COMPUTATIONAL AGENTS TO SUPPORT COOPERATIVE WORK

Kevin Crowston and Thomas W. Malone

Masssachusetts Institute of Technology

Sloan School of Management

WP 2008-88

COMPUTATIONAL AGENTS TO SUPPORT COOPERATIVE WORK

In the past, most uses of computers focused on automation, that is, the replacement of human workers by a computer system. Examples of these systems are common. For example, before computers, most companies' accounts were maintained by teams of clerks. Today, most companies use computers. For routine work, this automation is reasonable, since computers are much better at certain repetitive operations than people are. Other kinds of work are less routine, however, and thus less suitable for straightforward automation.

More recently, therefore, attention has shifted to the use of computers as support tools, that is, using computers to make individuals more productive. For example, word processing systems make typists more productive by eliminating the need to retype revised documents. Analysts using spreadsheets can more quickly consider many alternatives. By automating certain routine tasks, computer systems can allow humans to concentrate on more demanding parts of their jobs.

However, most existing support systems are intended for an individual working alone. They offer little support for people working in groups, even though most jobs require some degree of cooperation. Furthermore, current design methodologies offer few specific suggestions for the design of systems to support such cooperative work.

In this paper, we will develop a new way of thinking about computers as *agents*. These computational agents act autonomously on behalf of their human *principals* by performing some, but usually not all of the tasks involved in cooperative work. Sometimes these agents help a principal process messages to and from *third parties;* in other cases the agents themselves carry out conversations on behalf of their principals. This view of computers as agents that can sometimes take actions and make commitments on behalf of people contrasts with the more restricted view proposed by Winograd and Flores (1986) that computers should only be used as "tools for conversations". We believe, however, that computers can be useful in this more expansive way, when used appropriately.

Viewing computers as agents is, of course, not a new idea. Alan Kay (1984) and others (e.g., Nilsson, Cohen, & Rosenschein, 1987) have been using the term in this way for years, and many agents are already in common use. For example, a computer system that sends bills and handles payments is a kind of agent, as are simpler machines, such as telephone answering machines, automatic teller machines (ATMs), and vending machines. We believe, however, that parallels between computerized and human agents may make computerized agents more understandable for both users and designers of systems. In this paper, we will focus on designers and show how this perspective can: (a) help generate ideas for new computer applications, and (b) help suggest how to use computer agents in socially desirable ways. We suspect that many other kinds of computer agents will eventually become as commonplace and accepted as answering machines and ATMs are today.

Computational agents

We define roles in a cooperative process in terms of the messages individuals send and receive and the ways they process those messages. For each such role, we can then imagine an agent to support it by automatically processing all or part of a set of messages. Although computer systems provide a great deal of flexibility and may therefore often make good agents, simpler machines may also be useful. For instance, a telephone answering machine is an example of an agent, since it autonomously carries on and remembers part of a conversation on behalf of its owner.

We will mainly discuss agents that support cooperative work by processing a flow of messages to or from a third party (that is, someone other than the agent and the principal). In addition to a supplier and consumer of messages, an agent has three components that control its message processing. First, the agent has *goals* that it gets from the principal that tell it what it should try to accomplish. Second, an agent has a set of *actions* it can take to achieve its goals. Finally, an agent has a set of *triggers* that start the actions. These agents are *semiautonomous*: autonomous, because once they have been created, they continue to process messages without the explict attention of their human user, semiautonomous because their knowledge and processing rules can always be inspected and modified by the principal and because they may often refer messages to the principal rather than handling the messages themselves.

It should be noted that agents could also be used to support an individual working alone on some task. For example, an agent could be used to scan the environment and inform its principal when some condition holds, such as a server crashing in a distributed environment. Unlike most conventional support systems, however, agents can also support people working in groups by supporting their conversations with third parties and in this paper, we will focus on this type of agent.

Theoretical bases

The two primary theoretical bases for our view of agents are economic agency theory and the information processing view of organizations.

Agency theory

One basis for our concepts of agents and principals is agency theory in economics (e.g., Pratt & Zeckhauser, 1985; Ross, 1973). Agency theory views an organization as a web of principal-agent relationships, where the principal depends on the agent to carry out some actions. For example, the managers of a firm can be considered principals and their subordinates, their agents. In a human organization, these relationships have a cost because the principal does not know everything the agent knows and can not be sure the agent will act in the principal's best interests. These costs can include monitoring agents' actions to ensure compliance, or arranging incentives to motivate agents to act to their principals' benefit.

Using computers as agents may reduce some of these costs because computer systems have no inherent interests of their own, and can therefore be made to share their principals values more precisely. Doing so also has a cost, however, so it may not always be worthwhile to make the match perfect.

Information-processing viewpoint

The information-processing viewpoint, as described by Crowston, Malone and Lin (1988), suggests how computers can support cooperative work. This viewpoint treats an organization as a group of cooperating and intercommunicating actors, where an actor may be an individual, a group, or a computer system, depending on the organization and the level of analysis. Important features of the organization are the patterns of communications between actors and the information processing done by the actors. These communications can be modeled in terms of the content and purpose of the messages exchanged and the actions these messages trigger in their recipients. (We use the term "message" here in an abstract sense that includes any communication, oral as well as paper or electronic.) These models are therefore similar to a computer program written in an object-oriented language (e.g., Stefik & Bobrow, 1986), because they specify the different classes of actors, the messages they understand, and the processing they do for each message. A series of messages forms a conversation that implements some cooperative process. A related analysis of conversations is proposed by Winograd (1988).

Examples of agents

Viewing computer systems as agents helps generate ideas for new uses of computers. First, we can look at what humans already use agents to do, and consider which things could be automated. Second, we can consider conditions that make the use of human agents costly and consider places where computer systems would decrease those costs.

Automated agents are probably most appropriate for handling routine interactions, where we can predict the kinds of information likely to be received and state rules for processing them. This restriction would seem to rule out the use of agents in cases where the information is nonroutine and difficult to characterize. We believe, however, that coordination processes are likely to be more generic than production processes, so agents that support common coordination processes may be useful for many tasks. We will discuss examples of different kinds of agents, including agents for sorting, scanning, or redistributing messages, for monitoring the environment, maintaining a database or for negotiating with other individuals or agents. (These examples are summarized in Table 1.) Our examples include some agents that have already been implemented (by us or by others), some we are currently developing, and several that are at this time only hypothetical.

Insert Table 1 about here.

Sorting agents

Information Lens mail sorting agent

The Information Lens system (Malone, Grant, Turbak, Brobst, & Cohen, 1987; Malone, Grant, Lai, Rao, & Rosenblitt, 1987) includes a message sorting agent that can automatically process electronic mail messages received by a user. Users (or "principals") indicate how they want their messages handled by writing rules. Each rules specifies the actions that should be taken when a particular type of message is received. One key feature of the Lens system is the use of message templates that indicate the general type of the message (e.g., a request that the principal do something or an announcement of a meeting) and that isolate specific kinds of information (e.g., the deadline for the requested action or the time of the meeting) in specific fields so the system can process some of the information in the messages without understanding the rest. Possible actions include deleting unwanted messages, moving messages to folders (e.g., an urgent folder for requests with a deadline of today), or forwarding the message to someone else to handle. In developing the Lens system, for example, we used the system to automate part of the bug handling process by allowing users to write rules that automatically loaded bug fixes when they were announced.

Scanning agents

Another use for agents is to look for information that the principal would like to have, but would otherwise not see. These searches could be done either on information in a database or on a stream of messages. The first kind of agent is like running a query on a database; the second is more like placing a standing order for certain kinds of information.

Information Lens "anyone" agents

The "anyone" agents in the Information Lens system help principals pick interesting messages out of a stream of messages they otherwise would not read. The principals specify the kinds of messages they want by writing rules, similar to those used in the Information Lens mail sorting agent. Then, when people send mail, they can add an additional recipient, "anyone", indicating that they are willing to have the message read by anyone else who is interested. General interest mailing lists may also have "anyone" as a recipient. All messages addressed to "anyone" are processed by a special server (the "anyone" server) where the users' "anyone" agents reside. When a user's "anyone" agent receives these messages, it filters them using the given rules, and forwards to the user all messages that are selected by the user's rules. For example, our research group receives a daily online feed of New York Times articles and individuals can use their "anyone" agents to select specific articles (e.g., movie reviews or articles that mention computers).

Buyer and seller agents

Another use for information scanning agents is to help buyers find products and sellers find buyers. A buyer's agent could search a database of products, such as airline flights, looking for one that best fits its principal's criteria and bring these to the principal's attention. A few early examples of such buyers' agents already exist and many more seem likely (Malone, Yates & Benjamin, 1987). A seller's agent could search for prospective customers, using credit card purchase histories or demographic data, and send likely candidates information about the product.

Redistribution agents

Redistribution agent

Another kind of agent that we are currently implementing allows the principal to write rules that search for different kinds of messages and send them to some other third party. These agents will reside on a central server like the "anyone" server and so will operate continuously, whether the principals receive their mail or not. One possible use for this kind of agent is to redistribute information to other people in a group who are likely to be interested. For example, we have an on-line distribution list for all the members of our research group. Much of the message traffic on this list involves detailed descriptions of bugs and their fixes, but some other information is included as well. The people in the group who are actively involved in developing software want to see all these messages, but most of the others do not. A redistribution agent will be able to automatically filter out the bug messages and send the remaining messages on to the non-programming members of the group. In this way, both groups of users will see the messages in which they are interested, without people in either group having to set up individual filtering rules and without the senders of messages having to worry about who in the group should see a particular message.

Someone agent

A second kind of redistribution agent can be used to automate the routing of certain kinds of information. For example, a travel expense report needs to be sent to various people for their approval and processing. A "someone" agent could automate this routing by forwarding the message to the appropriate next person. (Unlike the "anyone" agent which discards messages in which no one is interested, the "someone" agent guarantees that the message will go to someone). The principal could control the agent by specifying rules indicating the proper destination for different kinds of messages, taking into account features such as the total cost of the trip, the sender's manager or which people have already seen the message, and what they did with it.

Database agents

Some agents may handle messages by refering to a database, either to answer a question or to store the information received.

Question answering agents

Question answering agents could be used to automatically answer requests for information from third parties. These agents may implement complex access restrictions, or give different questioners different views of the data. One example of this sort of database agent is the *netlib* system (Dongarra & Grosse, 1987), which automates the distribution of small routines from a software library. The system receives electronic mail messages which may request a specific routine, a list of available routines in a particular library, or a search for all routines described by a particular keyword. The results of the search are returned in another mail message.

Calendar agent

An agent acting as the front end to a database could be used as a calendar agent to keep track of its principal's schedule. It could receive and automatically store announcements of events, or allow third parties to make appointments with its principal, perhaps making appointments automatically for specific times or individuals. Other requests could be forwarded on to the principal and the agent could also send reminders of scheduled events at appropriate times.

Monitoring agents

Project management agent

An agent could help support a project manager by monitoring the status of subtasks and informing the manager when the completion of a subtask was delayed. The agent could receive copies of messages negotiating the due date and announcing the completion of subtasks. If a completion message is not received by the expected date, the agent could automatically send a request to the person who is supposed to be doing the task asking for information about its current status or new expected completion date. In some cases, the response to such a query will be that the subtask has already been completed, in which case the agent might not need to inform the manager at all. In other cases, the agent will forward on to its principal the new information about why the task is delayed and the new expected completion date.

Negotiating agents

Buyer and seller agents

For some products, the buyer and seller scanning agents described above could be supplemented with the ability to actually interact and negotiate a deal to buy or sell the product. Such agents might be especially useful for frequent purchases of commodities, where the product is easy to describe and sellers differentiate mainly on price. Vending machines are a simple examples of sellers' agents, since they offer a product and carry out sales. A program trading system for financial instruments is a kind of buyer's agent. It receives messages from markets, indicating the current prices of various kinds of securities, and processes them based on rules given by the principal. Based on the results, the agent then sends messages to buy or sell the securities.

The "negotiation" in all these cases is relatively trivial, but it is easy to imagine more sophisticated agents that include rules for how to negotiate such matters as the selling price or to trade off one attribute for another (such as price for delivery date). For instance, if a seller's agent offers to sell a product for a price x to an agent for a buyer who is willing to pay a maximum price y, (where $y < x$), a simple bargining rule for the buyer's agent might be to offer $2y - x$, and then continue responding in this way to each new offer from the seller. Obviously, the actual negotiating rules encoded in such agents would be a matter of extreme secrecy, and it might even be desirable to include significant randomness in the process to deter agents who might try to infer another agent's negotiating rules.

Guidelines for using agents

The concept of agents helps analyze how computer systems can be used in socially desirable ways. Agents themselves are ethically neutral, but as a source of change they may cause problems. Some of these problems are similar to concerns raised by other uses of information technology, while others seem unique to agents. The actions taken by computer systems are usually actions that were done in other ways without computers, and so, even though the specific rules that governed those earlier situations may not apply directly, the underlying moral principals are often the same (Johnson, 1988). Since relationships between human principals and agents are governed by familiar and well articulated moral principles, it is often helpful to use moral principles from these relationships in analyzing the use of computerized agents.

We suggest two broad principles that we feel are a starting point for guiding the use of computerized agents. First, agents should be designed to serve the principal, and in many cases, the actual user of the agent. Second, some communication should always be done directly by humans, not their computer agents.

The principal should control the agent.

Even though in principle an agent should do what its principal wants, there may be some cases where an agent can impose its value system on the user, either because it is badly designed in an inflexible way or because it embodies a normative theory of the process to be supported. We believe that, in most cases, such an agent would be undesirable, because it would reduce the autonomy of the principal. Agents should be flexible enough that they do what the principal wants and not vice versa.

The user should usually be the principal.

In some cases, the imposition of the agent's value system on the user may be intentional. For example, it may be that the human user of the agent is not the principal, but rather that both the human and the computer are agents of someone else. In this case, the agent may act as a means of technical control, allowing the principal to control the human user. In some cases, using agents as a form of technical control may be an especially likely outcome, since systems are often designed to serve the managers of a company and not the direct users. Grudin (1987), for example, points out that many multi-user systems require many people to use them while benefiting only a few.

In most cases, however, we believe that computer agents will be most easily accepted and most effective when their primary direct user is also their principal. For example, the Information Lens system allows individual users to decide how messages should be handled by their agents, and relies on other organizational incentives to control how people make these decisions. Nothing in the Information Lens system, for example, prevents people from creating rules to automatically delete all messages from their bosses. We assume, however, that in most cases there are powerful organizational incentives that make this very unlikely.

Some communications should be done by humans, not agents.

Communications have a social content.

Agents may make interactions between humans more efficient by automating some of them. However, not all communications are work related and even many work-related interactions have an important social component. Agents that reduce social interactions between people may eventually damage the social fabric of their organizations. Even though these communications are not efficient, in the sense of directly accomplishing work, they may be very important in maintaining the organization. If using agents forces humans to only interact along formal channels, or decreases their desirable social interactions with other humans, then peoples' ability and desire to do their work may be greatly diminished. This suggests that agents should be quite flexible in the ways they allow people to communicate with each other, not just with other agents.

Using an agent also indicates the principal's rating of the importance of the communication and it may therefore be inappropriate to use agents in some contexts. Some of these circumstances can be seen by analogy to the case of a human secretary. For example, in an academic environment, professors would probably consider a call from the dean's secretary a reasonable way to ask for suggestions for names of tenure reviewers, but an inappropriate way to inform them of the results of their tenure decision. Messages that are very important or that have important interpersonal subtleties should usually be handled by people, not their agents.

Third parties should be able to "speak to the manager."

A possible undesirable reason for using agents is to avoid responsibility, by hiding the principal from people interacting with the system (Winograd & Flores, 1986, p. 155). For example, Waldrop (1987) points out that using an expert system to approve bank loans, "obscures the fact that the machine's 'decision' actually embodies a policy made by humans" (p. 35) and suggests that bank managers may deny responsibility for rejected applications by explaining that the computer made the decision. Agents do embody the value systems of their principals, and those principals should be clearly identified, just as a vending machine identifies whom to call in case of a problem.

Furthermore, third parties who interact with agents should be able to appeal directly to the agent's principal if they are dissatisfied with the agent's performance. For example, a computerized billing system is a type of agent. Its goal is to have the bills paid; to achieve this goal, it sends requests for payment, tracks the payments received and takes further action when the bills remain unpaid, such as sending further requests for payment, or shutting off service and informing other credit agencies. The problem with these systems is not so much that they send dunning letters as that they sometimes ignore the responses if the bill remains unpaid. This suggests that agents be designed to react to all responses, forwarding some to the principal if the system can not deal with them itself. (The principal may choose to have human agents handle the response that the computerized agent can not handle; in this case, however, the human agents should be able to control the computerized agent and resolve the problem.

Conclusion

We believe that one of the most important uses of computers will be to help coordinate the activities of people working in groups, and that agents are a useful way to think about many of these systems. In particular, thinking about computer systems as agents that support individuals in their interactions with others suggests new applications for computers, indicates ways to use computers in socially desirable ways, and may provide a better user model. If used well, agents have the potential to dramatically increase our range of actions.

We expect, however, that it will take time for some agents to be widely accepted. For example, when telephone answering machines first became available, many people would not own or talk to one. Today, it is almost expected that anyone you call will use one, and some companies have begun to depend on their internal voice mail systems. ATMs have similarly changed from a rarity to almost a requirement for a bank to do business. In the future, computerized agents may take their place with telephone answering machines, ATMs and other conveniences of modern life.

REFERENCES

Crowston, K., Malone, T. W., & Lin, F. (1988). Cognitive science and organizational design: A case study of computer conferencing. *Human-Computer Interaction, 3,* 59-85.

Dongarra, J. J., & Grosse, E. (1987). Distribution of mathematical software via electronic mail. *Communications of the ACM, 30,* 403-407.

Flores, C. F., & Ludlow, J. (1981). Doing and speaking in the office. In G. Fick & R. Sprague (Eds.), *DSS: Issues and challenges.* London: Pergamon Press.

Grudin, J. (1987). Social evaluation of the user interface: who does the work and who gets the benefit? *Proceedings of Human-Computer Interaction—Interact '87,* 805-811. Amsterdam: Elsevier Science Publishers B. V. (North Holland).

Johnson, D. (1988). *Computer ethics: Public and private in computer conferencing.* Unpublished manuscript, Rensselaer Polytechnic Institute.

Kay, A. C. (1984, September). Computer software. *Scientific American, 251,* 53-60.

Lai, K. Y., & Malone, T. W. (1988). *Object lens: A spreadsheet for cooperative work.* Manuscript submitted for publication.

Malone, T. W., Grant, K. R., Turbak, F. A., Brobst, S. A., & Cohen, M. D. (1987). Intelligent information-sharing systems. *Communications of the ACM, 30,* 390-402.

Malone, T. W., Grant, K. R., Lai, K. Y., Rao, R., & Rosenblitt, D. (1987). Semi-structured messages are surprisingly useful for computer-supported communications. *ACM Transactions on Office Systems, 5,* 115-131.

Malone, T. W., Yates, J., & Benjamin, R. I. (1987). Electronic markets and electronic hierarchies. *Communications of the ACM, 30,* 484-497.

Nilsson, N. J., Cohen, P. R., & Rosenschein, S. J. (1987). *A proposal for research on intelligent communicating agents.* Unpublished manuscript, Stanford University, Stanford, CA.

Pratt, J. W., & Zeckhauser, R. J. (Eds.). (1985). *Principals and agents: The structure of business.* Boston, MA: Harvard Business School Press.

Nilsson, N. J., Cohen, P. R., & Rosenschein, S. J. (1987). *A proposal for research on intelligent communicating agents*. Unpublished proposal submitted to DARPA, Stanford University.

Ross, S. (1973). The economic theory of agency: The principal's problem. *American Economic Review, 63*, 134-139.

Stefik, M., & Bobrow, D. G. (1986, Spring). Object-oriented programming: Themes and variations. *AI Magazine*, pp. 40-62.

Waldrop, M. M. (1987, Spring). A question of responsibility. *AI Magazine*, pp. 29-39.

Winograd, T. (1988). A language/action perspective on the design of cooperative work. *Human-Computer Interaction, 3*, 3-30.

Winograd, T. & Flores, F. (1986). *Understanding computers and cognition: A new foundation for design.* Norwood, NJ: Ablex Publishing Corporation.

Table 1. **Examples of computerized agents.**

Agent	Goals	Actions	Triggers
Information Lens mail sorting agent	Sorting and prioritizing mail	Deleting messages, moving messages to folders	Arrival of mail for principal
Information Lens "anyone" agent	Finding interesting messages	Sending mail to the principal	Third party sends a message
Buyer agents	Finding products	Informing the principal	Principal asks for a product
	Buying a product	Negotiating a purchase	
Seller agents	Finding a potential buyer	Telling the buyer about a product	Ongoing
	Selling a product	Negotiating a sale	
Redistribution agent	Redistributing messages to third parties	Sending mail to third parties	Third party sends a message
Someone agent	Sending messages to the "right" person	Selecting a recipient and resending messages	Third party sends a message
Question answering agents	Answering third parties' questions	Determining the answer, sending the reply	Third party sends a question
Calendar agent	Maintaining the principal's calendar	Recording announced events, scheduling appointments, reminding principal	Receipt of event announcement or requests for an appointment
Project management agent	Tracking progress of subtasks	Remember expected completion dates, and completions, send reminders for overdue projects	Arrival of deadlines; receipt of commitments or completion announcements

ACKNOWLEDGEMENTS

The work described in this paper was supported in part by the Management in the 1990s Research Program at the Sloan School of Managment, and by Wang Laboratories, Xerox Corporation and General Motors/Electronic Data Systems.

This paper has benefited from discussions with Wendy Mackay, Jintae Lee, Michael Munoz, David Rosenblitt, George Roth and Jane Salk.