A “Consciousness” Based Architecture for a Functioning Mind

Stan Franklin

Institute for Intelligent Systems and
Department of Mathematical Sciences
The University of Memphis
stan.franklin@memphis.edu

Abstract

Here we describe an architecture designed to accommodatemultiple aspects of human mental functioning. In a roughly star-shaped configuration centered on a “consciousness” module, the architecture accommodates perception, associative memory, emotions, action-selection, deliberation, language generation, behavioral and perceptual learning, self-preservation and metacognition modules. The various modules (partially) implement several different theories of these various aspects of cognition. The mechanisms used in implementing the several modules have been inspired by a number of different “new AI” techniques. One software agent embodying much of the architecture is in the debugging stage (Bogner et al. in press). A second, intending to include all of the modules of the architecture is well along in the design stage (Franklin et al. 1998). The architecture, together with the underlying mechanisms, comprises a fairly comprehensive model of cognition (Franklin & Graesser 1999). The most significant gap is the lack of such human-like senses as vision and hearing, and the lack of real-world physical motor output. The agents interact with their environments mostly through email in natural language.

The “consciousness” module is based on global workspace theory (Baars 1988, 1997). The central role of this module is due to its ability to select relevant resources with which to deal with incoming perceptions and with current internal states. Its underlying mechanism was inspired by pandemonium theory (Jackson 1987).

The perception module employs analysis of surface features for natural language understanding (Allen 1995). It partially implements perceptual symbol system theory (Barsalou 1999), while its underlying mechanism constitutes a portion of the copycat architecture (Hofstadter & Mitchell 1994).

Within this architecture the emotions play something of the role of the temperature in the copycat architecture and of the gain control in pandemonium theory. They give quick indication of how well things are going, and influence both action-selection and memory. The theory behind this module was influenced by several sources (Picard 1997, Johnson 1999, Rolls 1999). The implementation is via pandemonium theory enhanced with an activation-passing network.

The action-selection mechanism of this architecture is implemented by a major enhancement of the behavior net (Maes 1989). Behavior in this model corresponding to goal contexts in global workspace theory. The net is fed at one end by environmental and/or internal state influences, and at the other by fundamental drives. Activation passes in both directions. The behaviors compete for execution, that is, to become the dominant goal context.

The deliberation and language generation modules are implemented via pandemonium theory. The construction of scenarios and of outgoing messages are both accomplished by repeated appeal to the “consciousness” mechanism. Relevant events for the scenarios and paragraphs for the messages offer themselves in response to “conscious” broadcasts. The learning modules employ case-based reasoning (Kolodner 1993) using information gleaned from human correspondents. Metacognition is based on fuzzy classifier systems (Valenzuela-Rendon 1991).

As in the copycat architecture, almost all of the actions taken by the agents, both internal and external, are performed by codelets. These are small pieces of code typically doing one small job with little communication between them. Our architecture can be thought of as a multi-agent system overlaid with a few, more abstract mechanisms. Altogether, it offers one possible architecture for a relatively fully functioning mind. One could consider these agents as early attempts at the exploration of design space and niche space (Sloman 1998).

Autonomous 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. 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.

Global Workspace Theory

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

In his global workspace theory, Baars, along with many others (e.g. (Minsky 1985, Ornstein 1986, Edelman 1987)) , postulates that human cognition is implemented by a multitude of relatively small, special purpose processes, almost always unconscious. (It's a multiagent system.) 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 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.

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 an autonomous 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 cognitive theory and intelligent software. Minds can be viewed as control structures for autonomous agents (Franklin 1995). A theory of mind constrains the design of a "conscious" agent that implements 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. Moreover, every design decision taken during an implementation furnishes 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 1997).

"Conscious" Mattie

"Conscious" Mattie (CMattie) is a "conscious" clerical software agent (McCauley & Franklin 1998, Ramamurthy et al. 1998, Zhang et al. 1998, Bogner et al. in press) . 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 extensions of Maes' behavior nets (1989) , Hofstadter and Mitchell's Copycat architecture (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 et al. 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. 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 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 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 attention codelets who recognize novel or problematic situations (Bogner 1999, Bogner et al. 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.

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 select 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 job requisitions. Next might be a behavior that runs information concerning 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.

Deliberation is also used in IDA to implement voluntary action in the form of William James' ideomotor theory as prescribed by global workspace theory. Suppose scenarios have been constructed for several of the more suitable jobs. An attention codelet spots one that it likes, possibly due to this codelets predilection for low moving costs. The act of bring these candidate to consciousness serves to propose it. This is James' idea popping into mind. If now other attention codelet brings an objection to conscious or proposes a different job. A codelet assigned the particular task of deciding will conclude, after a suitable time having passed, that the proposed job will be offered and starts the process by which it will be so marked in working memory. Objections and proposals can continue to come to consciousness, but the patience of the deciding codelet dampens as time passes. Several jobs may be chosen with this process.

IDA's language generation module follows the same back and forth to "consciousness" routine. For example, in composing a message offering a sailor a choice of two billets, an attention codelet would bring to "consciousness" the information that this type of message was to be composed and the sailor's name, pay grade and job description. After the "conscious" broadcast and the involvement of the behavior net as described above, a script containing the salutation appropriate to a sailor of that pay grade and job description would be written to the working memory. Another attention codelet would bring this salutation to "consciousness" along with the number of jobs to be offered. The same process would result in an appropriate introductory script being written below the salutation. Continuing in this manner filled in scripts describing the jobs would be written and the message closed. Note that different jobs may require quite different scripts. The appeal to "consciousness" results in some version of a correct script being written.

The mediation by the "consciousness" mechanism, as described in the previous paragraphs 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, a human detailer would be conscious of each item individually. Hence, according to our principle, so must IDA be "conscious" of each retrieved personnel data item.

These agents are also intended to learn in several different ways. In addition to learning via associative memory as described above, IDA also learns via Hebbian temporal association. Codelets that come to "consciousness" simultaneously increase there associations. The same is true to a lessor extent when they are simply active together. Recall that these associations provide the basis coalition formation. Other forms of learning include chunking, episodic memory, perceptual learning, behavioral learning and metacognitive learning. The chunking manager gathers highly associated coalitions of codelets in to a single "super" codelet in the manner of concept demons from pandemonium theory (Jackson 1987) , or of chunking in SOAR (Laird et al. 1987). IDA's episodic memory is cased based in order to be useful to the perceptual and behavior modules that will learn new concepts (Ramamurthy et al. 1998), and new behaviors (Negatu & Franklin 1999) from interactions with human detailers. For example, CMattie might learn about a new piece of sonar equipment and the behaviors appropriate to it. Metacognitive learning employs fuzzy classifier systems (Valenzuela-Rendon 1991).

Conclusions

Here I hope to have described an architecture capable of implementing many human cognitive functions within the domain of a human information agent. I'd hesitate to claim that this architecture, as is, is fully functioning by human standards. It lacks, for instance, the typical human senses of vision, olfaction, audition, etc. Its contact with the world is only through text. These only the most rudimentary sensory fusion by the agents. They lack selves, and the ability to report internal events. There's much work left to be done.

Acknowledgements

This research was supported in part by ONR grant N00014-98-1-0332, and was produced with essential contributions from the Conscious Software Research Group including Art Graesser, Satish Ambati, Ashraf Anwar, Myles Bogner, Arpad Kelemen, Ravikumar Kondadadi, Irina Makkaveeva, Lee McCauley, Aregahegn Negatu, Hongjun Song, Allexei Stoliartchouk, Uma Ramamurthy, and Zhaohua Zhang.

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. Ph.D. Dissertation. University of Memphis.

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

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

Franklin, S. 1995. *Artificial Minds*. Cambridge MA: MIT Press.

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

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.

Hofstadter, D. R., and M. Mitchell. 1994. The Copycat Project: A model of mental fluidity and analogy-making. In *Advances in connectionist and neural computation theory, Vol. 2: logical connections*, ed. K. J. Holyoak, and J. A. Barnden. 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.

Johnson, V. S. 1999. *Why We Feel: The Science of Human Emotions*. Reading, MA: Perseus Books.

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

Kolodner, J. 1993. *Case-Based Reasoning*. : Morgan Kaufman.

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

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

Maturana, R. H., and F. J. Varela. 1980. *Autopoiesis and Cognition: The Realization of the Living, Dordrecht*. Netherlands: Reidel.

Maturana, H. R. 1975. The Organization of the Living: A Theory of the Living Organization. *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.

Minsky, M. 1985. *The Society of Mind*. New York: Simon and Schuster.

Negatu, A., and S. Franklin; 1999. Behavioral learning for adaptive software agents. Intelligent Systems: ISCA 5th International Conference; International Society for Computers and Their Applications - ISCA; Denver, Colorado; June 1999.

Ornstein, R. 1986. *Multimind*. Boston: Houghton Mifflin.

Picard, R. 1997. *Affective Computing*. Cambridge MA: The MIT Press.

Ramamurthy, U., S. Franklin, and A. Negatu. 1998. Learning Concepts in Software Agents. In *From animals to animats 5: Proceedings of The Fifth International Conference on Simulation of Adaptive Behavior*, ed. R. Pfeifer, B. Blumberg, J.-A. Meyer , and S. W. Wilson. Cambridge,Mass: MIT Press.

Rolls, E. T. 1999. *The Brain and Emotion*. Oxford: Oxford University Press.

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

Sloman, A. 1998. The ``Semantics'' of Evolution: Trajectories and Trade-offs in Design Space and Niche Space. In *Progress in Artificial Intelligence*, ed. H. Coelho. Berlin: Springer.

Valenzuela-Rendon, M. 1991. *The Fuzzy Classifier System: a classifier System for Continuously Varying Variables. In: Proceedings of the Fourth International Conference on Genetic Algorithms*. San Mateo CA: Morgan Kaufmann.

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