Position Paper—AAAI Fall Symposium on Dialogue Systems for Health Communication—October 22-24 2004, Washington DC

A Triage Information Agent (TIA) based on the IDA Technology

Stan Franklin and Dan Jones, M.D.
Institute for Intelligent Systems
The University of Memphis
Memphis, TN 38152 USA
franklin@memphis.edu, djones@nwaft.com

Abstract

Busy hospital emergency rooms are concerned with shortening the waiting times of patients, with relieving overburdened physicians, and with reducing the number of mistakes made by triage nurses. Here we propose a software agent, dubbed TIA (Triage Information Agent) that, via dialogue in English, would gather both logistical and medical information from a patient for later use by the triage nurse. TIA would also give tentative, possible diagnoses to the triage nurse, along with recommendations for non-physician care. The IDA Technology makes a software agent such as TIA feasible, at least in principle.

Introduction

With waiting times in busy hospital emergency rooms measured in hours, hospital administrators are looking for ways to shorten them. Overwhelmed triage nurses often make mistakes, sometimes leading to malpractice suits. Though a solution for neither, a software Triage Information Agent (TIA) could help alleviate both problems. After a triage nurse rules out an immediately life-threatening situation, a patient would engage in a dialogue with TIA, who would then pass information about the patient's condition (chief complaint and differential diagnoses) and recommendations for prioritization and non-physician care to the nurse. After perusing the information and suggestions from TIA, the nurse would further observe and interview the patient as needed before making the appropriate decisions.

Originally developed for personnel work for the U. S. Navy, the IDA technology permits the automation of human information agents, that is, of the daily tasks of people who dialogue with clients, consult databases, adhere to company policies, make decisions, and produce text-based products (Franklin 2001). This would include travel agents, customer service agents, loan officers, and insurance agents. The IDA technology will also allow the development of TIA. TIA is envisioned as a conversational, decision making agent without an on-screen avatar (Cassell and Vilhjálmsson 1999; Traum and Rickel 2002).

IDA

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, by understanding the content and producing life-like responses. Sometimes she will initiate conversations. She must access several databases, again understanding the content. She must see that the Navy's needs are satisfied by adhering to some ninety policies 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 write the orders and start them on the way to the sailor. At this writing an almost complete version of IDA is up and running and had been demonstrated and tested to the satisfaction of the Navy.

The IDA Technology

The IDA Technology is based on a number of highly connected modules each built on its distinct mechanism. Most of these are up and running. A few are still being developed, and a couple are designed but not yet implemented. Figure 1 portrays these interconnections.

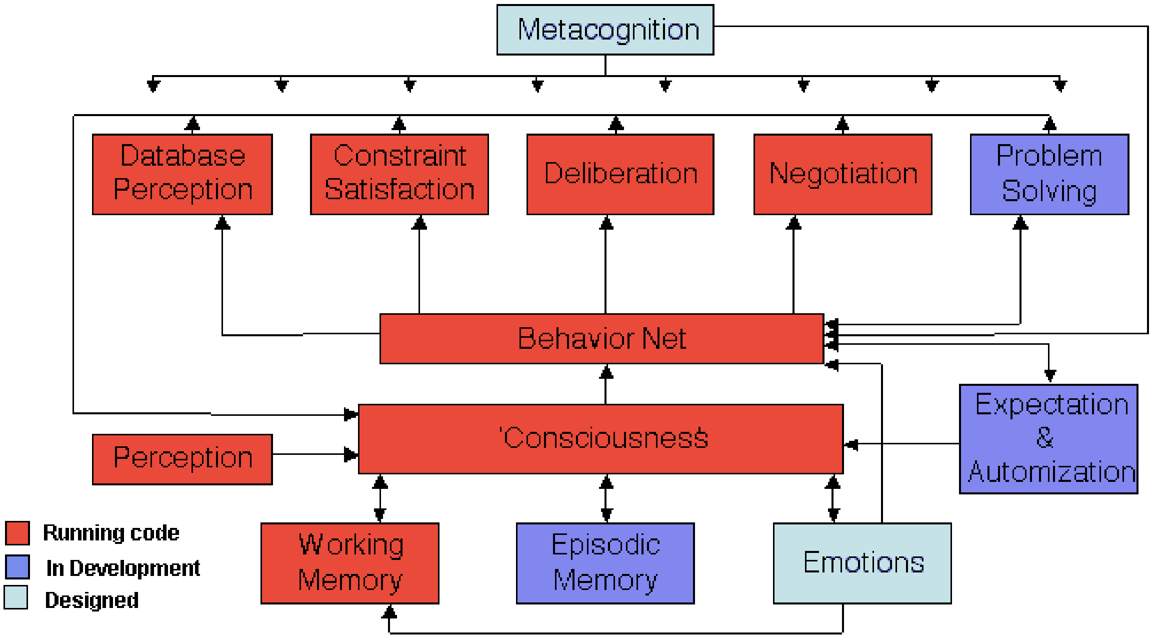

Figure 1. The IDA Architecture

Following Hofstadter’s terminology (see below) a codelet is a special purpose, relatively independent, mini-agent typically implemented as a small piece of code running as a separate thread. IDA depends heavily on such codelets for almost every module. In what follows we will encounter several different types of codelets such as perceptual codelets, attention codelets, information codelets, behavior codelets and language generation codelets. Many codelets play the role of demons (as in an operating system) waiting patiently for the conditions under which they can act. Some codelets subserve some higher-level construct, while others act completely independently.

In this section we describe several of the IDA modules that would play a role in TIA.

Perception

Perception in IDA consists mostly of processing incoming email messages in natural language (Zulandt Schneider et al. 2001). 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. We hypothesize that much of human language understanding results from a combined bottom up/top down passing of activation through a hierarchical conceptual net with the most abstract concepts in the middle.

Thus IDA’s language-processing module has been implemented as a Copycat-like architecture with perceptual codelets that are triggered by surface features and a slipnet (Hofstadter & Mitchell 1994), a semantic net that passes activation. The slipnet stores domain knowledge. In addition there’s a pool of perceptual codelets specialized for recognizing particular pieces of text, and production templates used by codelets for building and verifying understanding. Together they constitute an integrated sensing system for IDA, allowing her to recognize, categorize and understand.

It’s important to be clear about what is claimed by the work “understand” as used in the previous sentence. An example may help. A secretary sending out an email announcement of an upcoming seminar on Compact Operators on Banach Spaces can be said to have understood the organizer’s request that she do so even though she has no idea of what a Banach space is much less what compact operators on them are. In most cases it would likely require person years of diligent effort to impart such knowledge. Nonetheless, the secretary understands the request at a level sufficient for her to get out the announcement. In the same way IDA understands incoming email messages well enough to do all the things she needs to with them. An expanded form of this argument can be found in *Artificial Minds* (Franklin 1995). Glenberg also makes a similar argument (1997).

IDA must also perceive the contents read from databases, a much easier task. An underlying assumption motivates our design decisions about perception. Suppose, for example, that IDA receives a message from a sailor saying that his projected rotation date (PRD) is approaching and asking that a job be found for him. The perception module would recognize the sailor's name and social security number, and that the message is of the please-find-job type. This information would then be written to the workspace. The general principle here is that the contents of perception are written to working memory before becoming conscious.

Workspace

IDA solves routine problems with novel content. This novel content goes into her workspace, which roughly plays the same role as human working memory. Perceptual codelets write to the workspace as do other, more internal codelets. Quite a number of codelets, including attention codelets (see below) watch what's written in the workspace in order to react to it. Part, but not all, the workspace, called the *focus*[1], by Kanerva (1988)) is set aside as an interface with long-term LTM. Retrievals from LTM are made with cues taken from the focus and the resulting associations are written to other registers in the focus. The contents of still other registers in the focus are stored in (written to) associative memory as we will see below. Items in the workspace decay over time, and may be overwritten. Not all of the contents of the workspace eventually make their way into consciousness.

Associative memory

IDA employs sparse distributed memory (SDM) as her major associative memory (Kanerva 1988, Anwar & Franklin 2003). SDM is a content addressable memory that, in many ways, is an ideal computational mechanism for use as a long-term associative memory (LTM). Any item written to the workspace cues a retrieval from LTM, returning prior activity associated with the current entry. LTM is accessed as soon as information reaches the workspace, and the retrieved associations will be also written to the workspace.

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

Consciousness mechanism

The apparatus for "consciousness" consists of a coalition manager, a spotlight controller, a broadcast manager, and a collection of attention codelets whose job it is to bring appropriate contents to "consciousness" (Bogner et al. 2000). Each attention codelet keeps a watchful eye out for some particular occurrence that might call for "conscious" intervention. In most cases the attention codelet is watching the workspace, which will likely contain both perceptual information and data created internally, the products of "thoughts." Upon encountering such a situation, the appropriate attention codelet will form a coalition with the small number of information codelets that carry the information describing the situation. This association should lead to the collection of this small number of information 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. If or when successful, its contents will be broadcast. Broadcast contents are also stored in (written to) associative memory as the contents of "consciousness" should be.

Action selection (decision making)

IDA depends on a behavior net (Maes 1989, Negatu & Franklin 1999) for high-level action selection in the service of built-in drives. She has several distinct drives operating in parallel. These drives vary in urgency as time passes and her environment changes. Behaviors are typically mid-level actions, many depending on several behavior codelets for their execution. A behavior net is composed of behaviors, corresponding to goal contexts in GW theory, and their various links. A behavior looks very much like a production rule, having preconditions as well as additions and deletions. It's typically at a higher level of abstraction often requiring the efforts of several codelets to effect its action. A behavior can be thought of as the collection of its codelets (processors) in accordance with global workspace theory. Each behavior occupies a node in a digraph. The three types of links, successor, predecessor and conflictor, of the digraph are completely determined by the pre- and post-condition of its behaviors (Maes 1989).

As in connectionist models (McClelland et al. 1986), this digraph spreads activation. The activation comes from that stored in the behaviors themselves, from the environment, from drives, and from internal states. The more relevant a behavior is to the current situation, the more activation it is going to receive from the environment. Each drive awards activation to those behaviors that will satisfy it. Certain internal states of the agent can also send activation to the behavior net. One example might be activation from a coalition of codelets responding to a "conscious" broadcast. Activation spreads from behavior to behavior along both excitatory and inhibitory links and a behavior is chosen to execute based

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

on activation. Her behavior net produces flexible, tunable action selection for IDA. As is widely recognized in humans the hierarchy of goal contexts is fueled at the top by drives, that is, by primitive motivators, and at the bottom by input from the environment, both external and internal.

The broadcast is received by appropriate behavior codelets who know to instantiate a behavior stream in the behavior net for dealing with the current situation. They also bind appropriate variables, and send activation to appropriate behaviors. If or when a particular behavior is chosen to be executed, behavior codelets associated with it jump into action each performing its task.

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

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). In her original domain, IDA constructs a list of a number of possible jobs in her workspace, together with their fitness values. She must construct a temporal scenario for at least a few of these possible billets to see if the timing will work out (say if the sailor can be aboard ship before the departure date). In each scenario the sailor leaves his or her current post during a approved 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 within 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. They are constructed in the workspace by the process proceeding from attention codelets, to "consciousness," to behavior net, to behavior codelets, as described previously.

Negotiation

After IDA has selected one or more jobs to be offered to a given sailor, her next chore is to negotiate with the sailor until one job is decided upon. The US Navy is quite concerned about retention of sailors in the service. This depends heavily on the sailor's job satisfaction. Thus the Navy gives a high priority to the assignment of a job that both satisfies the sailor's preferences and offers opportunity for advancement, sometimes including additional training. Whenever possible the final job assignment is made with the sailor's agreement. IDA must negotiate this agreement with the sailor.

When the initial job offerings are made the sailor may respond in several different ways. He may accept one of the jobs offered. He may decline all of them and request some different job assignment. He may ask for a particular job not among those offered. He may ask that the process be postponed until a new requisition list appears, hoping to find something more to his liking. IDA may accede to or deny any of these requests, the decision often dependent on time constraints and/or the needs of the service. The continuing negotiations offer many possible paths. It ends with one job being assigned to the sailor, most often with his agreement, but sometimes without.

IDA must be able to carry out such negotiations. This requires making decisions and responding to the sailor's messages.

There's more to the IDA architecture and mechanisms, but this is all that space will allow.

Description of TIA

The high-lever goals of TIA are (1) to shorten total patient time in the Emergency Department, and (2) to decrease triage-related errors and malpractice risk. These can both be reduced to more specific subgoals:

1. Shorten wait time for commencement of care. In most busy ED's, the bottleneck resource is physician time. If the time required for accurate triage is reduced from 10 minutes to 2 minutes, nothing is gained if the patient still has to wait a total of 3 hours for a physician to become available. So to shorten wait times, TIA must either (a) decrease the amount of time a physician spends with patients (on average); or decrease the average time spent waiting for non-physician care (e.g., nursing procedures, lab tests, x-rays, etc.). The obvous low-lying fruit here is to have TIA initiate routine non-physician care actions based on specific criteria.

   The most common ED patient is probably a '2-step' patient: the physician sees the patient, creates a 'differential diagnosis' (list of possible causes of the patient's problem), and orders specific tests or procedures to be done. That's step 1. Then, maybe two hours later, the nurse (or status board) notifies the physician that the patient's tests and procedures have been completed. At that time, the physician sees the patient again, reviews the test or procedure results, and arrives at a provisional diagnosis and disposition for the patient. That's step 2. Some patients ('quickies') only require one step, and some require three or more, but two it probably most typical.

   An effective triage system, such as TIA, could effectively convert many or most 2-step patients to 1-step patients by automatically triggering orders for specific tests and procedures based on triage information. For example, a patient with a sore throat and fever should have a 'Strep screen'; and a patient with cough and fever plus chest pain or shortness of breath needs a chest x-ray. The best ED triage nurses become reasonably fluent at recognizing and 'pre-ordering' only the most obvious such tests and procedures. TIA could accomplish pre-ordering of needed tests and procedures much more comprehensively and consistently than most triage nurses; and (in the physician author's opinion) more efficiently and effectivly than most physicians, who tend to be inconsistent, frequently omitting important tests and often ordering unnecessary ones

2. Decrease triage-related errors and malpractice risk. In general, error reduction equates to malpractice risk reduction. TIA can reduce errors of two types:
    a) Errors of delay. In a busy ED, resources are limited and there is an unavoidable 'average wait' for patients to obtain care. The primary function of triage is to sort patients according to their 'urgency' (need for early or immediate attention to avoid death, disability or suffering). The best triage nurses (with years of experience) become 'reasonably good' at recognizing which patients need urgent attention, and which can wait. By consistently recognizing an unlimited number of prioritization criteria, TIA can lend greater reliability and consistency to the prioritization function.
    b) Errors of oversight or omission. One of the most common causes of physician malpractice is failure of the tired, fatigued or overworked physician to formulate a reasonable 'differntial diagnosis' (list of possible causes), and to obsessively test further for the most serious possibilities. There is a constant human tendency, when fatigued and/or under time pressure, to focus in on the most obious or likely diagnosis, and ignore or overlook less likely but more serious causes. By consistently recognizing and flagging for physician attention the most serious causes of specific patient complaints, TIA can reduce physician oversights, thereby increasing the quality of care and reducing malpractice risk.

Issues of Concern

Although TIA can be expected to reduce waiting times to be seen by a triage nurse, by reducing the time a triage nurse spends with each patient, a triage nurse is still going to have to take the vital signs and recognize immediately life-threatening situations. That's about all most triage nurses do, other than enter a 'chief complaint' and patient demographics. These can be off-loaded to TIA, perhaps reducing nurse time per patient.

The real savings of time will come from converting two-step patients to one-step patients. TIA can add consistent triggering of early ordering of tests and procedures, and consistent warning regarding consistent diagnostic possibilities, and those functions can reduce overall time in the emergency room, as well as reducing errors.

We expect that TIA will gather information from patients by conversing with them in colloquial English using voice recognition and speech synthesizing. Though the quality of such voice recognition an speech synthesizing systems are steadily improving, various parts of this country are becoming more and more bilingual, which may create problems. Errors in voice recognition could often be handled conversationally by TIA as is done by humans. Still, voice recognition would be the likely stumbling block for an implemented and fielded TIA in the near future. Concerns about the early feasibility of a natural language interface directly between TIA and patient arise for multiple reasons: the patient's age, degree of incapacity due to illness/stress, intelligence level, English fluency, etc., all vary tremendously; one's interview technique often has to be radically modified on the fly; a question often has to be asked three different ways to find one the patient understands; and the source of the information often moves around from the patient to the little brother to the mother and then the aunt, etc. All of these foreseen difficulties can, in principle, be overcome with our current technology as soon as the voice recognition becomes sufficiently reliable.

Knowledge engineering into TIA will require identifying the topics of conversation that TIA should broach. These will surely include patient identifying data and demographics, the 'chief complaint' and related symptoms, and qualifications thereof (e.g., the nature of the pain, duration of symptoms, etc.). The cost of such knowledge engineering for the TIA system should be on the order of magnitude of ten person years or $1,000,000. The gains, spread over hundreds of emergency rooms, should surely justify this cost.

Will patients willingly interact with TIA? This will depend on TIA's ease of use, and on the patient's perception of TIA's benefits. Some educational effort may well be needed. Still, people are becoming more accustomed to, and more comfortable with, dealing with software agents of various types. Thus patients' willingness shouldn't be a major problem.

References

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

Anwar, A., and S. Franklin. 2003. Sparse Distributed Memory for "Conscious" Software Agents. *Cognitive Systems Research* 4:339–354.

Cassell, J., and H. Vilhjálmsson. 1999. Fully Embodied Conversational Avatars: Making Communicative Behaviors Autonomous. *Autonomous Agents and Multi-Agent Systems* 2:45-64.

Ericsson, K. A., and W. Kintsch. 1995. Long-term working memory. *Psychological Review* 102:211–245.

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

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

Glenberg, A. M. 1997. What memory is for. *Behavioral and Brain Sciences* 20:1–19.

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.

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.

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

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.

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

Traum, D., and J. Rickel. 2002. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*. New York: ACM Press.

Zulandt Schneider, R. A., R. Huber, and P. A. Moore. 2001. Individual and status recognition in the crayfish, Orconectes rusticus: The effects of urine release on fight dynamics. *Behaviour* 138:137–154.