An Autonomous Software Agent for Navy Personnel Work: a Case Study

Stan Franklin

Computer Science Department and the Institute for Intelligent Systems
University of Memphis
Memphis, TN, 38152, USA
franklin@memphis.edu

Abstract

IDA is an autonomous software agent whose task is to assign a sailor to a new tour of duty at the end of the old. Her task input is most often initiated by an email from the sailor; thereafter she acts completely autonomously. The task requires IDA to access databases, to deliberate, to perform complex constraint satisfaction, and to negotiate with the sailor in natural language. IDA's architecture and mechanisms are motivated by a variety of computational paradigms and implement a number of cognitive models including the Global Workspace Theory of consciousness. IDA is able to interact in depth with sailors in a relatively complex real-world environment involving job requirements, Navy policy, location, travel, training, dates, costs, preferences, etc. Here we briefly describe the IDA technology and her interactions as a case study of the interaction of sophisticated software agents with humans.

IDA and her task

IDA (Intelligent Distribution Agent) is a "conscious" software agent that was 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, understanding the content and producing life-like responses. Sometimes she will initiate conversations. She must access several, quite different, databases, again understanding the content. She must see that the Navy's needs are satisfied by adhering to some sixty policies and seeing that job requirements are fulfilled. 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 reach agreement with the sailor as to the new assignment, if at all possible. If not, she assigns.

IDA's Interaction with Sailors

As does a human detailer, IDA was designed to interact with a particular community of sailors characterized by a common small set of jobs and a small range of pay grades. Most communities contain from 500 to 2500 sailors. One to three or four detailers serve a community. IDA's community is comprised of aircraft technicians.

IDA is currently capable of automating the entire set of tasks of a human detailer, with the exception of writing the final orders. The original plan was to add this feature and to eventually deploy an appropriately knowledge engineered version of IDA for each community. Two years into this five year project, the Navy redirected the project toward a multi-agent system in which each of more than 300,000 sailors and each command would have it own, individual agent. The plan was to complete IDA and then to use the IDA technology in the design of the multi-agent system. IDA is complete, for these purposes, and a year's work on the multi-agent system has produced its first demonstrable product.

As a consequence of this change of plans, we have no online interactions between IDA and sailors actively seeking reassignment to report. We do, however, have testing interactions between IDA and human detailers in the role of such sailors. In the following paragraphs such interactions are described.

Most often the sailor initiates the correspondence. If the sailor's projected rotation date is much within a nine-month window, IDA will initiate. In a typical correspondence the sailor's first message will contain his or her name, social security number and paygrade, and will ask for reassignment. Often some preference for a specific location, a particular piece of equipment, additional training, even a particular assignment, etc. will be expressed.

Using name and social security number from the sailor's message, IDA is able to access pertinent data from the sailor's personnel record as maintained by the Navy. She then would typically consult the Navy's job requisition database performing a course search, using hard constraints, to find a short list of candidate jobs. Each of these jobs would be evaluated by IDA's constraint satisfaction module, and a fitness value assigned. The sailor's preferences, the job requirements, and the Navy's policies would all be considered.

Starting with the most fit job, IDA would create a temporal scenario with the sailor leaving his or her current job on a certain date within the appropriate time window. After spending some time on leave, some in travel, possibly some in training, the sailor would be expected to arrive at the new job on a calculated date. If this date is within a given time window for the job, the gap would be zero. For each month on either side of the window the gap

is one more. With an unacceptable gap IDA might begin again with a different starting date. After several unsuccessful tries, she might give up on this particular job.

As successful scenarios appear in the workspace, attention codelets begin "consciously" negotiating about which job or jobs to offer. When a conclusion is reached, IDA composes an email message to the sailor offering the selected job(s), and sends it.

Soon after, the negotiation between IDA and the sailor begins. For example, the sailor may decline the offered jobs and requeste another specific job. On receipt of the sailor's message, IDA must first establish a context for it. Then she would go through the constraint satisfaction and scenario building processes for this job and, "consciously" decide whether to assign the sailor to the desired job. If the decision is no, IDA will compose a message to the sailor saying so and giving reasons. She might offer some other job.

Other negotiating ploys include asking for a job at a particular base or, say, on the West Coast. Yet another is to ask to wait for the next job requisition list. In each such case, IDA must find the context, evaluate the request, decide on a response, and write persuasively to the sailor. Eventually, either the sailor accepts an offered job or IDA assigns one. The Navy wants to keep the latter cases to a minimum. A negotiation session with a particular sailor may take a half-dozen messages back and forth.

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 an autonomous software agent, designed for a specific task, and 'living' in a real world computing and network system.

Global Workspace Theory

The material in this section is from Baars' two books (1988, 1997) and superficially describes his global workspace (GW) theory of consciousness. In this 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 thus 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 situation, or in solving the current problem. Thus consciousness in this theory allows us to deal with novel 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 activity of processors 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. A key insight of GW theory says that each context is, in fact, a coalition of codelets.

We refer to software agents that implement GW theory as *"conscious" software agents*.

IDA's Architecture and Mechanisms

Table 1 succinctly summarizes IDA's various cognitive modules, the sources of inspiration for their mechanisms, and the various theories that they implement. A few of these modules, while part of the IDA conceptual model, have not yet been implemented. These will not be described below.

Perception

IDA senses only strings of characters Her perception 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 (Zhang et al. 1998b) 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 perception system for IDA, allowing her to recognize, categorize and understand. IDA must also perceive the contents read from databases, a much easier task. The contents of perception are written to working memory before becoming "conscious".

Memory

IDA employs sparse distributed memory (SDM) (Kanerva 1988) as her major associative memory (Anwar and Franklin. to appear). SDM is a content addressable memory that, in many ways, is an ideal computational mechanism for use both as a transient episodic memory (TEM) (Baars and Franklin. Forthcoming, Conway 2001) and as a long-term associative memory (LTM). Any item written to the workspace immediately cues a retrieval from TEM and LTM, returning prior activity associated with the current entry.

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 TEM and 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* by Kanerva (1988)) is set aside as an interface with TEM and LTM. Retrievals from both TEM and 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 written to TEM.

"Consciousness"

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 1999, 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 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. 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. It contents are also written to TEM.

Later, offline, the undecayed contents of TEM are consolidated into LTM.

IDA Module	Computational Mechanism motivated by	Theories Accommodated
Perception	Copycat architecture (Hofstadter & Mitchell 1994)	Perceptual Symbol Systems (Barsalou 1999)
Working Memory		(Andrade 2001),Long-term Working Memory (Ericsson & Kintsch 1995)
Emotions	Neural Networks (Rumelhart & McCellland 1982)	(Damasio 1999, Rolls 1999)
Associative Memory	Sparse Distributed Memory (Kanerva 1988)	
Episodic Memory	Sparse Distributed Memory (Kanerva 1988)	(Conway 2001, Shastri 2002)
"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, Baars 1988)
Language Generation	Pandemonium Theory (Jackson 1987)	
Metacognition	Fuzzy Classifers (Valenzuela-Rendon 1991)	Human-Like Agent Architecture (Sloman 1999)

Table 1. IDA's modules, inspiration for their mechanisms, and theories implemented

Action Selection

IDA depends on a behavior net (Maes 1989, Negatu & Franklin 2002) for high-level action selection in the service of built-in drives. She has several distinct drives operating in parallel that vary in urgency as time passes and the environment changes. Behaviors are typically mid-level actions, corresponding to goal contexts in GW theory, many depending on several behavior codelets for their execution A behavior looks very much like a production rule, having preconditions as well as additions and deletions. Each behavior occupies a node in a digraph called a behavior net that is composed of behaviors and their various links, which are completely determined by the behaviors.

As in connectionist models (McClelland et al. 1986), this digraph spreads activation. The activation comes from activation 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 every behavior that will satisfy it by being active. Certain internal states of the agent can also send activation to the behavior net. This activation, for example, might come from codelets responding to a "conscious" broadcast. Finally, activation spreads from behavior to behavior along the various links.

Behavior-priming codelets, responding to a "conscious" broadcast, instantiate a behavior stream, bind appropriate variables in its behaviors, and send activation to relevant behaviors A behavior is *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. Her behavior net produces flexible, tunable action selection for IDA.

Constraint Satisfaction

IDA is provided with a constraint satisfaction module (Kelemen et al. 2002 in press) 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.

Deliberation

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, Franklin 2000, Kondadadi & Franklin 2001). In considering a possible jobs for a sailor, she must construct a temporal scenario 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.

Voluntary Action Selection

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 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), which Baars incorporated into his GW theory (1988, Chapter 7). 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. The IDA model furnishes an underlying mechanism that implements the ideomotor theory of volition and its architecture in a software agent (Franklin 2000).

Suppose that at least one temporal scenario has been successfully constructed in the workspace as described above.. 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 by another attention codelet in a timely fashion, it will not do so.

Two proposing attention codelets may alternatively propose the same two jobs several times. There are several mechanisms in place that 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, in principle, 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.

Conclusion

The IDA technology, with additional knowledge engineering, is capable or automating the tasks of any human information agent (Franklin 2001). Such human information agents include insurance agents, travel agents, voter registrars, mail-order service clerks, telephone information operators, employment agents, AAA route planners, customer service agents, bank loan officers, and many, many others. Such human agents must typically possess a common set of skills. These would often include most of the following:

- Communicating with clients in their natural language;
- Reading from and writing to databases of various sorts (insurance rates, airline schedules, voter roles, company catalogs, etc.);
- Knowing, understanding and adhering to company or agency policies;
- Planning and decision making (coverage to suggest, routes and carriers to offer, loan to authorize, etc);
- Negotiating with clients about the issues involved;
- Generating a tangible product (insurance policy, airline tickets, customer order, etc.).

In these cases, voice recognition software would have to be included as a front end to IDA. Depending on the amount of knowledge that must be added, it might prove necessary to wait for the next generation of desktop workstations. The currently running IDA requires the most powerful processor generally available, and an almost full complement of RAM. The only additional human-computer interaction concerns would be to insure that IDA produced appropriately worded messages. This would be of more concern in a civilian setting than in the military. One wouldn't want IDA to send an angry message to a customer.

Acknowledgements

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

References

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

Andrade, J. 2001. *Working Memory in Perspective*. New York: Psychology Press.

Anwar, A., and S. Franklin. to appear. Sparse Distributed Memory for "Conscious" Software Agents. *Cognitive Systems Research*.

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.

Baars, B. J., and S. Franklin. submitted. How conscious events may interact with Working Memory: Using IDA, a large-scale model of Global Workspace theory. *Trends in Cognitive Science*.

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

Conway, M. A. 2001. Sensory-perceptual episodic memory and its context: autobiographical memory. In *Episodic Memory*, ed. A. Baddeley, M. Conway, and J. Aggleton. Oxford: Oxford University Press.

Damasio, A. R. 1999. *The Feeling of What Happens*. New York: Harcourt Brace.

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. 2000. Deliberation and Voluntary Action in 'Conscious' Software Agents. *Neural Network World* 10:505–521.

Franklin, S. 2001. Automating Human Information Agents. In *Practical Applications of Intelligent Agents*, ed. Z. Chen, and L. C. Jain. Berlin: Springer-Verlag.

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., 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.

Jackson, J. V. 1987. Idea for a Mind. *Siggart* Newsletter, 181:23–26.

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.

Kelemen, A., Y. Liang, R. Kozma, and S. Franklin. 2002 in press. Optimizing Intelligent Agent's Constraint Satisfaction with Neural Networks. In *Innovations in Intelligent Systems*, ed. A. Abraham, and B. Nath. Heidelberg, Germany: Springer-Verlag,.

Kondadadi, R., and S. Franklin. 2001. A Framework of Deliberative Decision Making in "Conscious" software Agents. In *Proceedings Of Sixth International Symposium on Artificial Life and Robotics (AROB-01)*.

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.

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

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

Negatu, A., and S. Franklin. 2002. An action selection mechanism for 'conscious' software agents. *Cognitive Science Quarterly* 2.

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

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

Rumelhart, D. E., and J. L. McCellland. 1982. *Parallel Distributed Processing*, vol. 1. Cambridge, Mass.: MIT Press.

Shastri, L. 2002. Episodic memory and cortico-hippocampal interactions. *Trends in Cognitive Sciences* 6:162–168.

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.

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. 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*.