

Delegating to Software Agents

Allen E. Milewski

AT&T Laboratories , 600 Mountain Avenue, Room 2A304, Murray Hill, NJ 07974-0636
USA, aem@research.att.com

Steven H. Lewis

AT&T Laboratories , 101 Crawfords Corner Rd, Room 1K312, Holmdel, NJ 07733-3030
USA, shlewis@att.com

Summary

There is currently a great deal of interest in the development of intelligent agents. While there is little agreement on exactly what constitutes an intelligent agent, many definitions embody a user interface model that differs from the traditional one where users perform tasks with the help of computer-based "tools." In contrast, the "delegation" model associated with agents is based on entrusting tasks to an autonomous, sometimes anthropomorphized system, whose performance is monitored and evaluated. This change in user interface model is a dramatic one since delegation can be a difficult and often-avoided behavior in humans. Agent interface designs need to overcome well-established drawbacks in delegation. For this purpose, designers should find the management sciences and organizational psychology literatures to be as relevant as that of traditional human factors. This paper describes issues regarding task delegation as they pertain to the design of intelligent agent user interfaces.

1. Introduction

There is currently a great deal of interest in the development of intelligent agent technology (e.g., Riecken, 1994; Maes, 1994). While the origins of this technology goes back many years to include expert systems, robotics and supervisory control systems (e.g. Negroponte, 1970, Minsky, Roth, Bennett and Woods, 1987; Woods, Johannesen and Potter, 1991), its popularity has grown rapidly in the past few years. One significant event for the popularity of agents was a 1987 promotional video from Apple Computer called the Knowledge Navigator. The fictitious Knowledge Navigator had exaggerated capabilities compared with current agents, but it is often held as a goal. In this video, a professor was helped in his work by an artificial intelligence on his futuristic desktop computer. This intelligent agent interacted with the professor in a human manner, conversing with the professor, expediting the professor's activities, setting up meetings, making telephone calls, searching databases, making suggestions for improvements, and performing clerical tasks.

Designing an intelligent agent user interface like the Knowledge Navigator requires consideration of very different issues than those associated with the design of, for example, a contemporary graphical user interface (GUI). The user interface model of most GUIs treats programs as tools. These tools lie readily available, but inactive, until the user chooses to wield them. The work itself is largely done, or at least impelled, by the user's own strategies, goals and initiative; the tools aid portions of the work (Chin, 1991). The user interface model associated with intelligent agent systems differs from the "tool" model in several ways. It can involve collaboration on decisions (Chu and Rouse, 1979) and the acceptance of advice (Carroll, and McKendree, 1987). But, the most common characteristic of this model is that it is based on authority and delegation of work

(Shoham, 1993, Maes, 1994). Delegation has received a significant amount of study within social and organizational psychology and in management science (e.g., Moore, 1982, Leana, 1986, 1987). There has not been much study of delegation in the context of computer science and user interface design. Development of design guidelines for "good" intelligent agent user interfaces will require a significant accumulation of experience with real systems; this accumulation is now only beginning. The purpose of this paper is to review what an intelligent agent user interface is and to describe some major design issues posed by its delegation-oriented interactions.

2. Traits of an Intelligent Agent

Currently, the term "agent" is used to describe such an extremely wide range of systems that some are concerned that the term may be nearly meaningless (Shoham, 1993). These systems range from simple shell scripts (e.g., Hewlett-Packard's New Wave) to sophisticated, artificial intelligence systems that use inference and planning (Etzioni, and Weld, 1993). Part of the confusion results from the fact that each writer associates with agents a somewhat different set of traits (Roesler and Hawkins, 1994). Nearly all definitions contain some combination of the following traits.

2.1 Ability to work asynchronously and autonomously

The trait most generally assumed for intelligent agents is that they can function autonomously, and without intervention from humans (Shoham, 1993; Maes, 1990). It is this trait, for example, that permits simple shell-scripting facilities to be considered an agent.

2.2 Ability to change behavior according to accumulated knowledge

The ability to "learn" may be the second most generally assumed trait associated with intelligent agents. For example, "adaptive aiding" has been featured as a central feature in intelligent interface architectures (Morris, Rouse and Ward, 1988; Rouse, Geddes, and Curry, 1988). Similarly, an often proposed use of agents is to anticipate the user's information needs in searching databases by maintaining an "interest profile" based on past search preferences. Since any user's interest changes across time, so must the agent's understanding of it.

2.3 Ability to take initiative

Chin (1991) has argued that a truly intelligent agent must have the ability to perform tasks based on its own goal structure, separate from that of the user. For Chin, this constitutes an agent acting on its own initiative. The environments that led to this proposal were those in which the agent knows a great deal more than the user about both the subject matter and strategies for using it to solve problems (e.g., UNIX consultation by agent)

2.4 Inferential capability

A trait often associated with agents is the ability to make inferences. By this is meant the ability to go beyond the specific, concrete instructions of the user and solve problems using some form of symbolic abstraction (Shoham, 1993). For some definitions, these abstractions are in the form of rule-based inferences, such as those used in expert systems. For other definitions, inferences can be based on probabilistic or case-based reasoning.

2.5 Prior knowledge of general goals and preferred methods

This trait can be viewed as an extension of both the "learning" and the "inference" traits. It has to do with the recognition in both human and AI problem solving literature that solutions to many real-world problems require an understanding of general goals (Wilensky, 1983, Maes, 1990). For example, many real-world problems are ill-defined in the sense that there is not, at the outset, a clear-cut path toward a solution. For such problems, many (e.g., see Adams, Tenney & Pew, 1991) have proposed a "goal-hierarchy" procedure, where solution paths can be altered in midstream based on an abstract understanding of what constitutes an adequate solution. This level of understanding is likely to be required for many of the more sophisticated versions of intelligent agents. Not only must an intelligent agent be capable of generating its own prioritized goal structure, but it must also understand the goal structure of the user for which it is acting as an agent (Chin, 1991).

2.6 Natural language

For simple problems, shell-like scripting languages may be a sufficient method for the user to specify a desired outcome and method to an agent. However, since many views assume that agents will solve a wide variety of everyday problems for users, the specification of those problems becomes a complex task. Both because of this complexity and because of natural user preferences, many writers assume a natural language interface (Shoham, 1993).

2.7 Personality

Some researchers have found it attractive to attach human-like personality characteristics to agents (e.g., Laurel, 1990, Bates, 1994). The assumption here appears to be that as the problem solving and interactive communication requirements become so great and complex, only a human-like entity could handle the problem. While personality may not logically be required for these complex tasks, the linkage between personality characteristics and cognition in humans is well established. Similarly in humans, communication of many kinds of information is closely related to personality.

Clearly the concept of intelligent agents is a complicated, multifaceted one. Despite agreement on the gist of the concept and a short list of common agent characteristics, there remains confusion over the term because the necessary and sufficient subset of traits is not agreed upon. This shouldn't detract from the generally shared view that software agents act on behalf of users by *autonomously* carrying out *delegated activities* made up of *multiple subtasks*. For agent software to be able to accomplish these activities autonomously, it will need to have domain expertise, knowledge of methods, strategies for subtasking, inferential abilities and decision-making skills -- in fact a good measure of the cognitive skills that make up *thinking* in humans.

That software agents can vary greatly in the degree to which they are *person-like* contributes to the confusion over their definition. It is easy to imagine software agents that work just fine but don't communicate hearing and producing natural speech, and that are not personified -- these are surely not necessary characteristics of intelligent agents. However, because delegation has been a fundamentally interpersonal activity, people may find it easier to delegate to, and work with, agents that are more, rather than less, human-like.

2.1 Agent Interfaces vs Agent Architectures

In addition to the loose set of traits associated with intelligent agents, another reason for confusion is that the term is used in two different senses. One uses "agent" to describe an element in a particular kind of distributed software architecture. An agent architecture is characterized by multiple, autonomous or background processes (Maes, 1990) that communicate with one another, often intelligently, to accomplish tasks (e.g., Bushman, Mitchell, Jones and Rubin, 1993; Norrie and Kwok, 1992; Werner and Demazeau, 1991; Speth, 1987; Riecken, 1994). It is a form of object-oriented software architecture (Shoham, 1993).

The "software architecture" sense of the term agent is quite distinct from its connotation as a user interface model in which the user delegates and collaborates. Many writers interchange these meanings, and many systems employ both agent architectures and agent interfaces. However, the software and user interface agents need not have a one-to-one relationship (Wittig, 1992). It is possible to implement an agent software architecture that has nothing to do with user interactions (see Maes, 1990, O'Hare and Jennings, 1996). Finally, there are intelligent agent user interfaces not implemented as distributed, autonomous processes (e.g., Chin, 1991).

3. Delegating to an Agent is not a Panacea

Distinguishing between user interface agents and software architecture agents avoids some of the confusion associated with the term. In addition, the distinction downplays defining agents by their traits, and highlights the mode of interaction between the user and agent. This mode of interaction is a social one and is based on delegation.

Delegation has been a popular topic in both the academic and popular literature of the organizational and management sciences; it is viewed as one of the most important of managerial activities. Delegation is the process of passing on responsibility for a task to a subordinate by giving him/her authority to act on your behalf, but without giving up control, or ultimate accountability (Jenks and Kelly, 1985). It is clear both from this literature and from the large number of popular management articles that managers of all levels fail to delegate as often as they ought to (Axley, 1992; Moore, 1982; Jenks and Kelly, 1985; Anderson, 1992; Gadsen, 1993; Gaedeke, 1992). Indeed, this literature suggests that, for many reasons, delegation is often an unnatural and taxing activity (Jenks and Kelly, 1985, Moore, 1982):

- Managers often feel that they can perform a task better than can a subordinate.
- Managers often enjoy doing certain tasks even if it may be more efficient to delegate them.
- For urgent tasks, needing to be done immediately, managers often believe that explaining the task to a delegate will waste time.
- Managers who are especially interested in or concerned about a task may not want to lose control. Since delegation involves autonomous work, simply not being constantly aware of the task's status can cause anxiety in managers.
- Some managers who feel that their own job depends on "looking busy" may do tasks themselves to avoid appearing idle.
- Managers view some tasks as so simple that delegation would complicate things. Delegation involves difficult and time consuming activities such as task explanation, monitoring, subsequent evaluation of the subordinate, etc.
- Managers may want to get the credit for a particular task for themselves, rather than for a subordinate.
- Managers fear that the subordinate will fail at the task; not only may the task not be accomplished, but the subordinate may feel badly
- Managers when delegating, worry that subordinates will think they are a tyrant

The frequency of failing to delegate, and the large number of reasons for the failure suggests that delegation among humans is not a natural or easy activity in many situations. The extent to which these difficulties apply when humans delegate tasks to a computer will only be known for certain after considerably more experience with intelligent agents. The reasons at the end of the list would seem to apply less well to computer-based intelligent agents than the others. For example, it is unlikely that the user of an intelligent agent will worry about competition, rivalry or emotions related to failure.¹ Similarly, users of intelligent agents do not have to exert the overhead associated with such things as planning personal and career development for computer agents. However, the majority of issues associated with delegation can apply equally well to human subordinates and computer-based intelligent agents.

4. Design issues for delegation-based interfaces

While delegation of work to intelligent agents may not solve all the users' problems, it's likely that this technology will continue to develop and find its way into users' lives. The appropriate strategy, then, for designers of intelligent agents is to understand the potential pitfalls of delegation and to design around them. The following are major issues related to delegation that are relevant to designers of intelligent agent user interfaces.

4.1 The benefits of delegation need to exceed the cost

Having an intelligent agent perform a task is not always easier than performing the task oneself. It is unlikely that intelligent agents in the near-term will be implemented with perfected technologies.

¹ Although, losing one's job to automation is a real fear that is likely to increase with widespread use of competent intelligence agents. Moreover, some have argued that effective artificial intelligence may eventually require emotional components.

Artificial intelligence and natural language capabilities are still rudimentary. Moreover, even if we had technologies developed enough to match the complex cognitive and rich communication capabilities of humans, it is questionable that delegation would always be easier than performing the task oneself. Some tasks are more appropriate for delegation than others.

There are costs and benefits associated both with the user doing tasks oneself and with delegating tasks to an agent, and they differ across tasks. A direct, quantitative cost/benefit comparison may be difficult, but, nonetheless, some kind of comparison is done regularly by managers as they decide whether to delegate or not.

Costs of self-performed tasks

One way to assess the cost of self-performed tasks is provided by task management formulations. Task management includes all of the processes and strategies employed by a human to accomplish multiple tasks overlapping in time. It includes cognitive strategies such as the switching of selective attention, use of memory structures to minimize disruptions due to interruptions, goal creation, etc. Task management approaches view workers as actively managing their time and cognitive resources as compared with more traditional "workload" formulations which view workers as a fixed set of limited resources applied passively across a set of tasks (Adams, et al., 1991). According to task management, the cognitive "costs" to the individual confronted with multiple tasks include:

*"maintaining the queue of to-be-attended tasks
updating the status of tasks in the queue
resolving conflicts among high-priority goals
planning optimal points at which to switch from task to task
calling up goal-related knowledge and information required to act
evaluating the significance of unexpected events
replanning to accommodate unexpected events"* (Adams, et al., 1991)

Of course, in addition to the cognitive costs of task management, there are costs involved in actually performing a task (Card, Moran and Newell, 1983). These include the time involved on the task and the time spent learning skills to do the task.

Costs of delegation

As with self-performed work, there are also costs associated with delegating work. Many delegation environments, especially those with low-level subordinates and close supervision (Moore, 1982), include precisely the same items listed above for self-performed work. Indeed, assuming a task management approach highlights that large degree of overlap between self-performing and delegation. However, in addition to the costs that overlap with self-performing, delegation involves some extra costs. These include:

Assessment of agent competence
Monitoring of agents work and progress
Communication of desired outcome and strategies
Anxiety associated with loss of control

Benefits

There are also benefits associated with both delegation and self-performed work. Many of the benefits of self-performed work are implied by the reasons for failing to delegate. They include enjoyment over doing a task one's self, and the likelihood, in some cases that the task will be done better. On the other hand, there is a major benefit associated with delegation: the efficiency and time management associated with not actually doing the task.

Design implications

A cost/benefit approach to agents has several design implications:

- Agents are more appropriate for some tasks than for others
Proponents of Intelligent Agents have argued that agents can be used for any task that requires expertise, skill or labor (e.g., Laurel, 1990). However, a cost/benefit analysis suggests that the use of agents be limited to tasks such as:
 - Tasks that take much time to do, but require little training or explanation time. Often, tasks like searching a series of databases for information concerning a term or simple group of terms can be time consuming and are easy to explain.
 - Tasks involving skills that, for the user, are not worth learning because, for example, of infrequent use (e.g. development of an application using a rarely used tool)
 - Tasks that involve some complex algorithms. There are tasks that are too tedious for the human to do, but also involve the use of some heuristic that may be too difficult for the computer to initiate unilaterally (e.g. real-time monitoring of a suite of status levels in an operations center)
 - Tasks that occur by simple but persistent schedules, and for which monitoring is not substantial (e.g., file-system backup).
- Users must have the option of delegating vs. self-performing tasks
The costs/benefit comparison of delegation is complex and fluid, arguing against a predetermination of whether the interface should be based on delegation or not. Instead, whenever feasible, users should be provided both with an intelligent-agent based interface and a tools-based alternative. This way users have the option of either performing tasks themselves or by delegation.

4.2 Delegation requires sophisticated, interactive communication

According to many of the management articles on delegation, a key responsibility of the manager is to communicate effectively and precisely so that the delegate understands the assignment. In particular, Jenks and Kelly (1985) have stressed that:

- managers describe tasks in terms of goals, intentions and results, not in terms of detailed steps. This allows the delegate to complete the task even if the original methods become unfeasible. Doing so, however, can be difficult for less well defined tasks.
- managers describe reasons for the delegated activities.
- while detailed steps should not be proscribed, general strategies for accomplishing the task can be useful to the delegate.
- describing a delegated task must be interactive, with both the manager and delegate providing input. This insures both that the delegate understands the goals of the task, and gives the manager an opportunity to assess the delegate's commitment to it.

This stress on goal-oriented explanations and interactivity in delegative communication make this area well suited to Winograd and Flores' (1987) use of speech act theory (Searle, 1969) for language interpretation and cognition by computers. Also similar is the view that communication is essentially a social process. In the speech-act approach, language is viewed not as a simple transference of information but as the creation of a "mutual orientation," i.e., a more general understanding of the context of shared goals and intentions. Much of the communication that takes place during delegation is language aimed at getting someone to do something, speech acts called directives. According to Winograd and Flores, human directives are often ambiguous and misleading. An efficient directive might begin with a request for action, get shaped with counter proposals by delegate and/or delegator, and end in a promise to act. An interactive conversation continues until an agreement is made.

The form of this "conversation for action" assumes that participants understand the content of each other's statements, which is not always the case. Beyond negotiating a directive, an additional interactive natural language capability is conversation repair. Sociolinguists have shown that people understand what others are communicating through an interaction process (e.g., Luff,

1990). During repair, the speaker and listener negotiate meaning through an iterative process of statement, question and response. This negotiation is not limited to verbal exchanges but includes also a rich array of intonation, facial and other non-verbal gestures that are used to signal comprehension. Some of these gestures convey comprehension or lack of it directly, while others are used to regulate the flow of conversation, sometimes in order to maximize comprehension (Ekman and Friesen, 1969; Chovil, 1992).

Design implications

- Users should be encouraged to convey the intentions and goals of a task to the agent. Attempts to create intention-oriented programming languages could be useful for user-agent communication as well as for agent-agent protocols (e.g., Rosenschein and Zlotkin, 1994).
- Natural language interfaces can be used for tasks that are easy to describe. There have been advances in natural and pseudo-natural language parsing that makes such systems attractive for unambiguous tasks. For many tasks, however, natural language parsing is currently too difficult and is often ambiguous.
- For many complex task environments, interface dialogues could employ Speech Act structure. Interfaces using Speech-Act structure must be carefully designed to avoid being cumbersome, a common complaint of existing such systems.
- Agents must be designed to indicate clearly when instructions are not understood. The process of communicating directives needs to be highly interactive.
- Anthropomorphizing agents with the use of facial displays and vocal intonation may be useful in conveying comprehension and lack of comprehension.
- For some tasks, it may be most efficient for the user to convey what is desired by demonstration (Maulsby and Witten, 1993).

4.3 Delegation requires trust

Since delegation involves ceding responsibility for a task but retaining accountability, trust is critical to any delegation-oriented interaction. In a organizations, trust determines how a manager decides whether to delegate, what to delegate, and to whom (Axley, 1992). Trust in computers is just as serious an issue, perhaps more so. Concerns have already been expressed in the popular press over trust in computer-based intelligent agents (Levy, 1993). Trust has long been an issue in designing decision support and control systems (Sheridan, 1980). Indeed, there is some evidence that people interacting with computers may be biased initially toward mistrust, whereas in social interaction the bias is toward trust (Muir, 1987).

In the social sciences literature, trust is defined broadly and multidimensionally. It involves expectations at three levels (Barber, 1983):

expectations that those we rely on are competent to perform their role. Assessment of competence is a continual behavior. People can assess separately others' expert knowledge, technical facility and performance of routine behaviors, and maintaining separate expectations for the three. McAllister (1995) has shown that the assessment of competence has both a cognitive basis (such as past performance), and also an affective basis (e.g., liking, similarity, etc.).

expectations that they are responsible and will make every effort to carry out their obligations -- These expectations go beyond observable performance and have to do with the delegate's motivations and intentions. Muir (1987) has suggested that this level of trust may not apply to computers. However, computer programs can involve intentions and often include parameters that are similar to motivations, such as the number of attempts it will make on a task before quitting, time-outs, and so forth.

expectations of persistence, i.e., that things will continue to act the way they have in the past -- This is a pervasive expectation about constancy in natural and social activities. The nature of trust changes as a human relationship develops and gradually becomes based more on intangibles (Rempel, Holmes and Zanna, 1985). Initially, trust is based on the perceived predictability of the other's behaviors. From that early stage, the relationship moves into a stage in which trust is based on a more general attribution of the other's dependability, especially for events involving

risk. Finally, a completely developed trust is based on faith in the other person's intrinsic motivations.

At all three levels, trust assumes that there is a form of self-interest within the trusted party. Between humans, for example, trustful expectations are often based on the assumption that the other party's self-interest (e.g., well-being, continued employment or reputation) will prevent him/her from engaging in irresponsible behavior. It is common for computer users to make social attributions that are at least as complex as these, even without thinking that the computer is human-like (e.g., Nass, Steuer and Tauber, 1994).

Design implications

In designing intelligent agent user interfaces, the goal is not necessarily to get the user to trust the agent, since both too little and too much trust are equally dangerous. Instead, the goal is to encourage the user to correctly calibrate trust in the agent to the task and situation. There are several things that can be done to accomplish this:

- Build agents to be reliable and use them in stable environments --
The most straightforward way to encourage a perception of predictability is to have them reliably accomplish the tasks expected of them. To do so may require iterative design techniques in order to fully understand what user's expectations are. Reliability also is dependent upon the task and environment: the most carefully designed of agents will appear unpredictable if it is working in an environment that keeps changing. People tend to underestimate the role of environmental variability and attribute problems to the actor (in this case the agent). In social psychology, this is called the fundamental attribution error (Ross, 1977). Because of this attribution error, agents deployed in more stable environments are likely to be trusted more.
- Create specialized agents capable of a small, circumscribed set of capabilities, and emphasize their expertise --
People might trust computers more if, as users, they could make their own allocation of tasks across computer systems (Muir, 1987). In many cases, trust in another is assessed relative to one's own self-confidence on a task (Lee and Moray, 1994; Wærn and Ramberg, 1996). Providing flexible allocation lets the user make use of this assessment. It also places the user in control, which can reduce alienation (Sheridan, 1980). For agent-based interfaces, flexibility can be achieved by providing a set of agents, rather than just one. However, this means the user has the additional burden of selecting the right system for the task at hand. Delegate selection is difficult for managers (Jenks and Kelly, 1985). It involves both clear understanding of the skills required for a particular task and also a mental inventory of each worker's skills and a grasp of their current workload and task priorities. In the design of intelligent agents, it would be useful to have the agent provide these two final pieces of information to the user.
- Increase the observability of the agent's behaviors --
If trust depends on behavioral predictability, at least early in a relationship, its calibration can be increased by giving the user access to as much of the behavioral data as possible (Muir, 1987). To some extent, this violates the autonomous essence of intelligent agents. However, it is consistent with the emphasis, in the management sciences, on monitoring and control techniques used during delegation. Some of these techniques are discussed in detail below. For now, it is sufficient to say that substantial monitoring is needed, especially in the early stages of a user-agent relationship
- Provide the user with data about the predictability of the agent's behaviors --
Overconfident judgments tend to be made on the basis of a too small a set of evidence, and people inappropriately concentrate on prediction-confirming cases at the expense of base rates. Both these biases can be overcome by providing the user with clear, summary measures of statistical reliability (Fong, Krantz & Nisbett, 1986). In some cases, this might be accomplished automatically by keeping track of delegation requests and their tangible

outcomes of success or failure. In other cases, however, an automatic process might be difficult since perceived success of a task is more subjective.

- Design the agent to evolve, with the user, through stages of trust --
Assuming that trust between user and agent evolves in a way similar to that between humans, a good design strategy is to have agent behaviors change accordingly. In the early stages of a relationship, agent behaviors should be very observable, but observability can decrease with experience. Also, risky tasks are especially notable in the middle stages of trust, so the agent might want to take special care to help the user feel comfortable while they are being done. Agents may go through similar stages of trust in a user. For example, early in the relationship, it may not be safe for agents to assume that a user's explanation of a task, or even choice of an agent, is a competent one, whereas later, there can be more trust.
- Train users to understand how the agent works --
Faith in and perception of a delegate's motivations are central to trust in human relationships. These assumptions about the other person's "internals" let trust move beyond observable behavior. A counterpart for user-agent interactions may be a user's understanding of how the agent works, (Muir, 1987). With an understanding of the internal rules and processes by which the agent accomplishes tasks, users are likely to feel more comfortable in lieu of observables. Teaching the user how the internals of a computer program function runs counter to the common field practice of hiding as many of the internals as possible from the user, but it may promote trust (Norman, 1987). Another way to accomplish this goal of understanding might be to design agents to accomplish tasks using the same means that the user would use. For example, if an agent is scheduling a flight, it might work through the same interfaces that are available to the user, rather than having its own special method.
- Use anthropomorphized agent interfaces --
It can be argued that the most effective way to convey trustworthiness and competence about an agent is via interpersonal cues that have evolved in humans specifically for that purpose. Many facial expressions, body gestures and vocal intonations are interpreted in terms of trustworthiness, and can be used (sometimes deceptively) to convey trustworthiness (DePaulo, 1992), although their interpretation and use are highly dependent on the context in which they are used. While social attributions can occur in the absence of most of these cues (e.g. Nass, Moon, Fogg, Reeves and Dryer, 1995), it has been shown that even simple, cartoon renditions of facial expressions can enhance interactions with agents, (Bates, 1994; Maes, 1994). Walker, Sproull and Subramani (1994) found generalized social facilitation with a computer-animated face as user interface. It is likely that more sophisticated displays can be useful in accurately reflecting trustworthiness and competence.

4.4 Performance controls are a key part of delegation

According to Jenks and Kelly (1985), the key to good delegation in a business environment is the controls the manager maintains on delegates' performance. Their concept of controls spans a range of behaviors and includes:

Preactivity Controls are management controls that prevent the delegate from making mistakes while performing the task. They include planning and strategizing about methods, and also the setting of performance criteria and scheduling of subtasks and deadlines.

Real-Time Controls are controls the manager uses while the task is being performed by the delegate. They include monitoring of behavior as well as midstream correction of methods and goals while the task is being done. Allan (1981) reports that monitoring of employees' ongoing behavior is rated as one of the six most crucial activities by managers of all levels.

Postactivity Controls include evaluation of the delegate's performance, as well as rewarding or punishing his/her behavior.

Design implications

There are several implications regarding performance controls for intelligent agents:

- Designers need to emphasize subtasking, scheduling and deadlines. These preactivity controls are especially important with agents of limited abilities and flexibility. Users may need to break work into smaller subtasks for intelligent agents than they would for self-performance, in order to be able to obtain adequate feedback on progress. Given the basic asynchronous nature of agent work, users need a way of scheduling and monitoring deadlines.
- Users need to be able to solicit and receive status reports at any time. This is a particularly difficult technical achievement in multi-agent architectures (Werner and Demazeau, 1991). If one agent receives a task and then delegates it to a second agent, the user needs to keep track of the task progress, preferably through interaction with the original agent.
- For some tasks, users need an independent way of checking on agents' performance while the task is being carried out. A parallel status channel may be needed for tasks where timely success is crucial to the user. Such a channel might also be useful in situations where the user has low trust in the agent.
- Users need a way of evaluating agents' performance in such a way that the agents subsequent performance will be improved or strengthened. Maes (1994) has recently experimented with methods an agent system can use to learn to do tasks better. In addition to observational learning and direct instruction, she has found user feedback on agent performance to be a potent learning technique.

4.5 Delegation depends on personality and culture

The relationship between personality traits and delegation effectiveness has not been studied directly. However, there is some evidence for a general relationship between personality traits and aggregate of behaviors of which delegation is often considered a part, namely leadership. According to a classic review (Gibb, 1954), leadership behaviors include (1) supervising and monitoring others' work, (2) planning, initiating and directing other's activities, (3) communicating well to subordinates (4) being accountable (5) showing loyalty to the group and (6) performing as a professional expert. Many of these activities are the same as those associated with delegation.

Personality of a Leader

The personality traits most closely associated with leadership (Gibb, 1954; Hogan, Curphy, and Hogan, 1994) are:

Self-confidence. Leaders tend to have higher self-confidence and self-assurance.

Extroversion. Leaders are extroverts. It is not as clear whether extroverts find it easier to engage in leadership behaviors, or are more often thrust into leadership roles.

Dominance. Leaders tend to be dominant and highly assertive, but some studies suggest that those people ranking highest in assertiveness are not effective as leaders

Empathy. Leaders tend to be good at estimating other group member's opinions and feelings.

Personality integration. Less strong relationships have been found between leadership and personality integration measures such as deliberate will control, perseverance, high achievement drive, decisiveness, self-identity, ego strength and the ability to avoid anxious worrying,

These personality traits determine leadership style (e.g., authoritarian, democratic, etc.) and, despite some successes (Nicholis, 1993), are often difficult to train (Gibb, 1954). However, these traits also interact with the situational variables and task at hand to determine which people will lead and delegate best.

Potent situational variables include the culture of the individual and the workplace's. Cultures differ widely both in individualism and in "power distance," the degree to which people feel comfortable while perceiving themselves organizationally distant from power (Luthans, 1981). Delegation in some cultures is likely to take the form of direct orders given by superior to subordinate. In other cultures, that form is inappropriate, and delegation is more supportive or even indirect.

Design Implications

Keeping in mind that delegation is only one component of a complex aggregate of leadership behaviors and cultural factors, several implications are worth considering when designing agents.

- Experience with delegation may make agent-based interfaces easier to use. Possibly, these interfaces may work best in applications utilized by managers or workers in leadership positions. Possibly, this type of user could have delegatory aspects of their applications emphasized more than other users.
- Managers who already are effective at delegating tasks to humans may prefer and excel at using intelligent agents, while ineffective delegators may be so with both human and computer-based delegates. Users should be given a choice, on a task basis, of either delegating or self-performing tasks
- Applications for international use need to be designed with careful considerations for cultural differences in leadership and delegation.

5. Summary and Conclusions

As the market popularity and potential for intelligent agents escalates, it has become increasingly difficult to define what an agent is. This analysis takes a different approach in focusing less on the traits and behaviors of agents and more on the interaction between agents and users. This interaction is most often based on delegation. While delegation is often useful, a review of the relevant literature and popular press reveals a number of difficulties. Delegation is often an unnatural and taxing activity. Delegation can be cognitively costly. It requires sophisticated communication, control mechanisms, and trust. Finally, the ability to delegate effectively depends on the delegator's personality and culture.

While one might speculate that it will be easier and more effective to delegate to a software agent than to a human, the same issues that apply to human delegation apply to delegating to agents. Development of useful and usable intelligent agent-based systems will require careful attention to user-agent interactions during the design phase. Our analysis has distinguished between intelligent agent user interfaces and software architectures, but many of the interface design issues discussed have architectural implications as well. For example, monitoring status requires that a communication channel be available to do so. Similarly, the degree to which agents are specialized in their functions affects the software design. Finally, there is still an incomplete understanding of the processes and behaviors associated with delegation, let alone its use with computers. The development of intelligent agent user interfaces provides an opportunity for computer scientists, application designers and behavioral scientists to collaborate in a more complete investigation of this common workplace activity that will, in turn, lead to more effective user interfaces.

References

- Adams, M.J., Tenney, Y.J. and Pew, R.W. (1991). Strategic Workload and the cognitive management of advanced multi-Task Systems, *Report for the Crew system ergonomic information analysis center (CSERIAC)*.
- Allan, P. (1981). Managers at work: a large scale study of the managerial job in New York city government. *Academy of Management Journal*, **24**, 613-619.
- Anderson, D. (1992). Supervisors and the "hesitate to delegate" syndrome, *Supervision*, November, 9-11.
- Axley, S.R. (1992). Delegate: Why we should, why we don't and how we can. *Industrial Management*, September-October, 16-19.
- Barber, B. (1983). *The Logic and Limits of Trust*, New Brunswick: Rutgers University Press.
- Bates, J. (1994). The Role of Emotion in Believable Agents, *Communications of the ACM*, **37**, 122-125.
- Bushman, J.B., Mitchell, C.M. Jones, P.M. and Rubin, K.S. (1993). ALLY: An operators associate for cooperative supervisory control systems. *IEEE Transactions on Systems, Man and Cybernetics*, **23** (1).
- Card, S.K., Moran, T.P. and Newell, A. (1983), *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum.
- Carroll, J.M. and McKendree, J. (1987). Interface design issues for advice-giving expert systems, *Communications of the ACM*, **30**(1) 14-31.
- Chin, D.N. (1991). Intelligent Interfaces as Agents, in J.W. Sullivan and S.W. Tyler, (Eds.) *Intelligent User Interfaces*, ACM Press.
- Chovil, N. (1992). Discourse-Oriented Facial Displays in Conversation. *Research on Language and Social Interaction*, **25**, 163-194.
- Chu, Y and Rouse, W.B. (1979). Adaptive Allocation of Decisionmaking Responsibility between human and computer in multitask situations., *IEEE Transactions on Systems, Man and Cybernetics*, **9**(12).
- DePaulo, B. (1992). Non-verbal behavior and self presentation. *Psychological Bulletin*, **111**, 203-243.
- Etzioni, O. and Weld, D. S. (1994). Softbot-Based Interface to the Internet. *Communications of the ACM*, **37**, 72-75.
- Ekman, P. and Friesen, W.V. (1969). The repertoire of nonverbal behavior: categories, origins, usages and coding. *Semiotics*, **1**, 49-98.
- Fong, G., Krantz, D. & Nisbett, R. (1986). The effects of statistical training on thinking about everyday problems. *Cognitive Psychology*, **18**, 253-292.
- Gadson, R. (1993). You need help: delegate or die in the 1990's. *Manager's Magazine*, **68**, 15-17.
- Gaedeke, A.R. (1992). Managers Handbook: Ease your burden--delegate, *Manager's Magazine*, **67**, 30-31.
- Gibb, C.A. (1954). Leadership, In G. Lindzey and E. Aronson, (Eds.) *The Handbook of Social Psychology*, vol.4. Reading, MA: Addison-Wesley,
- Hogan, R. (1994). Curphy, G.J. and Hogan, J., What we know about Leadership. *American Psychologist*, **49**, 493-504.
- Jenks, J.M. and Kelly, J.M. (1985). *Don't Do, Delegate!*. New York: Franklin Watts.
- Laurel, B., (1990). Interface Agents: metaphors with character. in B. Laurel, Ed. *The art of human computer interface design*. Reading, MA: Addison Wesley
- Leana, C.R., (1986). Predictors and Consequences of delegation. *Academy of Management Journal*, **29**/4; 754-774.
- Leana, C.R. (1987). Power Relinquishment versus Power Sharing: Theoretical clarification and empirical comparison of delegation and participation. *Journal of Applied Psychology*, **72** (2) 228-233.
- Lee, J.D. and Moray, N. (1994). Trust, self-confidence, and operators' adaptation to automation. *International Journal of Human-Computer Studies*, **40**, 153-184.
- Levy, S. (1993). Let your agents do the walking. *MACWORLD*, May, 37-42.

- Luff, P. (1990). Introduction, In P. Luff, N. Gilbert and D. Frohlich, (Eds.) *Computers and Conversation*, Academic Press.
- Luthans, F. (1981). *Organizational Behavior*. New York: McGraw Hill.
- Maes, Pattie, (1994). Ed. *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*. Cambridge, MA:MIT Press.
- Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, **37**, 31-40.
- Maulsby, D. and Witten, I.H. (1993). Metamouse: An Instructable Agent for Programming by Demonstration. In Cypher, A. (Ed.) *Watch what I do: Programming by Demonstration*, Cambridge, MA: MIT Press. 155-182.
- McAllister, D.J. (1995). Affect- and Cognition-Based Trust as Foundations for Interpersonal Cooperation in Organizations. *Academy of Management Journal*, **28**, 24-59.
- Minsky, M. (1985). *The Society of Mind*. New York: Simon and Schuster.
- Moore, F.G. (1982). *The Management of Organizations*. New York: John Wiley & Sons.
- Morris, N.M. Rouse, W.B. and Ward, S.L. (1988). Studies of dynamic task allocation in an aerial search environment. *IEEE Transactions on Systems, Man and Cybernetics*, **18**(3).
- Muir, B.M. (1987). Trust between humans and machines; the design of decision aids. *International Journal of Man-Machine studies*, **27**, 527-539.
- Nass, C., Moon, Y., Fogg, B.J., Reeves, B. and Dryer, D.C. (1995). Can Computer Personalities be Human Personalities?, Unpublished manuscript, Stanford University.
- Nass, C., Steuer, J. and Tauber, E.R. (1994). Computers are Social Actors, Paper presented at meetings of the ACM Computer-Human Interaction Group, Boston, MA.
- Negroponte, N. (1970). *The Architecture Machine: Towards a more Human Environment*. Cambridge, MA: MIT Press.
- Nicholis, J. (1993). The Jonico window--integrating leadership and delegation training. *Training & Management Development Methods*, **7**, 1-11.
- Norman, D.A., (1987). Some Observations on Mental Models, *Readings in Human-Computer Interaction*. R.M. Baecker and W. Buxton, (Eds.). Los Altos: Morgan Kaufmann. 241-244.
- Norman, D. (1994). How might people interact with Agents? *Communications of the ACM*, **37**, 68-71.
- Norrie, D. and Kwok, A. (1992). Intelligent Agent simulation of an automated guided vehicle system. In T.G. Beaumariage and R.K. Ege, (Eds.) *Proceedings of the 1992; Object-Oriented Simulation Conference*. San Diego: Simulation Council Inc.
- O'Hare, G.M.P. and Jennings N.R. (1996). *Foundations of distributed artificial intelligence*. New York: Wiley.
- Oncken, W. and Wass, D.L. (1974). Management time: Who's got the monkey? *Harvard Business Review*, November-December, 6-11.
- Rempel, J.K., Holmes, J.G. and Zanna, M.P. (1985). Trust in close relationships, *Journal of Personality and Social Psychology*, **49**, 95-112.
- Riecken, D. M. (1994). Intelligent Agents: Introduction, *Communications of the ACM*, **37**, 19-21.
- Riecken, D. M. (1994). An Architecture of Integrated Agents. *Communications of the ACM*, **37**, 107-116.
- Roesler, M. and Hawkins, D.T. (1994). Intelligent Agents, *Online*, July, 18-32.
- Rosenschein, J.S. and Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. Cambridge, MA: The MIT Press.
- Ross, L. (1977). The intuitive psychologist and his shortcomings: Distortions in the attribution process. In L. Berkowitz, (Ed.), *Advances in experimental social psychology* (Volume 10). New York: Academic Press.
- Roth, E.M., Bennett, K.B. and Woods, D.D. (1987). Human Interaction with an "intelligent" system. *Int. Journal of Man-Machine Studies*, **27**, 479-525.
- Rouse, W.B. Geddes, N.D. and Curry, R.E. (1987-88). An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems. *Human-computer interaction*, **3**, 87-122.
- Searle, J.R. (1969). *Speech Acts*. Cambridge: Cambridge University Press.
- Sheridan, T.B. (1980). Computer control and Human alienation. *Technology Review*, **83**, 60-76.
- Shoham, Y. (1993). Agent -oriented programming. *Artificial Intelligence* , **60**, 51-92.

- Speth, R. (1987). Ed., Message handling systems : state of the art and future direction :
Proceedings of the IFIP TC 6/WG 6.5 Working Conference on Message Handling Systems,
Munich, F.R.G., 27-29.
- Sproull, L., Subramani, R., Walker, J.H. and Kiesler, S. (1994). When the interface is a face,
Paper presented at meetings of the ACM Special Interest Group in Computer-Human
Interaction, Boston.
- Wærn, Y. and Ramberg, R. (1996). People's perception of Human and Computer Advice.
Computers in Human Behavior, **12**, 17-27.
- Werner, E. and Demazeau Y. (1991). (Eds.) , *Proceedings of the Third European Workshop on
Decentralized A.I. 3 Modelling Autonomous Agents in a Multi-Agent World*, Kaiserslautern,
Germany.
- Wilensky, R. (1983). *Planning and understanding: a computational approach to human reasoning*.
Reading MA: Addison-Wesley.
- Winograd, T. and Flores, F. (1987). *Understanding Computers and Cognition*. Reading MA:
Addison-Wesley Co..
- Wittig, T. (1992). (Ed.), *ARCHON: an architecture for multi-agent systems*. New York: Ellis
Horwood.
- Woods, D.D., Johannesen, L. and Potter, S.S. (1991). Human Interaction with Intelligent Systems:
An Overview and Bibliography, *SIGART Bulletin*, **2**(5).