Electric Elves: Immersing an agent organization in a human organization

**David V. Pynadath, Milind Tambe, Yigal Arens, Hans Chalupsky, Yolanda Gil, Craig Knoblock,
Haeyoung Lee, Kristina Lerman, Jean Oh, Surya Ramachandran, Paul S. Rosenbloom, Thomas Russ**
Information Sciences Institute and Computer Science Department
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
agents-united@isi.edu

Introduction

Future large-scale human organizations will be highly agentized, with software agents supporting the traditional tasks of information gathering, planning, and execution monitoring, as well as having increased control of resources and devices (communication and otherwise). As these heterogeneous software agents take on more of these activities, they will face the additional tasks of interfacing with people and sometimes acting as their proxies. Dynamic teaming of such heterogeneous agents will enable organizations to act coherently, to robustly attain their mission goals, to react swiftly to crises, and to dynamically adapt to events. Advances in this agentization could potentially assist all organizations, including the military, civilian disaster response organizations, corporations, and universities and research institutions.

Within an organization, we envision that agent-based technology will facilitate (and sometimes supervise) all collaborative activities. For a research institution, agentization may facilitate such activities as meeting organization, paper composition, software development, and deployment of people and equipment for out-of-town demonstrations. For a military organization, agentization may enable the teaming of military units and equipment for rapid deployment, the monitoring of the progress of such deployments, and the rapid response to any crises that may arise. To accomplish such goals, we envision the presence of agent proxies for each person within an organization. Thus, for instance, if an organizational crisis requires an urgent deployment of a team of people and equipment, then agent proxies could dynamically volunteer for team membership on behalf of the people or resources they represent, while also ensuring that the selected team collectively possesses sufficient resources and capabilities. The proxies must also manage efficient transportation of such resources, the monitoring of the progress of individual participants and of the mission as a whole, and the execution of corrective actions when goals appear to be endangered.

The complexity inherent in human organizations complicates all of these tasks and provides a challenging research testbed for agent technology. First, there is the key research question of *adjustable autonomy*. In particular, agents acting as proxies for people must automatically adjust their own autonomy, e.g., avoiding critical errors, possibly by letting people make important decisions while autonomously making the more routine decisions. Second, human organizations operate continually over time, and the agents must operate continually as well. In fact, the agent systems must be up and running 24 hours a day 7 days a week (24/7). Third, people, as well as their associated tasks are very heterogeneous, having a wide and rich variety of capabilities, interests, preferences, etc. To enable teaming among such people for crisis response or other organizational tasks, agents acting as proxies must represent and reason with such capabilities and interests. We thus require powerful matchmaking capabilities to match two people with similar interests. Fourth, human organizations are often large, so providing proxies often means a big scale-up in the number of agents, as compared against typical multiagent systems in current operation.

Our *Electric Elves* project is currently investigating the above research issues and the impact of agentization on human organizations in general, using our own Intelligent Systems Division of USC/ISI as a testbed. Within our research institution, we intend that our Electric Elves agent proxies automatically manage tasks such as:

- Select teams of researchers for giving a demonstration out of town, plan all of their travel arrangements and ship relevant equipment; also, resolve problems that come up during such a demonstration (e.g., a selected researcher becomes ill at the last minute)

- Determine the researchers interested in meeting with a visitor to our institute, and schedule meetings with the visitor

- Reschedule meetings if one or more users are absent or unable to arrive on time at a meeting

- Monitor the location of users and keep others informed (within privacy limits) about their whereabouts

This short paper presents an overview of our project, as space limitations preclude a detailed discussion of the research issues and operation of the current system. We do have a *working prototype* of about 10 agent proxies running almost continuously, managing the schedules of one research group. In the following section, we first present an overview of the agent organization, which immerses several heterogeneous agents and sets of agents within the existing human organization of our division. Following that, we describe the current state of the system, and then conclude.
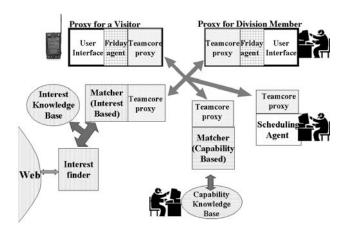
Figure 1: Agent organization for Electric Elves.

Overview of Electric Elves Agent Organization

As seen in Figure 1, we have constructed the agent organization of Electric Elves using Teamcore, a domain-independent, decentralized, teamwork-based integration architecture (Pynadath *et al.* 1999; Tambe, Pynadath, & Chauvat 2000). Each and every agent in the organization has an associated Teamcore proxy that records its membership in various teams and keeps track of active commitments made to these teams. The Teamcore proxies communicate with their corresponding agents to monitor the agents' ability to fulfill commitments and to inform the agents of changes to those commitments. Using a previously developed teamwork model (STEAM (Tambe 1997)), the Teamcore proxies also communicate amongst themselves to ensure coherent execution of team plans and robust achievement of joint goals. The proxies form joint commitments to team plans such as holding meetings, hosting and meeting with visitors, arranging demonstrations, etc. Given their teamwork knowledge, the proxies automatically substitute for missing roles (e.g., if a key person were absent for a demonstration) and inform each other of critical factors affecting a team plan.

Although we have previously used Teamcore to successfully construct organizations of software agents, we need additional agent functionality to support the incorporation of human users. One such addition is the Friday agent (named after Robinson Crusoe's servant and companion, Friday) that serves the user's interests in dealing with the Teamcore proxy. Since the Teamcore proxy considers only the team-level activities, the Friday agent is essential for ensuring proper consideration of each person's *individual* preferences. Thus, the Friday agent reasons about the person's willingness to accept joint commitments and assigned roles. Figure 1 also shows a Teamcore proxy and Friday agent for representing a visitor's role in the team plans, perhaps taking monitoring and communication actions through a personal digital assistant (PDA).

Friday's decision making on behalf of a person naturally leads to the issue of adjustable autonomy. Friday must auto-

matically adjust its autonomy, so that it acts autonomously under appropriate circumstances. For instance, Friday must not overwhelm the user with requests for confirming its own decisions in noncritical situations, yet it must consult the user when making significant commitments on behalf of the user (e.g., volunteering for a presentation). Rather than hand-tune the right level of autonomy for specific users, we are exploiting a learning system using C4.5 (Quinlan 1993). In *training* mode, the user gives Friday feedback about the actions the person would take and also whether the person wants Friday to take such decisions autonomously. Friday uses C4.5 to learn a decision tree from this feedback. In *actual use*, Friday applies its learned knowledge in circumstances where the user wished for it to act autonomously. In other circumstances, it defers the decision to the user. (We have submitted a separate writeup on the details of our use of C4.5 for adjustable autonomy).

The Teamcore proxies also need to reason about the human agents' capabilities when dynamically assigning roles to people during team formation. The capabilities matcher, wrapped by a Teamcore proxy, fulfills this need. We face important challenges in terms of matching agent capabilities that do not arise when agent communities are small and do not include human agents. Human agents can perform many different tasks in many different ways, depending on the context and circumstances, so the advertisements and invocations of agents are more dynamic and complex in nature. When agent communities are large and heterogeneous, the agents that issue a request cannot be expected to be aware about what other agents are available and how they have to be invoked. In this kind of environment, it is likely that most requests will be imprecise or even ill-formulated, and advertisements of agent capabilities need to be able to (1) support the requesting agent in formulating the request properly, (2) support negotiation and brokering, and (3) adapt dynamically as the agents or their environment change over time.

To address the above concerns, our capabilities matcher draws from previous work on matching problem-solving methods within the EXPECT architecture (Gil & Swartout 1995). In our approach, task descriptions are tightly integrated with ontological descriptions, in our case specified in Loom (MacGregor 1991; Gil & Gonzalez 1996), a knowledge representation system based on description logic. Using ontologies, we represent information about people, projects (their members, funding agencies, software, etc.), equipment, etc. The agent capabilities are translated into Loom descriptions, and are organized in a subsumption hierarchy that exploits other ontological descriptions as well. When a request is posted, the matcher translates the request to a concept description and returns a suite of agents that match the request. We have implemented several kinds of matches, including exact match, subsumption match, reformulation match, reverse subsumption match, and partial match.

In addition to the capability-based team formation, the Teamcore proxies must also dynamically form new teams based on shared interests or areas of expertise. For instance, if a researcher is visiting the division, a commitment forms to host the visitor and arrange meetings of interest. To this

end, the Teamcore proxies must reason about the research interests of the various users. The interest matcher accomplishes this reasoning. The interest matching problem has a variety of interesting characteristics that influenced our solution approach. (1) Relevant information ranges from highly structured to highly unstructured. Simple factual information such as attended universities, home country, etc., is relatively easily obtainable and formalizable, while a person's interests are often only implicitly documented, for example, by their publications. (2) Manual formalization of relevant information is only feasible if the information is very static. Manual maintenance requirements must also be kept to a minimum to ensure agent autonomy and quality of decisions. (3) Decisions by the matchmaker agent need to be explainable to the affected people depending on the chosen level of autonomy. People should also be able to influence or customize the matchmaking behavior, for example, to override automatically derived information that is incorrect.

To handle the problem characteristics described above, our interest matcher combines statistical information-retrieval (IR) techniques with knowledge-based matchmaking based on logical inference. The knowledge-based approach explicitly formalizes the matchmaking decision process which facilitates explanation. It also allows us to easily handle structured, factual information, to capture information not available from on-line sources, and it eases customization by users. The IR techniques, on the other hand, are good at dealing with unstructured information. They can deliver results in case the knowledge-based techniques fail (e.g., because the ontology is incomplete), they do not require manual modeling, and they require little or no maintenance, since they can directly consult on-line sources. The seamless combination of IR with knowledge-based reasoning is a non-trivial problem, however, and requires the development of new solutions.

Electric Elves: Current Status

We have agentized about 10 members of our division, i.e., provided them with corresponding Teamcore proxies for managing meetings as team-level activities and Friday agent proxies for responding to individual preferences and communicating with the user through a workstation or PDA. We are thus applying the results of our research to our own daily lives within our division. We have also integrated the capability matcher within this agent organization. The Teamcore proxies consult the capability matcher any time they must reason about their users' capabilities. We have constructed the interest matcher and are in the process of integrating it into the operating agent organization.

In the early stages of our agentized effort, we ran the agent proxies to coordinate real meeting schedules for a sub-group of five agentized people, working on five workstations and also equipped with two PDAs and one Global Positioning Service (GPS) device (providing accurate information about a user's location), as in Figure 2, over the course of two weeks. Teamcore proxies managed team-level commitments (i.e., meetings), and Friday agent proxies monitored individual users' locations and made decisions about what

Figure 2: PDA (Palm VII) with GPS device

rescheduling actions to take, when necessary. We then extended the agentization to include ten members of our research group.

Having used the agent proxies for a few weeks, we have made several observations on the effectiveness of the architecture. For instance, when users had cause to delay the meeting, they found it much easier to have Friday request delays (e.g., through a dialog box), because the automatic notification of the other attendees saves effort over hand-generating email notifications. Friday's autonomy, in conjunction with its location reasoning, has successfully delayed meetings in useful situations. For instance, the proxies have correctly autonomously delayed meetings when users have been delayed while driving to ISI, when they are unable to provide any feedback to Friday.

Friday's ability to take autonomous actions has also caused users to adopt a different interaction style than they do with more passive or advisory groupware. Traditional scheduling tools can provide reminders when a meeting time is approaching, but users (at least within our research group) find it all too easy to casually dismiss these reminders and still arrive late for such meetings. However, when Friday pops up a meeting dialog box on a user's terminal, users usually had Friday report that they were attending, rather than explicitly delay the meeting itself. In addition, since the user's Teamcore proxy relays that information to the other attendees proxies', the users felt compelled to arrive on time for the meetings, rather than appear to have provided misleading information.

Adjustable autonomy has turned out to be a surprisingly important issue in practice, since small errors on agents' part may critically affect users. C4.5 has provided a useful initial tool in addressing adjustable autonomy, but recent experiments have revealed some of its limitations. For instance, in one or two instances agents have autonomously cancelled meetings in situations when the user would have actually preferred some other action. One potential explanation for these errors is that C4.5 does not weigh costs of erroneous classifications (e.g., mistakenly choosing the cancellation action). Such issues are at the top of our agenda for future work.

In addition to their ability to take autonomous actions, the agent proxies' monitoring of users' commitments supports increased social awareness by providing users with additional knowledge about the status (e.g., location) of their teammates. We have used Friday's location reasoning to

Friday User Information

David Pynadath

Current Location: 955 (attending a meeting)   Office: 923 (Tel: 88467)
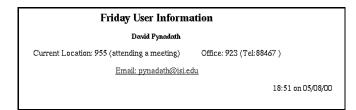
Email: pynadath@isi.edu

18:51 on 05/08/00

Figure 3: Sample Web page provided by People Locator.

construct a People Locator that publishes the whereabouts of members of our research group on a Web page, as in Figure 3. This automatically updated information provides a cheap means for increasing social awareness (similar to previous work in the field (Tollmar, Sandor, & Schömer 1996)). The Web page also provides each user's telephone number and email hyperlink. This allows someone visiting the People Locator to quickly contact the user through the medium most appropriate to the user's current location. For instance, if the People Locator provides the page shown in Figure 3, then Milind knows to use the email hyperlink to most reliably send a message to David, who is not in his office, but who could still receive it on his PDA. If David was shown to be in 923 itself, then Milind may choose to call him. We are aware of the privacy issues that the current implementation raises, but we are temporarily focusing on other issues as we conduct our experiment within our small-scale research group, where only members of our research group know the location of the People Locator's Web page.

Related Work

The Electric Elves project brings together a number of heterogeneous distributed agents and matchmakers, working within a distributed multi-agent integration architecture, Teamcore. There are several areas of related work that we can discuss very briefly.

With respect to multi-agent architectures, Jennings's seminal work on GRATE* (Jennings 1995) integration architecture is similar to Teamcore, in that distributed proxies, each containing a cooperation module, integrate heterogeneous agents. However, GRATE* proxies did not interface with people, and thus did not have to adapt to people nor address the matchmaking issues discussed in this paper. While the Retsina architecture (Sycara *et al.* 1996) also shares some similarities, we believe our goals are complementary to a large extent. In particular, Retsina is based on three types of agents: (1) interface agents; (2) task agents; and (3) information agents. Here, Teamcore's infrastructural teamwork could potentially facilitate such agents to work together in teams.

The RETSINA multi-agent framework also includes a matchmaking service that uses a language called LARKS (Sycara *et al.* 1999). A signature-style description of capabilities is used, which is more sophisticated than our approach in terms of describing input/output constraints but is not as explicit in representing the task action itself. It is similar to our approach in that it also relies on semantic matches of descriptions, and some of the matching process is performed using additional information retrieval techniques.

Interest-based matchmaking has been studied in a variety of systems. The Visitor-Hoster application (Sycara & Zeng 1996) uses an explicit, manual specification of a visitor's interest which is matched with explicitly specified interests in a personnel database. The Yenta system (Foner 1997) approaches the problem by using statistical text analysis methods similar to ours. Interests are inferred from the newsgroup articles people read, the email messages they receive, etc., and interest overlap is measured by analyzing the similarity of relevant documents. None of these systems combine IR with knowledge-based methods to handle structured and unstructured information in a uniform framework, to allow an explicit formalization of the matchmaking process, and to provide explanations in support of adjustable autonomy. Relating documents to concepts in an ontology — document classification — is an active field of research in IR.

Conclusion

This paper presents our vision for a complex, heterogeneous agent organization that automates numerous tasks central to most human organizations. As part of our initial steps towards this goal, we have implemented a set of agents that act as proxies for members of our research group in managing their schedules. We have implemented the interaction between the capability matcher and its Teamcore proxy, and we are also in the process of integrating the interest matcher into the organization. We have run parts of the agent organization of Figure 1 as a 24/7 system within a single research group for a few weeks. We will have the entire organization running within our entire research group in the near future, thus providing a rich testbed for our solutions to the problems addressed in this paper, as well as for our ongoing attempts to address the more general issues of agentizing large-scale human organizations.

Acknowledgments

References

Foner, L. 1997. Yenta: A multi-agent, referral-based matchmaking system. In Johnson, W., and Hayes-Roth, B., eds., *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, 301–307. New York: ACM Press.

Gil, Y., and Gonzalez, P. 1996. Subsumption-based matching: Bringing semantics to goals. In *International Workshop on Description Logics*.

Gil, Y., and Swartout, B. 1995. Expect: Explicit representations for flexible acquisition. In *Proc. Ninth Knowledge Acquisition for Knowledge-Based Systems Workshop*.

Jennings, N. 1995. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence* 75.

MacGregor, R. 1991. Inside the LOOM description classifier. *ACM SIGART Bulletin* 2(3):88–92.

Pynadath, D. V.; Tambe, M.; Chauvat, N.; and Cavedon, L. 1999. Toward team-oriented programming. In Jennings, N. R., and Lespérance, Y., eds., *Intelligent Agents VI: Agent Theories, Architectures and Languages*. Springer-Verlag. 233–247.

Quinlan, J. R. 1993. *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Sycara, K., and Zeng, D. 1996. Coordination of multiple intelligent software agents. *International Journal of Cooperative Information Systems* 5(2&3).

Sycara, K.; Decker, K.; Pannu, A.; Williamson, M.; and Zeng, D. 1996. Distributed intelligent agents. *IEEE Expert* 11:36–46.

Sycara, K.; Lu, J.; Klush, M.; and Widoff, S. 1999. Matchmaking among heterogeneous agents in the internet. In *AAAI Spring Symposium on Intelligent Agents in Cyberspace*.

Tambe, M.; Pynadath, D. V.; and Chauvat, N. 2000. Building dynamic agent organizations in cyberspace. *IEEE Internet Computing* 4(2).

Tambe, M. 1997. Towards flexible teamwork. *Journal of Artif. Intell. Research* 7:83–124.

Tollmar, K.; Sandor, O.; and Schōmer, A. 1996. Supporting social awareness @Work design and experience. In *Proceedings of the ACM Conference on COmputer Supported Cooperative Work*, 298–307.