Modeling Cognition with Software Agents

Stan Franklin (franklin@memphis.edu)
Institute for Intelligent Systems, The University of Memphis, Memphis TN 38152, US
Art Graesser (a-graesser@memphis.edu)
Institute for Intelligent Systems, The University of Memphis, Memphis TN 38152, US

Abstract

We propose the use of autonomous software agents as cognitive models that generate testable hypotheses about human cognition. While such agents are typically produced to automate practical human tasks, they can be designed within the constraints of a psychological theory. As an example we describe an agent designed within global workspace theory that accommodates several other theories as well. We discuss various resulting hypotheses, including a new interpretation of the readiness potential data of Libet.

Introduction

Computational models have long been a major, and perhaps indispensable, tool in cognitive science. Many of these model some psychological theory of a particular aspect of cognition, attempting to account for experimental data. Others aspire to be a general computational model of cognition, such as the construction-integration model (Kintsch 1998), SOAR (Laird et al. 1987), and ACT-R (Anderson 1990). Most of these computational models are computer simulations of subjects in psychological laboratories, and are capable of performing tasks at a fine-grain level of detail. The simulated data ideally fit the human data like a glove. The theories on which the simulations are based are periodically revised so that new simulations conform more closely to the data. The computational models are judged on how closely they predict the data. A model may be judged by the amount of change required in core, as opposed to peripheral, parameters that are needed to fit the data. Alternatively, the models are evaluated on a course-grain level, by observing whether a number of qualitative predictions (i.e., directional predications, such as condition A > B) fit the data. And finally, all of the models have been evaluated by observing how well they fit data in practical, everyday tasks in real-world environments. For example, some such agents, based on SOAR, simulate battlefield performers such as fighter pilots and tank commanders (Hirst & Kalus 1998). These data fitting approaches to testing theories have been hugely successful, and account for a large body of what is now known in cognitive science.

In this paper, we propose another class of computational models, which fall under the general heading of autonomous software agents (Franklin & Graesser 1997). These agents are designed to implement a theory of cognition and attempt to automate practical tasks typically performed by humans. We have been developing two such agents that implement global workspace theory Baars 1988), one with a relatively simple clerical task (Zhang et al. 1998b) and the other with a rather complex personnel assignment task (Franklin et al. 1998). These models do not merely produce output that solves a specific engineering problem, as do typical software agents like web bots. They have mechanisms that simulate human cognition and their design decisions generate hopefully testable hypotheses (Franklin 1997), thus potentially providing research direction for cognitive scientists and neuroscientists.

This paper briefly describes the architecture and mechanisms of one such agent. In Table 1 we point out examples of relevant hypotheses that arise from our design decisions. It is beyond the scope of this article to specify all of the hypotheses and associated support.

Theoretical Frameworks

According to *global workspace (GW) theory* (Baars 1988), one principal function of consciousness is to recruit the relevant resources needed for dealing with novel or problematic situations. These resources may include both knowledge and procedures. They are recruited internally, but partially driven by stimulus input. GW theory postulates that human cognition is implemented by a multitude of relatively small, special purpose processes, almost always unconscious. Communication between them is rare since they mostly communicate through working memory and over a narrow bandwidth. They are individually quite simple and incapable of dealing with complex messages. Coalitions of such processes find their way into a global workspace. This limited capacity workspace serves to broadcast the message of the coalition to all the unconscious processors (bringing it to consciousness) in order to recruit relevant processors to join in handling the current novel situation, or in solving the current problem. Thus consciousness allows us to deal with novel or problematic situations that cannot be dealt with efficiently, if at all, by habituated unconscious processes. Consciousness recruits appropriately useful resources, and thereby manages to solve the relevance problem.

An *autonomous agent* (Franklin & Graesser 1997) is a system situated in, and part of, an environment, which senses that environment, and acts on it, over time, in pursuit of its own agenda. In biological agents, this agenda arises from drives that evolve over generations; in artificial agents its designer builds in the drives. Such drives which act as motive generators (Sloman 1987) must be present, whether explicitly represented or derived from the processing trajectory. The agent also acts in such a way as to possibly influence what it senses at a later time. In other words, it is structurally coupled to its environment (Maturana 1975). Biological examples of autonomous agents include humans and most animals. Non-biological examples include some mobile robots, and various computational agents, including artificial life agents, software agents and many computer viruses. We are immediately concerned with autonomous software agents, designed for specific tasks, and ‘living’ in real world computing systems such as operating systems, databases, or networks.

A *“conscious” software agent* is one that implements GW theory. In addition to modeling this theory (Franklin & Graesser 1999), such “conscious” software agents should be capable of more adaptive, more human-like operations, including being capable of creative problem solving in the face of novel and unexpected situations. However, there is no claim that the agent is a sentient being. What the agent truly feels or what the conscious experience actually is are not the relevant concerns.

IDA is one recent instantiation of our theory of intelligent software agents. IDA (Intelligent Distribution Agent) is a “conscious” software agent being developed for the US Navy (Franklin et al. 1998). At the end of each sailor's tour of duty, the sailor is assigned to a new billet. This assignment process is called distribution. The Navy employs some 280 people, called detailers, to effect these new assignments. IDA's task is to facilitate this process by completely automating the role of detailer. IDA must communicate with sailors via email in natural language, by understanding the content and producing life-like responses. Sometimes she will initiate conversations. She must access several databases, again understanding the content. She must see that the Navy's needs are satisfied by adhering to some ninety policies. She must hold down moving costs, but also cater to the needs and desires of the sailor as well as is possible. This includes negotiating with the sailor via an email correspondence in natural language. Finally, she must write the orders and start them on the way to the sailor. A partial prototype of IDA with most of the functionality described is now up and running. It should be complete before the beginning of the year.

Architecture and Mechanisms

Table 1 succinctly specifies several of the underlying assumptions or hypotheses that guided the design of IDA Some of these assumptions and hypotheses are self-explanatory and are not directly addressed in this paper. Those under focus here will be discussed in some detail.

IDA is intended to model a broad range of human cognitive function. Her architecture is comprised of a number of different modules each devoted to particular cognitive processes. Table 2 lists most of these modules. It also gives pointers to the sources of their underlying computational mechanisms, and to the psychological theories they support.

The processors postulated by GW theory are implemented by codelets, small pieces of code, each an independent thread. These are specialized for some simple task and often play the role of demons waiting for appropriate conditions under which to act. There are several kinds of codelets, as will

Table 1. Hypotheses from Design Decisions

Module	Hypotheses from Design Decisions
Perception	Much of human language understanding employs a combined bottom up/top down passing of activation through a hierarchical conceptual net, with the most abstract concepts in the middle.
Working Memory	The contents of perception are written to working memory before becoming conscious.
Long-term Memory	Part, but not all, of working memory, the focus, is set aside as an interface with long-term associative memory (LTM). Reads from LTM are made with cues taken from the focus and the resulting associations are written there. Writes to LTM are also made from the focus.
Consciousness	Human consciousness must have a mechanism for gathering processors (neuronal groups) into coalitions, another for conducting the competition, and yet another for broadcasting
Motivation	The hierarchy of goal contexts is fueled at the top by drives, that is by primitive motivators, and at the bottom by input from the environment, both external and internal
Goal Contexts	In humans, processors (neuronal groups) bring perceptions and thoughts to consciousness. Other processors, aware of the contents of consciousness, instantiate an appropriate goal context hierarchy, which motivates yet other processors to perform internal or external actions.
Emotions	Action selection will be influenced by emotions via their effect on drives. Emotions also influence attention and the strength with which items are stored in associative memory.
Voluntary Action	Voluntary action in humans is controlled by a timekeeper who becomes less patient as the time for a decision increases. Each time a proposal or objection reaches consciousness, its chance of becoming conscious again diminishes.
Language Production	Much of human language production results from filling in blanks in scripts, and concatenating the results.

be explained below. From a biological point of view, these codelets may well correspond to Edelman's neuronal groups (1987).

Perception in IDA consists mostly of processing incoming email messages in natural language. In sufficiently narrow domains, natural language understanding may be achieved via an analysis of surface features without the use of a traditional symbolic parser (Jurafsky & Martin 2000). Allen describes this approach to natural language understanding as complex, template-based matching (1995). Ida's relatively limited domain requires her to deal with only a few dozen or so distinct message types, each with relatively predictable content. This allows for surface level natural language processing. Her language-processing module has been implemented as a Copycat-like architecture (Hofstadter & Mitchell 1994) with codelets that are triggered by surface features. The mechanism includes a slipnet that stores domain knowledge, a pool of codelets (processors) specialized for recognizing particular pieces of text, and production templates for building and verifying understanding. Together they constitute an integrated sensing system for IDA, allowing her to recognize, categorize and understand. IDA must also perceive the contents read from databases, a much easier task. An underlying assumption motivates our design decisions about perception appears in Table 1. We will not defend this assumption in this paper.

Suppose, for example, that IDA receives a message from a sailor saying that his projected rotation date (PRD) is approaching and asking that a job be found for him. The perception module would recognize the sailor's name and social security number, and that the message is of the please-find-job type. This information would then be written to the workspace (working memory). The general principle here is that the contents of perception are written to working memory before becoming conscious.

IDA employs sparse distributed memory (SDM) as her major associative memory (Kanerva 1988). SDM is a content addressable memory that, in many ways, is an ideal computational mechanism for use as a long-term associative memory (LTM). Any item written to the workspace cues a retrieval from LTM, returning prior activity associated with the current entry. In our example, LTM will be accessed as soon as the message information reaches the workspace, and the retrieved associations will be also written to the workspace.

At a given moment IDA's workspace may contain, ready for use, a current entry from perception or elsewhere, prior entries in various states of decay, and associations instigated by the current entry, i.e. activated elements of LTM.. IDA's workspace thus consists of both short-term working memory (STM) and something very similar to the long-term working memory (LT-WM) of Ericsson and Kintsch (1995).

Since most of IDA's cognition deals with performing routine tasks with novel content, most of her workspace is structured into registers for particular kinds of data and particular usage of that data. Part, but not all, the workspace, called the *focus*[1], by Kanerva (1988)) is set aside as an interface with long-term LTM. Retrievals from LTM are made with cues taken from the focus and the resulting associations are written to other registers in the focus. The contents of still other registers in the focus are stored in (written to) associative memory. All this leads to the perception hypothesis in Table 1.

Not all of the contents of the workspace eventually make their way into consciousness. The apparatus for "consciousness" consists of a coalition manager, a spotlight controller, a broadcast manager, and a collection of attention codelets who recognize novel or problematic situations (Bogner et al. 2000).

Each attention codelet keeps a watchful eye out for some particular situation to occur that might call for "conscious" intervention. In most cases the attention codelet is watching the workspace, which will likely contain both perceptual information and data created internally, the products of "thoughts." Upon encountering such a situation, the appropriate attention codelet will be associated with the small number of codelets that carry the information describing the situation. (In the example of our message, these codelets would carry the sailor's name, his or her social security number, and the message type.) This association should lead to the collection of this small number of codelets, together with the attention codelet that collected them, becoming a coalition. Codelets also have activations. The attention codelet increases its activation in order that the coalition, if one is formed, might compete for the spotlight of "consciousness". Upon winning the competition, the contents of the coalition is then broadcast to all codelets. This leads us to the consciousness hypothesis in Table 1.

Baars addresses the question of how content arrives in consciousness (1988, pp. 98-99), offering two possible high-level mechanisms both consistent with neurophysiological timing findings. He also devotes an entire chapter (1988 Chapter 3) to neurophysiological evidence consistent with the basic concept of a global workspace. Yet no mechanisms are proposed for the three distinct processes identified as being needed in our hypothesis above. Here we have a good example of engineering, as well as psychological, considerations giving direction to neurophysiological research.

Continuing our example, an attention codelet will note the please-find-job message type, gather information codelets carrying name, ssn and message type, be formed into a coalition, and will compete for consciousness. If or when successful, its contents will be broadcast.

IDA depends on a behavior net (Maes 1989) for high-level action selection in the service of built-in drives. She has several distinct drives operating in parallel. These drives vary in urgency as time passes and the environment

[1] Not to be confused with focus as in focus of attention, an entirely different concept.

Table 2. IDA's Modules and Mechanisms and the Theories they Accommodate

Module	Computational Mechanism motivated by	Theories Accommodated
Perception	Copycat architecture (Hofstadter & Mitchell 1994)	Perceptual Symbol System (Barsalou 1999)
Working Memory	Sparse Distributed Memory (Kanerva 1988)	Long-term Working Memory (Erricsson & Kintsch 1995)
Emotions	Neural Networks (McCellland & Rumelhart 1986)	(Damasio 1999, Rolls 1999)
Associative Memory	Sparse Distributed Memory (Kanerva 1988)	
Consciousness	Pandemonium Theory (Jackson 1987)	Global Workspace Theory (Baars 1988)
Action Selection	Behavior Nets (Maes 1989)	Global Workspace Theory (Baars 1988)
Constraint Satisfaction	Linear Functional (standard operations research)	
Deliberation	Pandemonium Theory (Jackson 1987)	Human-Like Agent Architecture (Sloman 1999)
Voluntary Action	Pandemonium Theory (Jackson 1987)	Ideomotor Theory (James 1890)
Language Generation	Pandemonium Theory (Jackson 1987)	
Metacognition	Fuzzy Classifers	Human-Like Agent Architecture (Sloman 1999)

changes. Behaviors are typically mid-level actions, many depending on several behavior codelets for their execution. A behavior net is composed of behaviors, corresponding to goal contexts in GW theory, and their various links. A behavior looks very much like a production rule, having preconditions as well as additions and deletions. It's typically at a higher level of abstraction often requiring the efforts of several codelets to effect its action. Each behavior occupies a node in a digraph. The three types of links, successor, predecessor and conflictor, of the digraph are completely determined by the behaviors.

As in connectionist models (McClelland et al. 1986), this digraph spreads activation. The activation comes from that stored in the behaviors themselves, from the environment, from drives, and from internal states. The more relevant a behavior is to the current situation, the more activation it is going to receive from the environment. Each drive awards activation to those behavior that will satisfy it. Certain internal states of the agent can also send activation to the behavior net. One example might be activation from a coalition of codelets responding to a "conscious" broadcast. Activation spreads from behavior to behavior along both excitatory and inhibitory links and a behavior is chosen to execute based on activation. Her behavior net produces flexible, tunable action selection for IDA. As is widely recognized in humans the hierarchy of goal contexts is fueled at the top by drives, that is, by primitive motivators, and at the bottom by input from the environment, both external and internal.

Returning to our example, the broadcast is received by appropriate behavior-priming codelets who know to instantiate a behavior stream for reading the sailor's personnel record in the behavior net. They also bind appropriate variables with name and ssn, and send activation to a behavior that knows how to initiate the accessing of the database. If or when that behavior is executed, behavior codelets associated with it begin to read data from the sailor's file. This data is written to the workspace. Each such write results in another round of associations, the triggering of an attention codelets, the resulting information coming to "consciousness," additional binding of variables and passing of activation, and the execution of the next behavior. As long as it's the most important activity going, this process is continued until all the relevant personnel data is written to the workspace. In a similar fashion, repeated runs through "consciousness" and the behavior net result in a course selection of possible suitable jobs being made from the job requisition database.

The process just described leads us to speculate that in humans, like in IDA, processors (neuronal groups) bring perceptions and thoughts to consciousness. Other processors, aware of the contents of consciousness, instantiate an appropriate goal context hierarchy, which in turn, motivates yet other processors to perform internal or external actions.

IDA's design includes mechanisms for emotions (McCauley & Franklin 1998). She may "experience" such emotions as guilt at not getting a sailor's orders out on time, frustration at not understanding a message, and anxiety at not being able to convince a sailor to accept a suitable job. Action selection will be influenced by emotions via their effect on drives, modeling work on human action selection (Damasio 1994). Emotions also influence attention, the means of bringing certain content to consciousness. They also influence the strength of writes to LTM.

IDA is provided with a constraint satisfaction module designed around a linear functional. It provides a numerical measure of the suitability, or fitness, of a specific job for a given sailor. For each issue (say moving costs) or policy (say sea duty following shore duty) there's a function that measures suitability in that respect. Coefficients indicate the relative importance of each issue or policy. The weighted sum measures the job's fitness for this sailor at this time. The same process, beginning with an attention codelet and ending with behavior codelets, brings each function value to "consciousness" and writes the next into the workspace. At last, the job's fitness value is written to the workspace.

Since IDA's domain is fairly complex, she requires *deliberation* in the sense of creating possible scenarios, partial plans of actions, and choosing between them (Sloman 1999). In our example, IDA now has a list of a number of possible jobs in her workspace, together with their fitness values. She must construct a temporal scenario for at least a few of these possible billets to see if the timing will work out (say if the sailor can be aboard ship before the departure date). In each scenario the sailor leaves his or her current post during a certain time interval, spends a specified length of time on leave, possibly reports to a training facility on a certain date, uses travel time, and arrives at the new billet with in a given time frame. Such scenarios are valued on how well they fit the temporal constraints (the gap) and on moving and training costs. These scenarios are composed of scenes organized around events, and are constructed in the workspace by the same attention codelet to "consciousness" to behavior net to behavior codelets as described previously.

We humans most often select actions subconsciously, that is, without conscious thought. But we also make voluntary choices of action, often as a result of the kind of deliberation described above. Baars argues that such voluntary choice is the same a conscious choice (1997, p. 131). We must carefully distinguish between being conscious of the results of an action and consciously deciding to take that action, that is, of consciously deliberating on the decision. It's the latter case that constitutes voluntary action. William James proposed the *ideomotor theory* of voluntary action (James 1890). James suggests that any idea (internal proposal) for an action that comes to mind (to consciousness) is acted upon unless it provokes some opposing idea or some counter proposal. GW theory adopts James' ideomotor theory as is (1988, Chapter 7), and provides a functional architecture for it. The IDA model furnishes an underlying mechanism that implements that theory of volition and its architecture in a software agent.

Suppose that in our example at least one scenario has been successfully constructed in the workspace. The players in this decision making process include several proposing attention codelets and a timekeeper codelet. A proposing attention codelet's task is to propose that a certain job be offered to the sailor. Choosing a job to propose on the basis of the codelet's particular pattern of preferences, it brings information about itself and the proposed job to "consciousness" so that the timekeeper codelet can know of it. Its preference pattern may include several different issues (say priority, moving cost, gap, etc) with differing weights assigned to each. For example, our proposing attention codelet may place great weight on low moving cost, some weight on fitness value, and little weight on the others. This codelet may propose the second job on the scenario list because of its low cost and high fitness, in spite of low priority and a sizable gap. If no other proposing attention codelet objects (by bringing itself to "consciousness" with an objecting message) and no other such codelet proposes a different job within a given span of time, the timekeeper codelet will mark the proposed job as being one to be offered. If an objection or a new proposal is made in a timely fashion, it will not do so.

Two proposing attention codelets may alternatively propose the same two jobs several times. Several mechanisms tend to prevent continuing oscillation. Each time a codelet proposes the same job it does so with less activation and, so, has less chance of coming to "consciousness." Also, the timekeeper loses patience as the process continues, thereby diminishing the time span required for a decision. Finally, the metacognitive module watches the whole process and intervenes if things get too bad (Zhang et al. 1998a). A job proposal may also alternate with an objection, rather than with another proposal, with the same kinds of consequences. These occurrences may also be interspersed with the creation of new scenarios. If a job is proposed but objected to, and no other is proposed, the scenario building may be expected to continue yielding the possibility of finding a job that can be agreed upon.

Experimental work of neuroscientist Benjamin Libet lends support to this implementation of voluntary action as mirroring what happens in humans (Libet et al. 1983). He writes, "Freely voluntary acts are preceded by a specific electrical change in the brain (the 'readiness potential', RP) that begins 550 ms before the act. Human subjects became aware of intention to act 350-400 ms after RP starts, but 200 ms. before the motor act. The volitional process is therefore initiated unconsciously. But the conscious function could still control the outcome; it can veto the act." Libet interprets the onset of the readiness potential as the time of the decision to act. Suppose we interpret it, instead, as the time a neuronal group (attention codelet) decides to propose the action (job). The next 350-400 ms would be the time required for the neuronal group (attention codelet) to gather its information (information codelets) and win the competition for consciousness. The next 200 ms would be the time during which another neuronal group (timekeeper) would wait for objections or alternative proposals from some third neuronal group (attention codelet) before initiating the action. This scenario gets the sequence right, but begs the question of the timing. Why should it take 350 ms for the first neuronal group (attention codelet) to reach consciousness and only 200 ms for the third? Our model would require such extra time during the first pass to set up the appropriate goal context hierarchy (behavior stream) for the voluntary decision making process, but would not require it again during the second. The problem with this explanation is that we identify the moment of "consciousness" with the broadcast, which occurs before instantiation of the behavior stream. So the relevant question is whether consciousness occurs in humans only after a responding goal structure is in place? This leads us to the voluntary action hypothesis in Table 1.

Once one or more jobs are voluntarily selected, IDA must generate an email message offering them to the sailor. Due to her relatively narrow domain, IDA generates language (email messages) by filling in appropriate scripts. These reside in behavior codelets, and are chosen by the same "conscious" process as has been used so often previously. Thus much of language production results from filling in blanks in learned scripts, and concatenating the results.

Future Work

Though the IDA model cuts a broad swath, human cognition is far too rich to be easily encompassed. Still, we plan to extend the model in several ways. An alteration to the behavior net will allow habituation of actions. The capacity for learning from conversations with detailers are planned (Ramamurthy et al. 1998). A development/training period utilizing that ability is also anticipated for IDA (Franklin 2000). We're working on giving her the ability to report "conscious" activity in natural language. Though IDA deals intelligently with novel instances of routine situations, she should be able to also handle unexpected, and problematic non-routine situations. We're working on it. In modeling human cognition, there's always much left to do.

Acknowledgments

The first author was supported in part by ONR grant N00014-98-1-0332. The authors wish to acknowledge essential contributions from the Conscious Software Research Group: Ashraf Anwar, Ramesh Aitipamula , Arpad Kelemen, Ravikumar Kondadadi, Irina Makkaveeva , Lee McCauley, Aregahegn Negatu, Uma Ramamurthy, Alexei Stoliartchouk, and Zhaohua Zhang.

References

Allen, J. J. 1995. *Natural Language Understanding*. Redwood City CA: Benjamin/Cummings.

Anderson, J. R. 1990. *The Adaptive Character of Thought*. Hillsdale, NJ: Erlbaum.

Baars, B. J. 1988. *A Cognitive Theory of Consciousness*. Cambridge: Cambridge University Press.

Barsalou, L. W. 1999. Perceptual symbol systems. *Behavioral and Brain Sciences* 22:577–609.

Bogner, M., U. Ramamurthy, and S. Franklin. 2000. "Consciousness" and Conceptual Learning in a Socially Situated Agent. In *Human Cognition and Social Agent Technology*, ed. K. Dautenhahn. Amsterdam: John Benjamins.

Damasio, A. R. 1994. *Descartes' Error*. New York: Gosset; Putnam Press.

Edelman, G. M. 1987. *Neural Darwinism*. New York: Basic Books.

Ericsson, K. A., and W. Kintsch. 1995. Long-term working memory. *Psychological Review* 102:21–245.

Franklin, S. 1997. Autonomous Agents as Embodied AI. *Cybernetics and Systems* 28:499–520.

Franklin, S. 2000. Learning in "Conscious" Software Agents. In *Workshop on Development and Learning*. Michigan State U.; East Lansing, USA: April 5-7.

Franklin, S., and A. C. Graesser. 1997. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *Intelligent Agents III*. Berlin: Springer Verlag.

Franklin, S., and A. Graesser. 1999. A Software Agent Model of Consciousness. *Consciousness and Cognition* 8:285–305.

Franklin, S., A. Kelemen, and L. McCauley. 1998. IDA: A Cognitive Agent Architecture. In *IEEE Conf on Systems, Man and Cybernetics*. : IEEE Press.

Hirst, T., and T. Kalus; 1998. Soar Agents for OOTW Mission Simulation. 4th Int'l Command and Control Research and Technology Symposium. September.

Hofstadter, D. R., and M. Mitchell. 1994. The Copycat Project. In *Advances in connectionist and neural computation theory, Vol. 2: logical connections*, ed. K. J. Holyoak, and J. A. Barnden. Norwood N.J.: Ablex.

James, W. 1890. *The Principles of Psychology*. Cambridge, MA: Harvard University Press.

Jurafsky, D., and J. H. Martin. 2000. *Speech and Language Processing*. Englewood Cliffs, NJ: Prentice-Hall.

Kanerva, P. 1988. *Sparse Distributed Memory*. Cambridge MA: The MIT Press.

Kintsch, W. 1998. *Comprehension*. Cambridge: Cambridge University Press.

Laird, E. J., A. Newell, and Rosenbloom P. S. 1987. SOAR: An Architecture for General Intelligence. *Artificial Intelligence* 33:1–64.

Libet, B., C. A. Gleason, E. W. Wright, and D. K. Pearl. 1983. Time of Conscious Intention to Act in Relation to Onset of Cerebral Activity (Readiness-Potential). *Brain* 106:623–642.

Maes, P. 1989. How to do the right thing. *Connection Science* 1:291–323.

Maturana, H. R. 1975. The Organization of the Living. *International Journal of Man-Machine Studies* 7:313–332.

McCauley, T. L., and S. Franklin. 1998. An Architecture for Emotion. In *AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition*. Menlo Park, CA: AAAI Press.

McClelland, J. L., D. E. Rumelhart, et al. 1986. *Parallel Distributed Processing*, vol. 1. Cambridge: MIT Press.

Ramamurthy, U., S. Franklin, and A. Negatu. 1998. Learning Concepts in Software Agents. In *From animals to animats 5*, ed. R. Pfeifer, B. Blumberg, J.-A. Meyer , and S. W. Wilson. Cambridge,Mass: MIT Press.

Sloman, A. 1987. Motives Mechanisms Emotions. *Cognition and Emotion* 1:217–234.

Sloman, A. 1999. What Sort of Architecture is Required for a Human-like Agent? In *Foundations of Rational Agency*, ed. M. Wooldridge, and A. Rao. Dordrecht, Netherlands: Kluwer Academic Publishers.

Zhang, Z., D. Dasgupta, and S. Franklin. 1998a. Metacognition in Software Agents using Classifier Systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. Madison, Wisconsin: MIT Press.

Zhang, Z., S. Franklin, B. Olde, Y. Wan, and A. Graesser. 1998b. Natural Language Sensing for Autonomous Agents. In *Proceedings of IEEE International Joint Symposia on Intellgence Systems 98*. : .