

University of Memphis

University of Memphis Digital Commons

CCRG Papers

Cognitive Computing Research Group

1999

Action Selection and Language Generation in “Conscious” Software Agents

S. Franklin

Follow this and additional works at: https://digitalcommons.memphis.edu/ccrg_papers

Recommended Citation

Franklin, S. (1999). Action Selection and Language Generation in “Conscious” Software Agents. Retrieved from https://digitalcommons.memphis.edu/ccrg_papers/144

This Document is brought to you for free and open access by the Cognitive Computing Research Group at University of Memphis Digital Commons. It has been accepted for inclusion in CCRG Papers by an authorized administrator of University of Memphis Digital Commons. For more information, please contact khggerty@memphis.edu.

Action Selection and Language Generation in "Conscious" Software Agents

Stan Franklin^{1,2,3}

Institute for Intelligent Systems and
Department of Mathematical Sciences
The University of Memphis

Abstract

Here we'll briefly describe action selection and language generation mechanisms in two "life-like" software agents, CMattie and IDA, and discuss issues that bear on three workshop topics: architectures for behavior control, interdependencies between emotions and goal-based behaviors, and coordination of scripted and improvised behaviors. These agents are life-like in the sense of interacting with humans via email in natural language. They are "conscious" only in the sense of implementing a psychological theory of consciousness (Baars 1988, 1997). At this writing we are exploring the transition from scripted language production to more improvised speech generation. We are also investigating deliberative behavior selection mechanisms whereby alternative scenarios are produced and evaluated, and one of them chosen and acted upon.

Agents: Autonomous, Cognitive and "Conscious"

The rise of the web has spawned a use of the word "agent" in other than its more common meaning as in "travel agent" or "insurance agent" or "real-estate agent." In this new context an agent is a piece of software, a computer program, that in some sense acts on it's own, typically in the service of its user, a person. Here I've chosen to define the technical term "autonomous agent," at least partly to avoid the seemingly endless debates over exactly what is an agent. An *autonomous agent*, in this paper 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. It also acts in such a way as to possibly influence what it senses at a later time (Franklin and Graesser 1997). These autonomous agents include biological agents such as humans and most, perhaps all, animals.

¹ Supported in part by ONR grant N00014-98-1-0332

² With essential contributions from the Conscious Software Research Group including Art Graesser, Satish Ambati, Ashraf Anwar, Myles Bogner, Arpad Kelemen, Irina Makkaveeva, Lee McCauley, Aregahegn Negatu, Uma Ramamurthy, Zhaohua Zhang

³ stan.franklin@memphis.edu, www.msci.memphis.edu/~franklin

They also include some mobile robots, like the robots that deliver pharmaceuticals in hospitals. Some computational agents such as some artificial life agents who “live” in artificial environments designed for them within computer systems (Ackley & Littman, 1992) are also autonomous agents. Finally, so also are the objects of our attention in this paper, software agents, at least some of them. These autonomous software agents include task-specific agents like spiders that search for links on the web, such entertainment agents as Julia (Mauldin, 1994), and, much to the regret of many of us, computer viruses. The class also includes the “conscious” software agents we’ll describe here.

But the notion of autonomous agent turns out to be too broad for our needs; it even includes a thermostat. So, let’s restrict it to make it more suitable. Suppose we equip our autonomous agent, with cognitive features, interpreting “cognitive” broadly so as to include emotions and such. Choose these features from among multiple senses, perception, short and long term memory, attention, planning, reasoning, problem solving, learning, emotions, moods, attitudes, multiple drives, etc., and call the resulting agent a *cognitive agent* (Franklin 1997). Cognitive agents would include humans, some of our primate relatives, and perhaps elephants, some cetaceans, and perhaps even Alex, an African grey parrot (Pepperberg 1994). Examples are rare among non-biological agents, perhaps eventually including Rod Brooks’ humanoid robot Cog (Brooks 1997)], some agents modeled on Sloman’s architectural scheme (Sloman 1996), our own software agents VMattie, CMattie and IDA to be described below, and a handful of others.

Though quite ill defined, cognitive agents can play a useful, even a synergistic role in the study of human cognition, including consciousness. Here’s how it can work. A theory of cognition constrains the design of a cognitive agent that implements that theory. While a theory is typically abstract, functional and only broadly sketches an architecture, an implemented design must provide a fully articulated architecture, and the mechanisms upon which it rests. This architecture and these mechanisms serve to flesh out the theory, making it more concrete. Also, every design decision taken during an implementation constitutes a hypothesis about how human minds work. The hypothesis says that humans do it the way the agent was designed to do it, whatever “it” was. These hypotheses will suggest experiments with humans by means of which they can be tested. Conversely, the results of such experiments will suggest corresponding modifications of the architecture and mechanisms of the cognitive agent implementing the theory. The concepts and methodologies of cognitive science and of computer science will work synergistically to enhance our understanding of mechanisms of mind. I have written elsewhere in much more depth about this research strategy (Franklin 1997).

Here we’ll be concerned with “*conscious*” *software agents*. These agents are cognitive agents in that they consist of modules for perception, action selection (including constraint satisfaction and deliberation), several working memories, associative memory, episodic memory, emotion, several kinds of learning, and metacognition. They model much of human cognition. But, in addition, these agents include a module that models human consciousness according to global workspace theory (Baars 1988, 1997). Baars 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 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, according to 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. This is the key insight of global workspace theory. There's much, much more to it than is stated here.

CMattie and IDA

CMattie is a "conscious" clerical software agent (Bogner, Ramamurthy and Franklin in press; McCauley and Stan Franklin 1998; Ramamurthy, Bogner, and Franklin 1998, Zhang, Franklin and Dasgupta 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 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 extension of Maes' behavior nets (1990), Hofstadter and Mitchell's Copycat architecture (1994), Jackson's pandemonium theory, Kanerva's sparse distributed memory (1988), and Holland's classifier systems (1986).

In the CMattie architecture 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 demon waiting for appropriate condition under which to act. The apparatus for producing "consciousness" consists of a coalition manager, a spotlight controller, a broadcast manager, and a collection of "consciousness" codelets who recognize novel or problematic situations (Bogner, 1998; Bogner, Ramamurthy, and Franklin, to appear). Each "consciousness" 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 "consciousness" 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 "consciousness" codelet that collected them, becoming a coalition. Codelets also have activations. The "consciousness" codelet increases its activation in order that the coalition might compete for "consciousness" if one is formed.

CMattie's 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. Since association can both increase and diminish, the forming and tracking of coalitions is a dynamic process. Coalitions appear and disappear. Codelets may leave one coalition, and may join another. At any given time, one of these coalitions finds it way to "consciousness." This is effected by the spotlight controller who chooses 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.

IDA (Intelligent Distribution Agent) is to be a conscious software agent developed for the US Navy (Franklin, Kelemen, and 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 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 life-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. 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 in the service of action selection, where CMattie was able to do without. Her emotions will be involved in both of these.

Architectures for Behavior Control

Both CMattie and IDA depend on a behavior net (Maes 1990) 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 (in press). 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 and 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 its “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.

Emotions and Goal-based Behaviors

In both CMattie and IDA we include mechanisms for emotions (McCauley and 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. 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.

Scripted and Improvised Behaviors

CMattie’s behaviors consist almost entirely of sending email messages to seminar organizers, attendees and the system administrator. In every case these messages are composed by codelets filling out templates, that is scripts with blanks allowing them to be specialized to suit the current situation. This is even true when CMattie is in the process of learning new concepts and/or behavior via interactions with organizers (Ramamurthy, Bogner and Franklin 1998). If, for example, she writes “If a colloquium is different from a seminar, how is it different?” she has filled in a template adding “seminar” and “colloquium.” Of course, she has to be able to understand the reply.

IDA, on the other hand, must be able to respond to quite varied email messages from sailors concerning their next assignment. Many ideas from these messages will be about standard

requests and can be answered with a script. It may be that all such can be so answered. We're currently cataloging such ideas, and will soon begin producing appropriate scripts. But, what of the rare idea that isn't found in our catalog? If it's something about which IDA has no knowledge she, like a human, will not be capable of any intelligent response except, possibly, to try to learn. If IDA knows something of the subject of the idea, this knowledge will likely be found in her slipnet. An improvised response would then be created by a language generation module working from the same principles as the deliberation module described above. Improvised linguistic structures create in the workspace might be combined with scripts to produce an appropriate response. All this would be controlled by a stream of behaviors in IDA's behavior net, and would be mediated by her "consciousness" mechanism.

Conclusions

The "conscious" software agent architecture offers a promising vehicle for producing autonomous software agents that are life-like in their interactions with humans via email. They will be life-like in that they understand the human correspondent's natural language and are able to respond, also in natural language. The architecture and mechanisms underlying these abilities is, itself, life-like in that it is modeled after human cognition and consciousness. Such "conscious" software agents show promise of being able to duplicate the tasks of many different human information agents.

References

- Ackley, David, and Michael Liittman (1992). Interactions between Learning and Evolution. In *Artificial Life II*. (Langton, et al, ed.) Redwood City, CA: Addison-Wesley, 487-509.
- 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. (in press), *Perceptual Symbol Systems*, Behavioral and Brain Sciences.
- Bogner, M. (1998) Creating a "conscious" agent. Master's thesis, The University of Memphis.
- Bogner, Myles, Uma Ramamurthy, and Stan Franklin (to appear). "Consciousness" and Conceptual Learning in a Socially Situated Agent. in Kerstin Dautenhahn ed. *Human Cognition and Social Agent Technology*
- Brooks, Rodney A. (1997), The Cog Project. *J. of the Robotics Society of Japan*, 15, 968-70.
- Damasio, A. R. (1994) *Descartes' Error*, New York: Gosset/Putnam Press.
- Franklin, Stan. (1997) Autonomous Agents as Embodied AI, *Cybernetics and Systems*. Special Issue on Epistemological Aspects of Embodied AI 28:6 499-520.
- Franklin, Stan, Arpad Kelemen, and Lee McCauley (1998), IDA: A Cognitive Agent Architecture, *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, 2646-2651.
- Franklin, S. & Graesser, A. (1997) Is it an Agent, or just a Program?: A Taxonomy for

Autonomous Agents. Intelligent Agents III, Berlin: Springer Verlag, 21-35.

- Hofstadter, D. R. and Mitchell, M. (1994), "The Copycat Project: A model of mental fluidity and analogy-making." In Holyoak, K.J. & Barnden, J.A. (Eds.) Advances in connectionist and neural computation theory, Vol. 2: Analogical connections. Norwood, N.J.: Ablex.
- Holland, J. H. (1986), "A Mathematical Framework for Studying Learning in Classifier Systems." In D., Farmer et al, Evolution, Games and Learning: Models for Adaption in Machine and Nature. Amsterdam: North-Holland
- Kanerva, Pentti (1988), Sparse Distributed Memory, Cambridge MA: The MIT Press.
- Mauldin, M. L. (1994) Chatterbots, Tnymuds, And The Turing Test: Entering The Loebner Prize Competition. In: Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI Press, 16-21.
- Maes, Pattie (1990), 'How to do the right thing', Connection Science, 1:3.
- McCauley, Thomas L. and Stan Franklin (1998) An Architecture for Emotion, AAAI Fall Symposium "Emotional and Intelligent: The Tangled Knot of Cognition"
- Pepperberg, I. M. (1994), Numerical Competence in an African Grey Parrot. J. Comp. Psych 108, 36-44.
- Ramamurthy, Uma , Myles Bogner, and Stan Franklin (1998), "Conscious" Learning In An Adaptive Software Agent, From Animals to Animats 5, 372-377.
- Sloman, A. (1996) What Sort of Architecture is Required for a Human-like Agent? Cognitive Modeling Workshop , AAAI96, Portland Oregon.
- Zhang, Zhaohua, Stan Franklin and Dipankar Dasgupta (1998), Metacognition in Software Agents using Classifier Systems, Proc AAAI 98, 82-88