Franklin, S. 2000. Modeling Consciousness and Cognition in Software Agents. In *Proceedings of the International Conference on Cognitive Modeling, Groeningen, NL, March 2000*, ed. N. Taatgen. : .

Modeling Consciousness and Cognition in Software Agents

Stan Franklin **(stan.franklin@memphis.edu**)
Institute for Intelligent Systems
University of Memphis
Memphis, TN, 38152, USA

Abstract

Here we describe the architectures of two "conscious" software agents and the relatively comprehensive conceptual and computational models derived from them. Modules for perception, working memory, associative memory, "consciousness," emotions, action selection, deliberation, and metacognition are included. The mechanisms implementing the agents are mostly drawn from the "new AI," but include both symbolic and connectionist-like features. These models promise to yield testable hypotheses for cognitive science and cognitive neuroscience. We also explore briefly the relationship of these architectures to several recent theories of cognition including embodied cognition and the dynamical systems approach.

Autonomous and Cognitive Agents

Artificial intelligence pursues the twin goals of understanding human intelligence and of producing intelligent software and/or artifacts. Designing, implementing and experimenting with autonomous agents furthers both these goals in a synergistic way. In particular, designing and implementing within the constraints of a theory of cognition can further the first goal by providing conceptual and computational models of that theory. 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 evolved in drives and their associated goals; in artificial agents from drives and goals built in by its creator. Such drives which act as motive generators (Sloman 1987), must be present, whether explicitly represented, or expressed causally. 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, Maturana et al. 1980).

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'll be concerned with autonomous software agents, designed for specific tasks, and 'living' in real world computing systems such as operating systems, databases, or networks.

Such autonomous software agents, when equipped with cognitive (interpreted broadly) features chosen from among multiple senses, perception, short and long term memory, attention, planning, reasoning, problem solving, learning, emotions, moods, attitudes, multiple drives, etc., are called *cognitive agents* (Franklin, 1997a). 'Though ill defined, cognitive agents can play a synergistic role in the study of human cognition, including consciousness.

Global Workspace Theory

The material in this section is from Baars' two books (1988, 1997) and superficially describes his global workspace theory of consciousness.

In his global workspace theory, Baars, along with many others (eg. (Edelman, 1987; Minsky, 1985; Ornstein, 1986), postulates that human cognition is implemented by a multitude of relatively small, special purpose processes, almost always unconscious. Communication between them is rare and over a narrow bandwidth. Coalitions of such processes find their way into a global workspace (and into consciousness). This limited capacity global workspace serves to broadcast the message of the coalition to all the unconscious processors, in order to recruit other processors to join in handling the current novel situation, or in solving the current problem. Thus consciousness in this theory allows us to deal with novelty or problematic situations that can't be dealt with efficiently, or at all, by habituated unconscious processes. In particular, it provides access to appropriately useful resources, thereby solving the relevance problem. (Even more often, consciousness serves a lookout function, being aware of potential dangers or opportunities.)

All this takes place under the auspices of contexts: goal contexts, perceptual contexts, conceptual contexts, and/or cultural contexts. Baars uses goal hierarchies, dominant goal contexts, a dominant goal hierarchy, dominant context hierarchies, and lower level context hierarchies. Each context is, itself a coalition of processes. Though contexts are typically unconscious, they strongly influence conscious processes.

Baars postulates that learning results simply from conscious attention, that is, that consciousness is sufficient for learning. There's much more to the theory, including attention, action selection, emotion, voluntary action, metacognition and a sense of self. I think of it as a high level theory of cognition.

"Conscious" Software Agents

A "conscious" software agent is defined to be a cognitive software agent that implements global workspace theory. (No claim of sentience is being made.) I believe that "conscious" software agents have the potential to play a synergistic role in both modeling cognitive theory and in producing software with more human-like intelligence. Minds can be viewed as control structures for autonomous agents (Franklin, 1995). A theory of mind constrains the design of a cognitive agent that implements (models) that theory. While a theory is typically abstract and only broadly sketches an architecture, an implemented computational design provides a fully articulated architecture and a complete set of mechanisms. This architecture and set of mechanisms provides a richer, more concrete and more decisive theory, as well as both a conceptual and a computational model. Moreover, every design decision taken during an implementation translates into a hypothesis about how human minds work. These hypotheses may motivate experiments with humans and other forms of empirical tests. Conversely, the results of such experiments motivate corresponding modifications of the architecture and mechanisms of the cognitive agent. In this way, the concepts and methodologies of cognitive science and of computer science will work synergistically to enhance our understanding of mechanisms of mind (Franklin, 1997a).

"Conscious" Mattie

"Conscious" Mattie (CMattie) is a "conscious" clerical software agent ((Bogner, Ramamurthy, & Franklin, in press; Franklin, 1997b; McCauley & Franklin, 1998; Zhang, Franklin, Olde, Wan, & Graesser, 1998). She composes and emails out weekly seminar announcements, having communicated by email with seminar organizers and announcement recipients in natural language. She maintains her mailing list, reminds organizers who are late with their

information, and warns of space and time conflicts. There is no human involvement other than via these email messages. CMattie's cognitive modules include perception, learning, action selection, associative memory, "consciousness," emotion and metacognition. Her emotions influence her action selection. Her mechanisms include variants and/or extensions of Maes' behavior nets (1989), (1994), Jackson's pandemonium theory (1987), Kanerva's sparse distributed memory (1988), and Holland's classifier systems (Holland, 1986).

IDA

IDA (Intelligent Distribution Agent) is a “conscious” software agent being developed for the US Navy (Franklin, Kelemen, & McCauley, 1998). At the end of each sailor's tour of duty, he or she is assigned to a new billet. This assignment process is called distribution. The US Navy employs some 200 people, called detailers, full time to effect these new assignments. IDA's task is to facilitate this process, by playing the role of detailer. Designing IDA presents both communication problems, and action selection problems involving constraint satisfaction. She must communicate with sailors via email and in natural language, understanding the content and producing human-like responses. Sometimes she will initiate conversations. She must access a number of databases, again understanding the content. She must see that the Navy's needs are satisfied, for example, the required number of sonar technicians on a destroyer with the required types of training. In doing so she must adhere to some ninety policies. She must hold down moving costs. And, she must 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. IDA's architecture and mechanisms are largely modeled after those of CMattie, though more complex. In particular, IDA will require improvised language generation where for CMattie scripted language generation sufficed. Also IDA will need deliberative reasoning (Sloman, 1999) in the service of action selection, where CMattie was able to do without. Her emotions will be involved in both of these.

“Conscious” Software Architecture and Mechanisms

In both the CMattie and IDA architectures the processors postulated by global workspace theory are implemented by codelets, small pieces of code. These are specialized for some simple task and often play the role of demons waiting for appropriate conditions under which to act.

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, 1999; Bogner, Ramamurthy, & Franklin, in press). Each attention codelet keeps a watchful eye out for some particular situation to occur that might call for “conscious” intervention. 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. 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 might compete for “consciousness” if one is formed.

In CMattie and IDA the coalition manager is responsible for forming and tracking coalitions of codelets. Such coalitions are initiated on the basis of the mutual associations between the member codelets. At any given time, one of these coalitions finds it way to "consciousness," chosen by the spotlight controller, who picks the coalition with the highest average activation among its member codelets. Global workspace theory calls for the contents of "consciousness" to be broadcast to each of the codelets. The broadcast manager accomplishes this.

Perception

Perception in both CMattie and IDA consists mostly of understanding 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. Allen describes this approach as complex, template-based matching, natural language processing (1995). CMattie's limited domain requires her to deal with only a dozen or so distinct message types, each with relatively predictable content. This allows for surface level natural language processing. CMattie's language understanding module has been implemented as a Copycat-like architecture (Hofstadter & Mitchell, 1994) though her understanding takes place differently. The mechanism includes a slipnet storing domain knowledge, and a pool of codelets (processors) specialized for specific jobs, along with templates for building and verifying understanding. Together they constitute an integrated sensing system for the autonomous agent CMattie. With it she's able to recognize, categorize and understand. IDA, though much more complex, perceives in much the same way.

Action Selection

Both CMattie and IDA depend on a behavior net (Maes, 1989) for high-level action selection in the service of built-in drives. Each has several distinct drives operating in parallel. These drives vary in urgency as time passes and the environment changes. Behaviors are typically mid-level actions, many depending on several codelets for their execution. A behavior net is composed of behaviors and their various links. A behavior looks very much like a production rule, having preconditions as well as additions and deletions. A behavior is distinguished from a production rule by the presence of an activation, a number indicating some kind of strength level. Each behavior occupies a node in a digraph (directed graph). The three types of links of the digraph are completely determined by the behaviors. If a behavior X will add a proposition b, which is on behavior Y's precondition list, then put a successor link from X to Y. There may be several such propositions resulting in several links between the same nodes. Next, whenever you put in a successor going one way, put a predecessor link going the other. Finally, suppose you have a proposition m on behavior Y's delete list that is also a precondition for behavior X. In such a case, draw a conflictor link from X to Y, which is to be inhibitory rather than excitatory.

As in connectionist models, this digraph spreads activation. The activation comes from activation stored in the behaviors themselves, from the environment, from drives, and from internal states. The environment awards activation to a behavior for each of its true preconditions. The more relevant it is to the current situation, the more activation it's going to receive from the environment. This source of activation tends to make the system opportunistic. Each drive awards activation to every behavior that, by being active, will satisfy that drive. This source of activation tends to make the system goal directed. Certain internal

states of the agent can also send activation to the behavior net. This activation, for example, might come from a coalition of codelets responding to a "conscious" broadcast. Finally, activation spreads from behavior to behavior along links. Along successor links, one behavior strengthens those behaviors whose preconditions it can help fulfill by sending them activation. Along predecessor links, one behavior strengthens any other behavior whose add list fulfills one of its own preconditions. A behavior sends inhibition along a conflictor link to any other behavior that can delete one of its true preconditions, thereby weakening it. Every conflictor link is inhibitory. Call a behavior *executable* if all of its preconditions are satisfied. To be acted upon a behavior must be executable, must have activation over threshold, and must have the highest such activation. Behavior nets produce flexible, tunable action selection for these agents.

Action selection via behavior net suffices for CMattie due to her relatively constrained domain. IDA's domain is much more complex, and requires deliberation in the sense of creating possible scenarios, partial plans of actions, and choosing between them. For example, suppose IDA is considering a sailor and several possible jobs, all seemingly suitable. She must construct a scenario for each of these possible billets. In each scenario the sailor leaves his or her current position during a certain time interval, spends a specified length of time on leave, possibly reports to a training facility on a certain date, and arrives at the new billet with in a given time frame. Such scenarios are valued on how well they fit the temporal constraints and on moving and training costs.

Scenarios are composed of scenes. IDA's scenes are organized around events. Each scene may require objects, actors, concepts, relations, and schema represented by frames. They are constructed in a computational workspace corresponding to working memory in humans. We use Barsalou's perceptual symbol systems as a guide (1999). The perceptual/conceptual knowledge base of this agent takes the form of a semantic net with activation called the slipnet. The name is taken from the Copycat architecture that employs a similar construct (Hofstadter & Mitchell, 1994). Nodes of the slipnet constitute the agent's perceptual symbols. Pieces of the slipnet containing nodes and links, together with codelets whose task it is to copy the piece to working memory constitute Barsalou's perceptual symbol simulators. These perceptual symbols are used to construct scenes in working memory. The scenes are strung together to form scenarios. The work is done by deliberation codelets. Evaluation of scenarios is also done by codelets.

Deliberation, as in humans, is mediated by the "consciousness" mechanism. Imagine IDA in the context of a behavior stream whose goal is to find a billet for a particular sailor. Perhaps a behavior executes to read appropriate items from the sailor's personnel database record. Then, possibly, comes a behavior to locate the currently available billets. Next might be a behavior that runs each billet and that sailor through IDA's constraint satisfaction module, producing a small number of candidate billets. Finally a deliberation behavior may be executed that sends deliberation codelets to working memory together with codelets carrying billet information. A particular billet's codelets wins its way into "consciousness." Scenario building codelets respond to the broadcast and begin creating scenes. This scenario building process, again as in humans, has both it's "unconscious" and its "conscious" activities. Eventually scenarios are created and evaluated for each candidate billet and one of them is chosen. Thus we have behavior control via deliberation.

The mediation by the "consciousness" mechanism, as described in the previous paragraph is characteristic of IDA. The principle is that she should use "consciousness"

whenever a human detailer would be conscious in the same situation. For example, IDA could readily recover all the needed items from a sailor's personnel record unconsciously with a single behavior stream. But, observing and questioning human detailers indicate that they become conscious of each item individually. Hence, according to our principle, so must IDA be "conscious" of each retrieved personnel data item.

Associative Memory

Both CMattie and IDA employ sparse distributed memory (SDM) as their major associative memories (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. Being content addressable means that items in memory can be retrieved by using part of their contents as a cue, rather than having to know the item's address in memory.

The inner workings of SDM rely on large binary spaces, that is, spaces of vectors containing only zeros and ones, called bits. These binary vectors, called words, serve as both the addresses and the contents of the memory. The dimension of the space determines the richness of each word. These spaces are typically far too large to implement in an conceivable computer. Approximating the space uniformly with a possible number of actually implemented, hard locations surmounts this difficulty. The number of such hard locations determines the carrying capacity of the memory. Features are represented as one or more bits. Groups of features are concatenated to form a word. When writing a word to memory, a copy of the word is placed in all close enough hard locations. When reading a word, a close enough cue would reach all close enough hard locations and get some sort of aggregate or average out of them. As mentioned above, reading is not always successful. Depending on the cue and the previously written information, among other factors, convergence or divergence during a reading operation may occur. If convergence occurs, the pooled word will be the closest match (with abstraction) of the input reading cue. On the other hand, when divergence occurs, there is no relation -in general- between the input cue and what is retrieved from memory.

SDM is much like human long-term memory. A human often knows what he or she does or doesn't know. If asked for a telephone number I've once known, I may search for it. When asked for one I've never known, an immediate "I don't know" response ensues. SDM makes such decisions based on the speed of initial convergence. The reading of memory in SDM is an iterative process. The cue is used as an address. The content at that address is read as a second address, and so on until convergence, that is, until subsequent contents look alike. If it doesn't quickly converge, an "I don't know" is the response. The "on the tip of my tongue phenomenon" corresponds to the cue having content just at the threshold of convergence. Yet another similarity is the power of rehearsal during which an item would be written many times and, at each of these to a thousand locations That's the "distributed" pare of sparse distributed memory. A well-rehearsed item can be retrieved with smaller cues. Another similarity is forgetting, which would tend to increase over time as a result of other similar writes to memory.

How do the agents use this associative memory? As one example, let's suppose an email message for CMattie arrives, is transferred into the perceptual workspace (working memory), and is descended upon by perceptual codelets looking for words or phrases they recognize. When such are found, nodes in the slipnet (a semantic net type mechanism with activation passing that acts as a perceptual and conceptual knowledge structure) are activated, a message type is selected, and the appropriate template filled. The information thus created from the

incoming message is then written to the perception registers in the focus, making it available to the rest of the system.

The contents of the focus are then used as an address to query associative memory. The results of this query, that is, whatever CMattie associates with this incoming information, are written into their own registers in the focus. This may include some emotion and some previous action. Attention codelets then attempt to take this information to "consciousness." They bring along any discrepancies they may find, such as missing information, conflicting seminar times, etc. Information about the current emotion and the currently executing behavior are written to the focus by appropriate codelets. The current percept, consisting of the incoming information as modified by associations and the current emotion and behavior, are then written to associative memory. Those percepts carrying strong emotions are written repeatedly yielding stronger associations. IDA handles perception in much the same way.

Emotions

In both CMattie and IDA we include mechanisms for emotions (McCauley & Franklin, 1998). CMattie, for example may "experience" such emotions as guilt at not getting an announcement out on time, frustration at not understanding a message, and anxiety at not knowing the speaker and title of an impending seminar. Action selection will be influenced by emotions via their effect on drives, modeling recent work on human action selection (Damasio, 1994).

CMattie can "experience" four basic emotions, anger, fear, happiness and sadness. These emotions can vary in intensity as indicated by their activation levels. For example, anger can vary from mild annoyance to rage as its activation rises. A four vector containing the current activations of these four basic emotions represents CMattie's current emotional state. Like humans, there's always some emotional state however slight. Also like humans, her current emotional state is often some complex combination of basic emotions or results from some particular changes in them. The effect of emotions on codelets, drives, etc. varies with their intensity. Fear brought on by an imminent shutdown message might be expected to strengthen CMattie's self-preservation drive resulting in additional activation going from it into the behavior net.

CMattie's emotional codelets serve to change her emotional state. When its preconditions are satisfied, an emotional codelet will enhance or diminish one of the four basic emotions. An emotion can build till saturation occurs. Repeated emotional stimuli result in habituation. Emotion codelets can also combine to implement more complex secondary emotions that act by affecting more than one basic emotion at once. Emotion codelets also serve to enhance or diminish the activation of other codelets. They also act to increase or decrease the strength of drives, thereby influencing CMattie's choice of behaviors.

Metacognition

Metacognition should include knowledge of one's own cognitive processes, and the ability to actively monitor and consciously regulate them. This would require self-monitoring, self-evaluation, and self-regulation. Following Minsky, we'll think of CMattie's "brain" as consisting of two parts, the A-brain and the B-brain (1985) The A-brain consists of all the other modules of the agent's architecture. It performs all of her cognitive activities except metacognition. Its environment is the outside world, a dynamic, but limited, real world environment. The B-brain, sitting on top of the A-brain, monitors and regulates it. The B-brain's environment is the A-brain, or more specifically, the A-brain's activities.

One can look at a metacognitive module, in either CMattie or IDA, as an autonomous agent in its own right. It senses the A-brain's activity and acts upon it over time in pursuit of its own agenda. It's also structurally coupled to its quite restricted environment. Its agenda derives from built in metacognitive drives. One such drive is to interrupt oscillatory behavior. Another such might be to keep the agent more on task, that is, to make it more likely that a behavior stream would carry out to completion. Yet another would push toward efficient allocation of resources.

"Conscious" Agents as Cognitive Models

In this section we examine briefly the "conscious" software agent model of consciousness and cognition from the perspective of several different theories.

Global workspace theory.

Since we've specifically set out to implement this theory, these agents should be constrained by its dictums and should perform the functions it specifies. To what extent does CMattie do so? In an appendix to his most recent book (1997) Baars provides a "… technical summary of the major bodies of evidence about conscious experience," and challenges the reader to produce a theory that accounts for them, as he has with global workspace theory. The evidence in question comes from cognitive science and from cognitive neuroscience.

Almost all the psychological facts presented in the appendix are accounted for by the models specified by the agent architecture we've described (Franklin & Graesser, 1999). Of those that are not, most fail as a consequence of the choice of domain, for example, because CMattie has no visual sense. The facts not accounted for do not, in principle, present difficulty for the architecture. The weakest link seems to be a not completely adequate performance in habituation and in acquiring automaticity. We conclude that CMattie accounts for the evidence quite well, but not perfectly. The same is true of IDA.

Though this conclusion is both accurate and encouraging, there are still missing functions. CMattie has no self, though the theory calls for one. She is capable of voluntary action, but not in the full sense of James' ideomotor theory as called for by global workspace theory. IDA will implement voluntary action fully. But, she won't be able to report on her internal activity. These and other such missing functions are planned for inclusion in future, more complex, "conscious" software agents.

Sloman's Model

Aaron Sloman has developed a useful high-level model of the architecture of human action selection (Sloman, 1999). This architecture includes three layers, the reactive, the deliberative, and the meta-management layers, all highly interconnected. He describes the reactive layer as one in which "information is acquired through external sensors and internal monitors and propagates through and around the system, and out to effectors of various kinds." The deliberative layer, on the other hand, can "explicitly construct representations of alternative possible actions, evaluate them and choose between them, all in advance of performing them." Sloman goes on to describe agents with a meta-management layer as "capable of monitoring, evaluating, and modifying high level aspects of their own internal processes. These three layers can operate simultaneously and affect each in both a top-down and a bottom-up fashion.

CMattie has a reactive layer, what we've called the A-brain above. Information is created from sensations (email messages), "propagates through and around" the focus, the memories,

and the behavior net, and results in outgoing messages through the output module (an effector). CMattie doesn't have a deliberative layer. In her so simple domain there is little for her to deliberate about. Thus, her architecture illustrates that a purely reactive agent need not be simple. IDA, on the other hand, must deliberate. Her creation and choice of scenarios constitutes "conscious" deliberation, as does here voluntary "conscious" choice of which jobs to offer a sailor.

Both CMattie and IDA have meta-management layers in their metacognition modules, though in CMattie's case, meta-management is "impoverished" by the lack of its own ability to deliberate. IDA's metacognition module is not yet designed. CMattie provides an example of an autonomous agent with reactive and meta-management layers, but with no deliberative layer. This observation contributes in a small way to Sloman's program of exploring design space. "…we need to explore a space of possible designs for behaving systems (design space) …"

Glenberg's Model

Arthur Glenberg, suggests that we humans store concepts in terms of actions related to the concept (1997). This work is characteristic of a new paradigm of theories of cognition typically referred to as embodied, or situated, cognition (Varela, Thompson, & Rosch, 1991). This paradigm, stresses the importance of the body and its sensory-motor system to cognition. Glenberg's view of conceptual memory emphasizes the motor aspect. I understand a cup largely by virtue of what I can do with it.

Concepts in CMattie are found in slipnet nodes and their interconnections in the perception module. The actions Glenberg speaks of are embodied in these agents in the codelets (actions) that underlie such nodes and their neighbors. Actions associated with more abstract concepts, such as message type, lie in the behaviors appropriate to this message type and their underlying codelets. Also, developing concepts in CMattie's associative memory may well be accompanied by remembered behaviors (actions). CMattie's architecture, and IDA's, fits well with this view of the storing of concepts.

Barsalou's Perceptual Symbol Systems

As mentioned above, Barsalou, pushes the sensation side of sensory-motor embodiment. He wants to replace the arbitrary, amodal symbols with modal, analogical symbols tied directly to perception. He calls these neural pattern productions of the human perceptual systems *perceptual symbols* (Barsalou, 1999). "As collections of perceptual symbols develop, they constitute the representations that underlie cognition."

CMattie's sensory input consists primarily of the text of incoming email messages. Perceptual states (symbols) arise in the slipnet as a result of nodes being activated by codelets together with spreading activation. A subset of the state is extracted and sent to the focus, from whence it's stored in associative (long term) memory. Thus CMattie contains and builds perceptual symbols.

In addition to perceptual symbols Barsalou's theory relies on perceptual symbol simulators that allow humans to mentally recreate objects and events in their absence, and that implement a "conceptual system that represents types, supports categorization, and produces categorical inferences." CMattie's architecture allow for perceptual symbol simulators only in a rather unsophisticated way, e.g. the codelet recognizing the concept "Tuesday" being able to simulate its common forms. However, IDA 's architecture will rely heavily on such simulators

in both her deliberation and language production processes. These perceptual simulators will take the form of codelets who respond to “conscious” broadcasts with appropriately chosen and specified internal images written to the workspace.

Dynamical Systems Theory

Yet another newly popular paradigm in cognitive science is that of dynamical systems (Freeman & Skarda, 1990; Kaufaman, 1993; Thelen & Smith, 1994; van Gelder, 1999). In our “conscious” software agents, activation passes in the slipnet, in the behavior net, between behaviors and their associated codelets, from emotion codelets, and in several other places. In some places, attractor landscapes are easily discerned, as for example that built around the message type nodes (CMattie), or idea type nodes (IDA) in the slipnet as point attractors. Other such attractor landscapes are currently under investigation. How “consciousness” can be described dynamically in these architectures is also under study.

Conclusions

Here we’ve described two “conscious” software agents that implement relative comprehensive models of human cognition. This model accommodates, more or less well, Baars’ global workspace theory, Sloman’s high action selection architecture, Glenberg’s concepts as actions theory, Barsalou’s perceptual symbol systems, and the dynamical hypothesis. The model also incorporates many lessor facts from current cognitive science research. We expect it to lead to testable hypotheses for cognitive scientists and cognitive neuroscientists (Bogner, Franklin, Graesser, & Baars, In preparation). We hope to have provided evidence in favor of using autonomous software agents as vehicles for cognitive modeling.

Acknowledgements

This research was supported in part by NSF grant SBR-9720314 and by ONR grant N00014-98-1-0332. It was performed with essential contributions from the Conscious Software Research Group including Art Graesser, Sri Satish Ambati, Ashraf Anwar, Myles Bogner, Arpad Kelemen, Ravikumar Kondadati, Irina Makkaveeva, Lee McCauley, Aregahegn Negatu, and Hongjun Song.

References

Allen, J. J. (1995). Natural Language Understanding. Redwood City CA: Benjamin/Cummings; Benjamin; Cummings.

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

Baars, B. J. (1997). In the Theater of Consciousness. Oxford: Oxford University Press.

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

Bogner, M. (1999). Realizing "consciousness" in software agents. Unpublished Ph.D. Dissertation, University of Memphis.

Bogner, M., Franklin, S., Graesser, A., & Baars, B. J. (In preparation). Hypotheses From "Conscious" Software. .

Bogner, M., Ramamurthy, U., & Franklin, S. (in press). Consciousness" and Conceptual Learning in a Socially Situated Agent. In K. Dautenhahn (Ed.), Human Cognition and Social Agent Technology. Amsterdam: John Benjamins.

Damasio, A. R. (1994). Descartes' Error. New York: Gosset; Putnam Press.
Edelman, G. M. (1987). Neural Darwinism. New York: Basic Books.
Franklin, S. (1995). Artificial Minds. Cambridge MA: MIT Press.
Franklin, S. (1997a). Autonomous Agents as Embodied AI. Cybernetics and Systems, 28, 499–520.
Franklin, S. (1997b). Global Workspace Agents. Journal of Consciousness Studies, 4, 322–334.
Franklin, S., & Graesser, A. (1999). A Software Agent Model of Consciousness. Consciousness and Cognition, 8, 285–305.
Franklin, S., Kelemen, A., & McCauley, L. (1998). IDA: A Cognitive Agent Architecture. In IEEE Conf on Systems, Man and Cybernetics (Series Eds. 2646–2651). : IEEE Press.
Freeman, W. J., & Skarda, C. (1990). Representations: Who Needs Them? In J. L. McGaugh (Ed.), Brain Organization and Memory Cells, Systems, and Circuits (Series Eds. 375–380). New York: Oxford University Press.
Glenberg, A. M. (1997). What memory is for. Behavioral and Brain Sciences, 20, 1–19.
Hofstadter, D. R., & Mitchell, M. (1994). The Copycat Project: A model of mental fluidity and analogy-making. In K. J. Holyoak & J. A. Barnden (Eds.), Advances in connectionist and neural computation theory, Vol. 2: logical connections. Norwood N.J.: Ablex.
Holland, J. H. (1986). A Mathematical Framework for Studying Learning in Classifier Systems. Physica, 22 D, 307–317. (Also in Evolution, Games and
Learning. Farmer, J. D., Lapedes, A., Packard, N. H., and Wendroff, B. (eds.). NorthHolland (Amsterdam))
Jackson, J. V. (1987). Idea for a Mind. Siggart, Newsletter, 181, 23–26.
Kanerva, P. (1988). Sparse Distributed Memory. Cambridge MA: The MIT Press.
Kaufaman, S. A. (1993). The origins of order. Oxford: Oxford University Press.
Maes, P. (1989). How to do the right thing. Connection Science, 1, 291–323.
McCauley, T. L., & Franklin, S. (1998). An Architecture for Emotion. In AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition (Series Eds. 122–127). Menlo Park, CA: AAAI Press.
Minsky, M. (1985). The Society of Mind. New York: Simon and Schuster.
Ornstein, R. (1986). Multimind. Boston: Houghton Mifflin.
Sloman, A. (1999). What Sort of Architecture is Required for a Human-like Agent? In M. Wooldridge & A. Rao (Eds.), Foundations of Rational Agency. : Portland Oregon.
Thelen, E., & Smith, L. B. (comps.). (1994). A dynamic systems approach to the development of cognition and action. Cambridge, MA: MIT Press.
van Gelder, T. (1999). The dynamical hypothesis in cognitive science. Behavioral and Brain Sciences.
Varela, F. J., Thompson, E., & Rosch, E. (1991). The Embodied Mind. Cambridge, MA: MIT Press.
Zhang, Z., Franklin, S., Olde, B., Wan, Y., & Graesser, A. (1998). Natural Language Sensing for Autonomous Agents. In Proceedings of IEEE International Joint Symposia on Intellgence Systems 98 (Series Eds. 374–381). : .