wooldridge, Joerg Mueller, and Milind Tambe, editors, *Intelligent Agents Vol II (ATAL-95)*, pages 392–407. Springer-Verlag, 1996.
- [12] Ian P Wright and Aaron Sloman. Minder1: An implementation of a protoemotional agent architecture. Technical Report CSRP-97-1, The University of Birmingham, School of Computer Science, 1977.