Beyond the Electric Elves

Written by Craig Knoblock, Jose Luis Ambite, Hans Chalupsky, Yolanda Gil, Jerry Hobbs, Kevin Knight, and David Pynadath.

Technical Point of Contact: Craig Knoblock (knoblock@isi.edu)

USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

(310) 822-1511

Date of preparation: September 12, 2002

Innovative Claims

The operation of a human organization involves dozens of critical everyday tasks to ensure coherence in organizational activities, to monitor the status of such activities, to obtain information relevant to the organization, to keep everyone in the organization informed, etc. These activities are often well-suited for software agents, which can devote significant resources to perform these tasks, thus reducing the burden on humans. Such software agents enable organizations to act coherently, to attain their mission goals robustly, to react to crises swiftly, and to adapt to events dynamically.

Towards this vision, we previously developed a system called Electric Elves (Chalupsky, Gil, Knoblock et al. 2001, Ambite, Barish, Knoblock et al. 2002) which applies agent technology in service of the day-to-day activities. We have successfully deployed the Elves in both an office and travel environment. The travel application, called the Travel Elves, is in use today and provides a tremendously useful set of services for a traveler. In the process of building and deploying this successful system, we have learned a great deal about both its strengths and limitations. In general, the current state of agent technology means that agents today are carefully designed to work together, are only capable of responding to failures for which they were designed, cannot always explain their behavior, often require specialized interfaces to interact with them, and require significant effort to design and build.

In this white paper we propose a 3-5 year project that will result in the advances listed below, and we propose a seedling project of 3-4 months for designing the cognitive architecture that will be the foundation of such capabilities. Our long term goal is "intelligent agents" that robustly accomplish their tasks, responding appropriately to failures, communicate flexibly with humans and software agents, explain their behavior both on success or failure, dynamically compose new agents and behaviors from existing agents, rapidly build personalized agents without manual programming, and learn from their past experience to improve both robustness and performance.

The key innovations of our longer-term research will include:

- A framework for representing agent capabilities, goals, plans, actions, and histories that supports flexible and robust interactions among persistent agents.
- A uniform approach to communication among agents as well as between agents and humans that is more flexible than in previous systems. The communication will be close to and inspired by natural language facilitating people's communication with the agents.
- An integrated planning capability that supports the ability to accomplish tasks flexibly and to perform new tasks. This capability will be based on Planning by Rewriting techniques that we developed, which supports continuous planning.
- An integrated approach to autonomous learning and interactive instruction to build agents that adapt to their environment both through experience and direction. This work will be based on previous learning work on the Elves as well as other related projects.
- Agent construction tools that allow end users to create and add customized agents into the system without any programming. This work will extend on our earlier work on an agent wizard, which constructs new agents by leading a user through a series of simple questions about the task.

Technical Rationale

The Electric Elves project constitutes a successful proof of concept for a community of heterogeneous, interacting agents performing a useful class of tasks. We propose a long-term, 3- to 5-year effort to develop a more general, more flexible, and more widely applicable community of agents, and a 3- to 4-month seedling effort to develop the architecture required as the foundation for this. In this white paper we outline the key elements of the architecture, list the capabilities it would enable, and sketch how each of the capabilities would be achieved. In the project itself we would elaborate on this picture and begin to develop its most basic pieces.

The research will be driven by the following six goals:

1. Robust accomplishment of agents' tasks, with appropriate responses to failures.
2. Dynamic composition of new agents and behaviors from existing agents.
3. Flexible communication among humans and software agents.
4. Explanation of agents' behavior both on success and failure.
5. Rapid building of personalized agents without manual programming.
6. Agents or collections of agents learning from past experience.

Insofar as we can achieve these goals, the architecture will support a wide variety of agents as well as agents embedded in other systems, and a wide variety of tasks, as diverse as battlefield awareness, network management, and disaster response.

We will assume a heterogeneous collection of agents, each with specific capabilities and a level of self description that may vary from agent to agent. Two questions that must be answered are

- What is it desirable for each agent to include in its self description?
- What special-purpose agents would it be convenient to have in the community of agents?

Agents' self-descriptions

In the early stages of our research, we will assume the agents all speak the same language. That is, there will be a common ontology. However, it is neither necessary nor desirable for every agent to know everything. Rather, each agent's "theory of the world" would be a highly particular subset of the ontology, that allowed it to describe, reason about, and communicate with other agents about its own capabilities and needs, but no more. The common ontology would make minimal commitments about the nature of things, and each agent's specific knowledge about its own tasks and operation would be an extension of the common ontology and would be consistent with it. In the longer term, it will be necessary to accommodate differing ontologies since in the real world, agents will "grow up" in different communities. There has been much research lately on ontology matching, and late in the longer-term research we can incorporate the results of this research. But ontology matching is a hard problem, and adopting a common ontology in the early stages will allow us to focus more sharply on agents interacting to achieve complex goals.

The agents in any community of agents will be very diverse. However, there is one feature they will all have in common -- they are artifacts, constructed by people to perform functions people care about and speak about in natural language. Therefore, the language in which diverse agents communicate with each other will be a formal language, but one whose concepts are inspired by the concepts underlying natural language.

There would have to be a largely shared ontology of interaction, so that agents could match their goals with the capabilities of other agents, provide them with the required input, and receive the desired output. This ontology of interaction need not be complex -- in the simplest cases, perhaps involving no more than requests and responses. More advanced versions of the ontology could include models of negotiation.

An agent should know about its own function in the larger scheme of things. For example, it may be useful for a fax machine to know that its function is to send messages, so that other means of sending a message (email, voicemail) can be found when it cannot carry out its task. It should also have an abstract description of the top level or two of its own operation so that it can report on causes of failure. For example, a fax machine could report that the phone number it was given was not a fax machine or was busy, or that the connection was lost in the middle of the transmission. It would not be necessary for the agent to know the mechanical and electronic details of its own operation; in this respect the work we are proposing falls short of the ambitious goals of qualitative physics.

Smarter agents would have planning capabilities and be able to monitor the execution of their plans so they could replan in case some step fails. This would enable working around agents that had become inoperable. One form of replanning, of course, is simply to give up on the highest-level goal and report failure to the agent that invoked it, and ultimately to the human user.

Planning in an assemblage of agents would be distributed. Agent X may know only what Agent Y is capable of achieving, without knowing how Y achieves it. For X the task looks like an executable action -- just have Y do it. For Y the task is something that has to be planned.

Special-purpose agents

One can imagine a community of agents (Figure 1) that works strictly locally; agents find and invoke other agents that can achieve their subgoals; the intelligence would be "everywhere". But the existence of certain special-purpose agents can often greatly facilitate the accomplishment of the user's goals. Among these agents are

- A matchmaker: This agent would know about the capabilities of other agents and when provided with a need could find agents able to satisfy the need.
- A communication intermediary: Rather than building complex interaction capabilities into all the agents, they could simply communicate their need to a communication intermediary, and it would perform the complex negotiations with other agents.
- Natural language agents: These agents would translate between natural language and the language agents use to communicate with each other. They would include agents for text interpretation and generation and speech recognition and synthesis.
- A high-level planner: Individual agents could be equipped with basic planning capabilities, but where these fail, they could invoke the aid of a high-level planner to develop more complicated plans. Alternatively, the high-level planner could be used for top-down control of the entire process of satisfying the user's goal, in some cases.

- A personal assistant: This agent would know about the particular user's individual preferences and requirements, and could monitor the ongoing situation to make sure they are met.

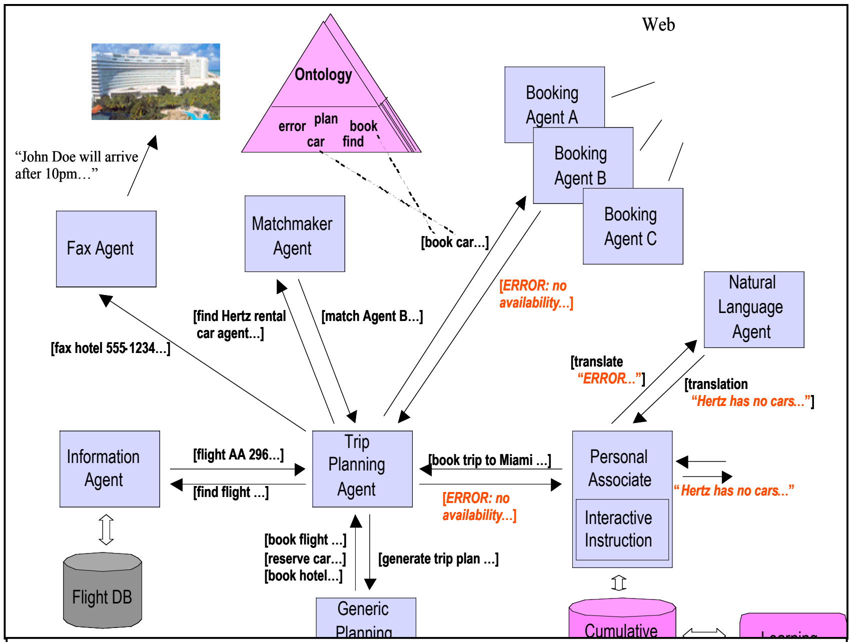

Figure 1: A Community of Agents for Travel Planning

Related Work

In Electric Elves, the agents coordinated their actions using Teamcore, a domain-independent, decentralized, teamwork-based integration architecture (Tambe et al., 1999). Teamcore uses a general-purpose teamwork model (Tambe, 1996) and provides core teamwork capabilities to agents by wrapping them with Teamcore proxies. By interfacing with Teamcore proxies, existing agents can rapidly assemble themselves into a team to solve a given problem. The Teamcore proxies form a distributed team-readiness layer that provides the following social capabilities: (i) coherent commitment and termination of joint goals, (ii) a limited capability for team reorganization in response to member failure, (iii) selective communication, (iv) incorporation of heterogeneous agents, and (v) automatic generation of tasking and monitoring requests. Although Teamcore does not achieve the long-term goals of our proposed research, it provides a solid basis on which to build.

There has been other research relevant to our proposed cognitive architecture. Jennings's seminal work on the GRATE* integration architecture (Jennings, 1995) is similar to Teamcore, in that distributed proxies, each containing a cooperation module integrate heterogeneous agents. One major difference is that GRATE* proxies do not adapt to individual agents, a critical capability if architectures are to integrate an increasingly heterogeneous, complex agent set.

The Open Agent Architecture (OAA) (Martin et al., 1999) is an important well-known agent integration architecture. OAA and similar architectures provide centralized facilitators to enable agents to locate each other, and a blackboard architecture to communicate with each other, but do not provide teamwork capabilities, or adaptation. Teamcore, as well as our proposed cognitive architecture, support a distributed approach that avoids a centralized processing bottleneck, and a central point of failure.

Two other related systems are the RETSINA (Sycara et al., 1996) and IMPACT (Subrahmanian, 1997) multi-agent frameworks. While the goals of these frameworks are somewhat similar to ours, their development appears complementary. For instance, RETSINA is based on three types of agents: (i) interface agents; (ii) task agents; and (iii) information agents. Middle agents allow these various agents to locate each other, but there is no explicit representation of team activities or goals among these agents as they interact. Thus, the adaptive, infrastructural teamwork in our proposed architecture may enable the different RETSINA agents to work flexibly in teams.

Proposed Seedling Project

Specifically, in the period of performance of the seedling contract we would do the following:

1. Specify the requirements for an infrastructure supporting a community of agents and pick one for our initial implementation.
2. Decide upon a formal language for agents to use in representing their knowledge and for communicating with other agents.
3. Develop five use cases of assemblages of agents performing some family of tasks. Two of the use cases would be the ones we have already implemented in Electric Elves -- meeting reminding and travel monitoring. The aim here would be to determine what agents would have to know and be capable of if such applications were to be assembled out of existing agents, or if they were to be able to self-assemble.
4. From the use cases, determine the common ontology of concepts the agents would have to know in order to communicate with other agents to accomplish these tasks.
5. From the use cases, determine the planning and self-monitoring capabilities the agents would need to accomplish the tasks.

ISI is submitting two other white papers that are relevant to this effort. The first concerns how agents would learn in this architecture, thus fleshing out the one major piece this seedling project will not focus on. The second concerns agents capable of self-repair and of devising work-arounds in case of component failures; this is one of the first concrete applications we would want to work on in a longer-term project and the first logical step in developing the architecture. Both of these projects would be strengthened if this project is funded, and this project would be greatly enriched if they are funded.