When People Delegate

Allen E. Milewski and Steven Lewis

AT&T Labs

I had requested that all our information about the new calling plan be collected and be ready on my desktop for review by 8:00 am Friday morning. I was worried that I might not be well understood, and so I took great care to make myself very clear. It is now 10:00 am. I am at my desk, and I have no idea where that information is or if I’m going to get it in time. I think I need to rethink the value of my assistant.

Some technologically-immersed readers may be inclined to assume that the above passage depicts a computer user interacting with his or her intelligent agent. We might be surprised to find that the passage has nothing to do with computer agents, or even technology, and instead was overheard in a friend’s workplace from a manager referring to his secretary. The passage depicts part of a delegatory relationship between two humans, and serves as a reminder that delegation behaviors and concerns apply not only to software agents, but, of course, to a long tradition of human interaction as well.

This chapter is about delegation. It is about both delegation from a human to a computer and from a human to other humans. The starting point of this chapter is the assumption that the existing literature describing the processes involved when a principal delegates work to an agent has something to add to our understanding beyond what our intuitions may tell us about delegation.

Delegation is something that we all have some experience with, whether it be in a business-management setting, in an athletic context, or in a parent-child relationship (Engel, 1983). However, the prevalence of delegation as a social activity does not make its workings intuitive and obvious. First of all, everyday, personal experiences are seldom sufficiently broad to permit the creation of a psychology, at least not a psychology with much explanatory generality (Ross, 1977). But more specifically, judging by the large number of popular management publications with titles such as “How to Delegate”, “Why Managers Don’t Delegate” and “Are you Delegating Enough?”, delegation is something that people either do not understand well, or are not very good at. In this context it is surprising that so many explorations of computer software agents have given minimal attention to and/or citation of the literature on delegation. (e.g. Maes, 1994; Negroponte, 1997). In truth, these particular treatments have been insightful and useful. Nonetheless, a more in-depth understanding of delegation as it has been carefully studied by social and behavioral scientists and management practitioners can only help our application of the concept to computer agents – even if that understanding only serves to support intuitions.

The application of delegatory interaction models to computers came with great excitement as early as the 1950’s (e.g. see Bradshaw, 1997). However, the exploration of delegation predates and is broader than that found in the study of computer-based agents. Earlier, we have pointed out the relevance of the behavioral sciences in understanding delegation, especially organizational psychology and management sciences (Milewski and Lewis, 1997). In this chapter we reiterate that multidisciplinary approach, and expand upon it to include work on automation and psychology, a significant set of newer empirical studies on human-computer interaction, and to a lesser extent, economics and sociology.
The goal of this chapter is to gain a greater understanding of what delegation is, and when people do it. We will review both what is known about delegation between two humans with what is known about delegation to software agents. A more detailed understanding of delegation can be useful to help establish design guidelines for software agents (Milewski, and Lewis, 1997), to decide when agent-based interfaces are appropriate (Erickson, 1997), and more generally, to understand the issues associated with distribution of tasks between humans and machines (Parasuraman and Riley, 1997).

**What is Delegation?**

*Delegation is a kind of interaction done in a context*

Delegation has to do with entrusting authority and power to another in order to better manage workload and often implies a certain degree of autonomy in carrying out tasks. (Strauss, 1963). To understand delegation better, its useful to view it from several different standpoints. Sociology and economics, for example, have viewed delegation from the “higher” standpoint of the organization. In these disciplines, delegation has been treated as a job-design or labor-distribution strategy. (Holstrom and Milgrom, 1991; Itoh, 1994). Such strategies are chosen by companies or organizations in order to maximize the benefits to the company and the principal (delegator) given the costs of employing agents (delegates). It is possible, via this standpoint, to model the conditions under which delegation is worthwhile as well as some of the factors that affect its costs and benefits. Similarly in organizational psychology, delegation has often been viewed on a continuum of management decision-making strategies that includes, e.g., autocratic, consultative, participatory and delegatory (Vroom and Yetton, 1973; Luthans, 1981). Again, these strategies develop in the context of an organization and match its motivational structure.

In addition to the organizational standpoint is the approach that focuses on delegation as an interaction between people. One example is Engle (1983), who has used a transactional analysis approach to describe to managers how to negotiate delegation as a social interaction, and why it can go wrong. Also, Keller (1994) has reviewed evidence stressing the critical nature of the manager-employee social relationship in determining the fruitfulness of delegation. Finally, Winograd and Flores’ (1987) speech-act approach has attempted to formalize the linguistics of the interaction that takes place when work and responsibility is delegated.

This multi-standpoint approach to delegation has its parallels in the field of software agents. On the one hand, there are agent-based architectures. In many ways, these are similar to the “higher-level”, organizational standpoint of economics. The concerns associated with this standpoint are the strategies of designing complex systems, efficiencies, optimizations, etc. (Shoham, 1993)

On the other hand, much of what has been written about software agents has to do with the interaction between agent and user. Concerns for this standpoint are usability, predictability, control, etc. Kay (1990), for example, has described agents in terms of an “indirect management” style wherein users accomplish things by describing general goals to their agents. Erikson (1997) explores how users’ expectations affect the interaction with agents that are adaptive and responsive. Finally, both Norman (1997) and Lewis (1998) have analyzed agent interactions in terms of cognitive psychology.

It is worth noting the partial separability of the two standpoints. As can be seen above, each standpoint has its own concerns and implications. A companies’ decision to use a vertical-management system certainly means that someone has to do some delegating. But, it doesn’t say much about how to do it, or the specific problems involved. Similarly, the use of an agent-based architecture has only indirect effect on the nature of the interaction. One example is the
commonly used mail protocol SMTP, which uses an agent architecture of sorts, but works so
deterministically and quickly, that users do not think of getting their mail as delegating. At the
level of interface style, the user often need not know if some function is carried out by a society of
agents communicating with one another via a negotiation language, or is being done by just a
standard software program. Indeed, it is possible that the notable difficulties computer scientists
have had defining what an agent is are in part due to the mixing of these standpoints. For
example, in the taxonomy proposed by Nwana (1996), some of the traits have to do with
interactions, and others have to do with architecture.

However, these standpoints for delegation are not entirely separable. How a company is
organized can determine the assumptions about what kind of delegation needs be done (e.g.
supervisory, monitoring, cooperative, prescribing). Similarly, the kind of software agent under
consideration determines aspects of delegation. For example, mobile and information agents
(Nwana, 1996) may need to take more time to do their work than local agents, and therefore
monitoring may need to be different. Collaborative agents need to engage in negotiation with
other agents, so users may have more uncertainty about how problems are solved. Reactive
agents, which possess no goals of their own, require different assumptions.

We concentrate on the interaction aspect of delegation, and this limits the kinds of software
agents considered. In lieu of concise definitions for software agents, several writers have recently
attempted to lay out taxonomies of agents (Franklin and Graesser, 1996; Nwana, 1996;
Brustoloni (in Franklin and Graesser, 1996); Zachary, et al., 1996. Of these, the class most
pertinent to our discussion is Lieberman’s (1997) autonomous interface agents\(^1\). These agents
most often have clear delegation behaviors associated with them. But, again, be reminded that
these classes often overlap and are far from exclusive.

**Delegation is a Process**

Even though delegation is a kind of interaction, it cannot be understood simply be describing the
act of a principal requesting that an agent take on some responsibility. Instead, it is a complex
process that spans a variety of behaviors including preparatory activities, communication
behaviors, monitoring and feedback. For example, a variety of management consultants (Engles,
1983; Jencks and Kelly, 1985; Rowbottom and Billis, 1987; Axley, 1992; Nelson, 1994) agree
that it includes a broad set of controls that a manager maintains over delegates’ performance.
These include:

- Planning/strategizing about methods
- Setting performance criteria
- Scheduling deadlines and subtasks
- Midstream performance monitoring
- Performance evaluation
- Making decisions about incentives
- Keeping a running lists of competencies for agents and matching tasks to agents

\(^1\) some writers have taken “autonomy” to mean that the agent is entirely self-sufficient, i.e. tasks
aren’t even delegated, but instead are initiated by the agent. However, from a realistic
standpoint, it is likely that most work will be initiated by humans and directed to agents, in much
the same way that workplace tasks are directed by managers to semi-autonomous workers.
Effective, goal-oriented communications
Activities that build long-term trust
Establishing an appropriate authority relationship with the delegate (e.g. supervisory, monitoring, cooperative, prescribing)
Development of an affiliative relationship

Clearly, there is a lot going on across time that cannot be observed as part of direct interactions between delegators and delegates. Delegation is sometimes thought of as a mind-set. In this sense, a delegation-oriented relationship has a lot in common with other user mental models (Erickson, 1997). Mental models define an entire set of expectations about interactions and activity scripts that underlie behavior. They are also often incomplete and dynamic, evolving with experience (Norman, 1983). One implication of this is that the study of delegation, like that of mental models, must take a broad view: a view that includes a wide variety of behaviors, and a significant amount of time to allow for evolution.

Why might delegation to humans and to computer agents be alike?

In the last several years, thinkers in areas as diverse as philosophy and communications have converged on some conclusions about how humans think about, or at least behave toward, inanimate objects. For example, Dennett (1987) argues that humans dealing with objects are capable of taking on a variety of "stances" to help predict how best to interact. When taking the "physical stance", for example, people base their predictions on physical characteristics and physical laws. In contrast, the "design stance" organizes people's expectations around assumptions of what the object was intended to do by its creators. Yet another stance, the "intentional stance" predicts behaviors from what it would be rational for the object to do, given an assumed set of beliefs and desires. One of Dennet's contributions is in noting that taking on a stance is an optional behavior that people do when it is useful to do so. They can switch between stances in the course of interacting with an object. Yet, while stances define a competence, the attribution of intention is presumably not limited to passive expectations. It also drives the behaviors and styles of interaction of those taking a particular stance.

The program of research by Reeves and Nass (1996) confirms this approach with substantial empirical data. For example, Nass, Steuer and Tauber (1994) have used a "tutoring" scenario to demonstrate a series of well-established social rules that are maintained when one member of a human dyad is replaced with a computer. Having received tutoring from one computer, participants are more polite in their evaluation of the tutor when responding on the same device than when responding on a different computer, or via pencil and paper. This mimics politeness rules in human-human social interaction. Participants judged the quality of tutoring as better when it was praised by a different machine compared with when praise came from the same machine as did the tutoring. Similar results were found when different voices on the same machine were used to represent different social entities. These results, and others concerning projected gender differences, expertise differences and team-membership all demonstrate that people will make social assumptions about inanimate machines and its software.

It is difficult to avoid the conclusion that, in some way, participants in Nass and Reeves' studies are "borrowing" rules from one (social) arena and applying them in another (inanimate) one. Perhaps the most surprising aspect of this research program is that the conditions which trigger the borrowing are minimal: a small amount of interactivity and a somewhat conversational style. Indeed, it seems that simply placing the computer in a social role that humans normally take is often sufficient to elicit social behaviors.

Given this substantial data for correspondences in interacting with humans and machines, it seems quite plausible that users will tend to delegate to computers in much the same way they do to other humans, at least when it is useful to do so. With apologies to Dennett (1987), we
have found it useful to view this correspondence as a sort of “delegation stance”, (although, of course, it is perhaps more likely that delegatory behaviors simply display an example of his intentional stance). A delegation stance would be a set of behaviors and reactions that people take on when they wish to delegate, regardless of who or what is being delegated to.

Assuming such a stance, in the next sections we combine evidence from several disciplines in reviewing delegation. Doing so takes advantage of a broader span of data. It also capitalizes on unique advantages of each discipline. For example, organizational psychology views delegation in large complex interactions of real people. Automation human factors can carefully experiment on specific behaviors. Agent interface work can explore the effects of sophisticated, intelligent capabilities.

When do People Delegate?

Combining findings across the disciplines, we now review a set of conclusions about delegation. The following list is by no means exhaustive, but highlights some key factors about delegation.

1. **When delegates appear to perform well**

Several organizational psychology theories predict that managers will delegate differing amounts of authority and work to subordinates based on their perceptions of the competence and performance of the subordinate. It is not clear whether this is the unilateral result of a tendency among leaders to make relative competence comparisons between subordinates (Dansereau, Graen and Haga, 1975), and/or partly the result of a cycle wherein subordinates respond to supervisors’ attention with higher levels of performance, and are then given more attention (Keller, 1994). Regardless, there is evidence for this relationship from a variety of standpoints. For example, workers who are viewed as more competent by managers are trusted more (Butler, 1991). Subordinates who are viewed by their superiors as being in the “in” group of competent workers believe they are delegated to more (Liden and Graen, 1980). Also, managers report being more willing to delegate when they perceive the subordinate as having enough knowledge and information to perform well (Vroom and Jago, 1978; Leana, 1987). Finally, Leana (1986, 1987) has empirically investigated delegation amongst insurance adjusters, as measured by the autonomous settlement amount permitted by supervisors. She has found that the supervisors’ perceptions of subordinate capabilities were a strong predictor of delegation, even when those perceptions were not veridical. These results for delegation are in contrast to supervisors’ behaviors when they use “participatory” or collaborative management techniques, which involve less loss of control. Leana’s data support the notion that delegation is distinct from collaborative management since the latter tends to be used when subordinate competence is viewed as low.

Similar relationships between perceived performance and delegation have been found with computer interactions. Muir (1994) and Muir and Moray (1996) have found, for example, that participants, given the task of controlling a mock pasteurizing plant with potential help from a computer, rate their trust in computers as lower when the computer’s errors are variable rather than consistent. Even small errors reduce trust substantially. These errors not only affect judgments. Errors also result in less tendency to delegate tasks to the computer (and more tendency to do the task one’s self). Lee and Moray (1992, 1994) found similar decreases in trust when an accurate system started to make errors. Again, subjective trust was correlated with delegation. Using a computer-based game setting, Riley (1996, and Parasuraman and Riley, 1997) found circumstances where users continue to rely on automation even after serious failures, but confirm that automation is either turned off sooner, or delayed when users believe the system to be less reliable. Finally, Roberts (1998) reports that when given a choice, operators of a military monitoring system delegate less often to a software agent that either fails more often, or fails in ways that have more serious consequences.
Effects of perceived competence have not only been found when the task involves delegating actions to computers, but also when users take advice from computers to guide their own actions. For example, users of an advanced traffic-route planning system tend to believe and follow system advice more when it provides consistently accurate information. (Hanowski, Kantowitz and Kantowitz, 1994; Kantowitz, Hanowski and Kantowitz, 1997).

It should be noted that in most of the empirical research, reservations about delegation are not all-or-nothing effects of the subordinate’s performance. Roberts (1998), for example, found that delegation rates remained above 80% even after the agent performed errors with serious penalties. Instead, trust and, in many cases, delegatory behaviors are related to the magnitude or frequency of delegate errors, although the relationship is often nonlinear. (Lee and Moray, 1992; Muir and Moray, 1996; Riley, 1996; Parasuraman and Riley, 1997). Also, even after failures, users’ trust judgments (Lee and Moray, 1992) and delegation to the unreliable delegate (Roberts, 1998) can recover if the delegate’s performance improves, especially if previous experiences with the delegate were successful. These findings should not be surprising, since it is generally assumed that the tendency to delegate is mediated by one’s trust in the subordinate (Muir, 1994). Most theories of trust (Barber, 1983; Rempel et al, 1985; Zuboff, 1988) include the notion of a generalized expectation which, while related to performance, is more a function of longer-term, average performance rather than momentary behavior (Rotter, 1971). Observable competence remains a critical predictor of performance (Lenox, Roberts and Lewis, 1997). But, in parallel, an element of faith develops across time that is related to, but does not exactly match performance. This may explain why, in Leana’s (1986) study of human subordinates, it was supervisors’ perceptions of agent competence, not the competence itself that predicted delegation. While it is not clear if people generally trust computers more (Quintanar, et al., 1982), or humans more (Leach and Prietula, 1989), several writers have suggested an overall bias toward trusting (Muir, 1994). One might expect people to keep delegating even in the face of some errors.

In addition to faith, there are situations where even a small amount of delegation to subordinates of modest competence can actually be beneficial. In her study of human insurance agents, for example, Leana (1986) found that agents judged lowest in competence were delegated the authority to make autonomous settlements of only $500 while agents judged highly competent were allowed $30,000. Nonetheless, in practice all agents were viewed as contributing.

1.1. Monitoring

Frequency is not the only aspect of delegation that can be measured. Some researchers have measured the amount of monitoring that is done by supervisors. Presumably, a supervisor would monitor more when they are less certain that the agent will do the task. However, monitoring is a particular good measure because it points to the basic conundrum of delegation: even though a supervisor may delegate authority and workload as a time-management technique, there are times when the need to monitor subordinate performance takes so much time and effort that it detracts from other activities the manager must do. In these cases, the value of delegation can be null (Rouse, 1988; McAllister, 1994; Milewski and Lewis, 1997) or even negative.

Based on their experience with work organizations, many management consultants have listed monitoring as an essential, and often time-consuming aspect of delegation (Moore, 1982; Jencks and Kelly, 1983; Nelson, 1994). And, Allen (1981) reports that monitoring employee’s behavior is

---

2 The term “faith” is sometimes used to refer to an independently measurable form of trust that is separate from performance. It is based on emotion and affect (McAllister, 1995, Rempel, et al, 1985). It may be, though not clearly shown, that faith can develop from repeated reliability (Muir, 1994).
considered one of the most critical and time-consuming management activities. However, there does not appear to be a clear empirical picture of what factors affect the amount of monitoring done by supervisors of their employee’s ongoing work.3

There have been studies of monitoring with computer automation, however. Muir and Moray (1996) found that users checked on the computer’s performance more often when the computer made errors. Similarly, Roberts (1998) reports a significant trend for users to check the computer system’s accuracy more often in conditions where it was making multiple errors. Parasuraman and Riley, (1997) review several studies showing that when automation is reliable, it is monitored less, and failures are detected less.

Eidelkind and Papantonopoulos (1997) have performed a small study of automation that most closely mimics the tension of the real-world delegation conundrum described above. While in many automation studies, users are explicitly given little else to do while the computer is performing tasks for them, this study defined two tasks which were both required for success, but which were difficult to do simultaneously by the same party. Participants were continually required to collect and log certain data at brief intervals. In addition, certain thresholds needed to be monitored, and participants could either delegate or attempt to monitor themselves. Even when doing so affected their own logging performance, participants monitored the agents performance substantially after the agent made an error. Indeed, by varying penalties for failures on either the logging or monitoring tasks, Eidelkind and Papantonopoulos could show that participants traded-off monitoring with their own performance so as to maximize the combined performance on both tasks.

2. When Delegates are perceived as being trustworthy and confident

In addition to the actual performance record of subordinates, there are other factors related to the projection of competence that may affect delegation. Among these factors are trustworthiness and confidence. Organizational theories cite trustworthiness as a key factor in the tendency to delegate (Dansereau, Graen and Haga, 1975; Liden and Graen, 1980). McAllister (1995) cites affective (emotional and affiliative) aspects of competence other than mere performance. However, in real-world settings, the exact relationship between delegation and trustworthiness is not simple. Indeed, Leana (1986) found ratings of trustworthiness and competence to be highly correlated (and both predict delegation amount), although Leana (1987) found no predictive value in rated trustworthiness.

However, in experimental contexts people clearly make judgments based on perceived trustworthiness. Indeed, it appears that there may be different perceptual cues used to judge trustworthiness and competence. For example, Brownlow (1992) found that the facial appearance of a speaker affected how easily participants could be dissuaded from believing a speaker’s content. Participants could be easily convinced that baby-faced speakers lacked expertise, but not that they were untrustworthy. Conversely, mature-faced speakers could more readily be portrayed as untrustworthy, but their expertise was hard to undermine. Both DePaulo (1982) and Cialdini (1993) review extensive literature showing the effects of appearance and behavior on perceived trustworthiness.

Sundstrom and Salvador (1994) have directly assessed the effects of software agent confidence. Participants’ task was to diagnose faults in a hypothetical network system. Participants received

3 Recently, McAllister (1994) made an interesting attempt to measure monitoring, but results were unclear, especially for performance-based monitoring.
suggested diagnoses from a software agent, accompanied by a numeric confidence score. Higher confidence scores increased participants’ tendency to utilize the agent’s diagnosis, and also decreased the amount of information participants checked to make their final decision.

3. **When Delegates are perceived as more similar to the supervisor**

A final set of delegate characteristics affecting amount of delegation has to do with how similar the delegator perceives the delegate to be.

Similarity effects are abundant in social psychology. First, there is a large literature suggesting that people like and are attracted to other people who are similar to themselves. Therefore, they prefer to interact with similar people, and use them more in social comparisons to decide how to behave (see Cialdini, 1993). Similarity is a potent contributor to satisfaction in long-term emotional relationships such as marriage (Eysenck and Wakefield, 1981). In workplace organizations, members of the same informal groups, which are often similar, are perceived as being more trustworthy, honest and cooperative (Brewer, 1979). Finally, there is some evidence suggesting that ethnic and cultural similarity (Light, 1984) and gender similarity (McAllister, 1995) affect workplace behaviors and trust.

There is also evidence for similarity effects in studies explicitly exploring delegation. One common similarity studied by organizational psychology is the degree to which subordinates are viewed as sharing the same organizational goals as the supervisor. This measure has been assumed to be an underlying factor in perceived trustworthiness, and both Vroom and Yetton (1973) and Leana (1987) have found that it can predict delegation. In that same study, Leana has analyzed the effects of subordinate characteristics on managers’ judgments of whether delegatory or participatory management practices would work best. She found a tendency to prefer delegation when subordinates were older, had more job experiences and were men. While she does not report participant demographics for that study, her earlier experiment (Leana, 1986) on the same insurance work-function reports that most managers were men older than 40.

Similarity effects on delegation with computer automation and software agents present an intriguing possibility, but are difficult to find evidence for. This is, in part, because it is often not clear what kinds of attributions users are making about the software. So, it is not clear on what bases to measure similarity. It is easier to see similarity effects when certain attributions are coaxed onto the computer. One example of this is Nass, Moon, Fogg, Reeves and Dryer (1995). These researchers arranged for computers to appear either dominant or submissive by varying prose style, confidence and conversational assertiveness. Humans participants, themselves selected for high dominance or high submissiveness, interacted with the computers on a judgment game. Dominant participants rated dominant computers as being more like themselves, more intelligent, knowledgeable, helpful and useful. Similar results were found for submissive participants with submissive computers. This experiment stopped short of directly measuring delegation or the impact of the computer’s advice on participants’ own judgments. But the results are suggestive that delegation differences might be found.

Another suggestive example that users may find it easier to delegate to agents perceived as being similar to themselves is shown in a small experiment from our laboratory. The study had to do with the appearance of a humanized agent named PAT. PAT was an agent that helped business-workers locate colleagues in their organization that might have expertise in a particular topical area (and might be willing to help out on a project). PAT accomplished this quite simply by taking one or two “seed” names provided by the user, and iteratively either soliciting a work-agreement from those people, or getting further suggestions from them about who might better be able to help. That is, PAT manually carried out a “referral-chaining” process (Kautz, Selman and Milewski, 1996), and eventually returned the names of potential project-members to the originator. PAT was implemented as an animated humanized agent in the desktop application. To help select artwork, 30 naïve coworkers were shown the three renditions shown below
Milewski, A.E. and Lewis, S. H., *The Delegation Stance*. Unpublished manuscript prepared for J. Bradshaw, *The Handbook of Agent Technology*. (versions 1, 2 and 3 in Figure 1. That figure also shows our final artwork for the project). They were also given a paragraph briefly describing what PAT was designed to do, but phrased in “delegatory” terms (i.e. “Since you might be too busy to ask a lot of people, you just need to explain what you want and PAT will go out and ask them for you”). Then, participants were asked to circle the picture of their preferred PAT. The results are shown in Figure 1. Nearly all participants chose either version 2 or 3. Version 1 was described as being too “weird”, “strange” or “space-cadet-like”. Interestingly, while versions 2 and 3 were chosen approximately equally, there was a significant age difference in selections. Participants selecting version 2 were older than those selecting version 3. A separate sample estimated the mean age of version 2 to be 32 years old and the age of version 3 to be 24 years old, suggesting a similarity effect.

![Figure 1. Agent Renditions and Preferences](image)

4. **When Delegators are Experienced or Confident in the Situation**

Delegation involves both a loss of control over tasks, and an increase in the ambiguity of a work situation because the delay in seeing results can be lengthy. Jencks and Kelly (1985), for example, list loss of control as one of the key reasons for a manager failing to delegate. And, the emphasis on monitoring and evaluation in management consultants stresses ambiguity as an issue. As a result of these factors, the ability of a manager to delegate well requires a great deal of personal confidence and knowledge about the task environment.

There is a large body of studies on leadership characteristics that bears on delegation and confidence (Gibb, 1954). Since taxonomies of leadership behaviors almost always include delegation, monitoring, managing, evaluating, etc. it is likely that leadership and delegation
overlap. And, there are several indications that leadership effectiveness is related to certain personality traits, two prominent ones of which are self-confidence and tolerance for ambiguity. (Hogan, Curphy and Hogan, 1994). Presumably, these traits influence delegation by making the delegator more comfortable with the loss of control. For example, its likely that delegators who are more self-confidence are less threatened by the prospect of competition or rivalry, or obsolescence (Anderson, 1992).

While it may be possible to explain some of the delegation situation in terms of semi-stable traits of the delegator, recent analyses have shown that supervisor-subordinate interactions are often more situational (Leana, 1986; Keller, 1997). In this vein, there is another aspect of self-confidence that is less of a trait, and more situational. It has to do with confidence associated with expertise in a particular situation.

The role of expertise in management delegation has not received substantial attention in organization theories. However, a number of management practices advice and text books list expertise as necessary for delegation. For example, there are claims that a manager needs topical expertise in order to plan work and monitor effectively (Jencks and Kelly, 1985) and also to be respected by subordinates (Moore, 1982). In addition to expertise in the topical area, there is a pervasive assumption in these books that managers learn to delegate better as they gain administrative experience (Engle, 1983; Rowbottom and Billis, 1987).

In computer automation and agent interactions, there is also evidence that expertise is associated with ease of delegation. For example Riley (1996) has shown significant differences between pilots and students in their tendency to delegate to an agent. The pilots are presumed to be more generally experienced with automation than students, although it is not clear how much specific transfer one might expect from piloting to the experimental task used. And Lewis (1997) has reported on survey data in which computer users were asked which features of popular desktop software they prefer. Some features acted more autonomously and gave the user less control (e.g. self-correcting spell checker). Others required more manual effort but also more control for the user. Often, it was the experienced users who chose to lose control and delegate the tasks, while the inexperienced users preferred more manual control. It seems unlikely that experienced users are showing a mindless, blind faith in technology. It seems more likely that experienced users have become more self-confident in using desktop features in general. Perhaps they have learned how auto-features usually work, which are useful and have little risk, and perhaps how to monitor their success more easily.

The conclusion that self-confidence results in better delegation does not apply directly to all studies of delegation, however. Lee and Moray (1994) for example, found that participants delegated tasks to an automatic system less when they were more self-confident that they could do the task themselves. The investigators used a similar explanation (in 1992) to explain why participants used automation more after committing errors, even though they trusted it less. Riley (1996) found that participants frequently invoke their confidence in their own abilities as a reason for not using automation. Finally Waern and Ramberg (1996) found that participants’ confidence in a computer’s advice decreased as their own self-confidence increased. It seems intuitive that users may not delegate when they excel at performing the task themselves. These findings then, rather than being contradictory, may just serve to emphasize the basic cost/benefit analysis that goes on when a delegator is deciding whether to delegate or not (Milewski and Lewis, 1997).

Presumably, the arrangement in these experiments led participants to believe that performance could be improved by self-performing when they felt they had the abilities. When they felt less capable, the situation may have provided no negative consequences for delegation. For example, it is unlikely that participants felt threatened by the automated system. This decision, no doubt, depends on the context, and other similar experiments with slightly different contexts have found no effect of self-confidence on delegation (Riley, 1996, Roberts, 1998). Regardless, a self-confident participant performing a task him/herself does not imply that they would not be comfortable delegating if the context were different.
5. When the amount and nature of the work requires it

It has been noted by several writers that delegation is not a panacea, but has its own cost/benefit ratios that managers need to analyze on a regular basis (Moore, 1982; Milewski and Lewis, 1997). On the one hand, delegation has been touted as a time-management device by management practitioners because delegated work is not performed directly by the superior (Oncken and Wass, 1974; Moore, 1982). On the other hand, there are also costs associated with delegation. In addition to many of the same task-management costs of self-performing, the superior needs to assess, monitor and evaluate delegates performance. In addition there may be cognitive costs associated with the loss of control (Milewski and Lewis, 1997).

These tradeoffs associated with delegation are shown in the relationship between a delegator’s workload, and his/her tendency to delegate. If the delegator’s workload is high, s/he is forced, all things being equal, to delegate more than if there is free time to do tasks one’s self.

The use of delegation as a strategy to cope with high workloads has certainly figured into traditional management theory (Vroom and Jago, 1974; Oncken and Wass, 1974). Leana (1986), in her empirical study of insurance workers found a significant correlation between objective workload measures and supervisors’ ratings of delegation. In a computer automation context, workload determines use of automation, but in complex interaction with other factors. Parasuraman and Riley (1997), for example, review both positive and negative results for workload, and conclude that the individual and context need to be taken into consideration.

It is a prevalent theme in both the organizational psychology literature and that of economic models that delegation is the inevitable consequence of a principal having more to do than can be done by him/herself. This is because workload is high in these areas. For example, Itoh (1994) deals with the situation where a principal starts his own company and quickly finds it impossible to do all the work required. Decisions at that point must include some degree of delegation, and the questions involve not whether to delegate, but how much and what subtasks to delegate. Lewis (1997) has noted a similar attitude in industrial automation.

Given this view of inevitability, it would seem that writers who have argued against the use of agent-based interfaces, in favor, say of direct manipulation interfaces (Schneiderman, 1997) must have a limited notion of the role of computers in our future lives. If for example, the role of computers is constrained to current personal tasks like word processing, calendaring and web-surfing, then the limitations of a direct manipulation interface (Bradshaw, 1997) may not be overwhelming. But, if pervasive computing begins to increase the number of things we expect to do in our daily lives, then delegation of some sort is likely to become unavoidable, whether it be to entities called agents or not.

But delegators are also sensitive to which tasks to delegate. For example, delegation strategies depend on which tasks are critical. Both Heller (1973) and Leana (1996) report that supervisors tend to delegate more when tasks are less organizationally important. For Leana’s insurance study, for example, important tasks were large-dollar settlements likely to be contested. Eidelkind and Papantonopoulos (1997) found that participants’ delegation and “trust-after-an-error” could be driven by the relative contribution of the software agent’s task and participant’s task to overall reward. In some contexts, importance has to do with the risk of penalty. Riley’s (1996) experiments found that after a failure on a high-risk task, pilots preferred to perform the task themselves, and took time to start to delegate again. The same finding was not found for lower risk tasks. Finally, Moffa and Stokes (1997) have replicated standard automation tasks in a medical environment and have noted that trust ratings decreased more for consistent errors than is typically found in industrial settings (e.g. Muir and Moray, 1996) where such errors may not have such serious, immediate consequences.
6. When delegation rewards the participants

Incentives and rewards are a key element of successful delegation, and they have both direct, short-term influence as well as more indirect, longer-term influence on the process. Obviously, there are direct consequences of rewarding delegates for good performance; evaluating and rewarding delegates have been listed among the important controls that managers must maintain over subordinates (Engles, 1983; Jencks and Kelly, 1985; Rowbottom and Billis, 1987; Nelson, 1994). Moreover, the reward structure for the delegator can directly determine whether delegation is pursued or not. When a delegator is deciding to delegate, workload may interact with his/her incentive structure, i.e. how performance evaluation and reward is done. Quantitative modeling of economic principal-agent theory has shown that when principals are rewarded for “aggregate” performance (i.e. of both principal and agent combined), the principal’s most effective decision is often to delegate more and spend time on other work items (Itoh, 1994). However, if individual performance can be measured and rewarded, it often is most rewarding to try to self-perform more of the tasks. In real-world organizations, aggregate performance is often the easiest to measure and what matters most, so that delegation is common.

Eidelkind and Papantonopoulos’ (1997) experiment with software agents is somewhat consistent with this. Participants traded-off performance on subtasks to maximize aggregate performance when aggregate performance was being rewarded. The reward contingencies are not always so clear in other empirical studies. It would be interesting to explicitly compare delegation under conditions of rewards for aggregate and individual contribution.

In addition to direct incentive effects, there are some indirect effects that may reveal themselves only across a longer period of delegatory interactions. One example has to do with loss of control on the part of the delegator. While delegation may generally involve the loss of control on individual decisions and tasks, the evaluation and reward portions of the process provide a way for the delegator to reestablish control. (Engel, 1983). The advantage to the delegator for evaluation is longer-term, however, because evaluation often occurs well after tasks have been completed by the delegate.

Another example is when delegation, itself, is used as an incentive for the delegate. Many of the management advice books (Moore, 1982, Jencks and Kelly, 1985) have coaxed managers to delegate, not just because it reduces workload, but because it increase the morale of subordinates and their development as employees.

Morale is increased because delegates, empowered to make their own decisions, develop a greater sense of efficacy, control and self-worth (Keller, 1994). In contrast, for example, to participatory decision-making strategies, in which a superior may collaborate with subordinates on decisions, delegation involves delegate autonomy (Locke and Schweiger, 1979, in Leana, 1997). And Vroom and Yetton (1973) have predicted better performance in employees who receive more delegated power, as long as they have the competence and attitude to take advantage of it. This form of delegation incentive has been incorporated into economic modeling as well (Aghion and Tirole, 1997).

In addition to building morale, delegation as an incentive is often also advised as a means of developing the skills and leadership capabilities of the subordinate (Nelson, 1994, Moore, 1982). This delegator activity may reflect a sense of responsibility for the personal development of the subordinate (McAllister, 1995). But, it is also likely that managers use delegation as a form of long-term planning since a subordinate with more skills and high morale is likely to be more productive in the future. The cost/benefit ratio for delegation to humans, then, includes not just the immediate factors. Delegators can treat the costs of delegating as an investment to be redeemed later.
Again, Leana (1986, 1987) provides some empirical evidence for these notions in a real-life work setting. In both of her studies, those workers who were delegated more decision-making authority subsequently tended to perform better on their jobs. This relationship was not true for participatory decision-making strategies. Further, the relationship was not the result of the tendency of managers to delegate to more competent subordinates. The relationship between delegation and later performance held true even after the competence judgments of supervisors were partialled out.

There have been impressive examples of software agents that can learn from their environment and from their user. For example, Maes (1994) reviews work on a variety of agent-based systems that learn users’ preferences. These systems simplify tasks associated with messaging, calendar scheduling and information retrieval. Maes’ strategy involves agents gathering evidence from several sources, including: watching the user’s behaviors, receiving explicit instruction from users in the form of examples, communicating with other agents and receiving direct and indirect feedback from users. Similarly, Hermann (1966) and Tecuci and Hieb (1996) demonstrate the importance of user actions and feedback for the performance of software agents with machine learning capabilities.

However, the role of feedback when using current software agents seems quite different than that of incentives when people delegate to other people. First, it is unlikely that users of software agents think of themselves as actively providing incentives for their agents’ performance. While a central theme in human psychology has to do with why people behave at all, generally, these things are hardwired into software programs (Franklin and Graesser, 1996). The typical assumption is that computers will do what they are told. Computer learning systems do not rely on the human user to incent the system to learn and/or act on what is learned. While agents have been given sophisticated goal structures (Maes, 1990), less attention has been given to user’s comprehension and shaping of those goals. Given the importance of incentives when delegating to humans, one can ask whether the experience of delegating to a software agent is fundamentally different because incentives are not a concern. Software agents may be the best of all situations since they provide the opportunity to delegate work, without requiring the overhead of providing incentives. On the other hand, users might experience a greater sense of control and trust, if they feel they are incenting the agent to act (Maes, 1994).

The second difference has to do with the long-term aspects of incentives in delegation. As Moore (1982) has noted in an organizational setting, “delegation is a continuing relationship”, and managers can be rewarded in the long-term with improved delegate performance. First, however, they may need to incur some of the short-term costs associated with training, empowering and morale-building. We neglected these long-term effects in our earlier cost/benefit analysis of agents (Milewski and Lewis, 1996) because they require such significant learning capabilities. Indeed, it seems unlikely that users of current computer agents experience enough improvement to be able to take such a long-term view. However, there is evidence that longer-term “relationships” can be formed with computers (Moravec, 1998), and as future software agents become more able to learn from their delegation experiences, this aspect of incentives may join the other similarities in delegation to humans and agents.

Summary

We have reviewed evidence about selected topics in delegation from a variety of disciplines. The goal of understanding delegation has been to facilitate the development and appropriate use of software agents. We have found that delegation occurs as a function of perceived delegate performance, trustworthiness and similarity to the delegator. We have also found effects of delegator confidence and experience, task type and load, and of rewards and incentives.

One fact that became clear was that delegation depends upon complex interactions between factors. The topics we covered are traditionally categorized as delegate characteristics, delegator
characteristics and situational variables (Leana, 1986). But, we found characteristics difficult to categorize. For example, delegate characteristics such as competence, trustworthiness and similarity all have to do with the delegator’s perceptions. On the other hand, traditional delegator characteristics such as self-confidence are largely situational. Finally, situation characteristics such as workload depend upon the individual. We began by assuming a “delegation stance”, that is, that people may delegate in much the same way regardless of whether the delegate is a human or a computer. Taking such a stance is a useful way to combine evidence from several disciplines. Indeed, there are many ways in which delegation to humans and to computers is alike. There are also some ways in which delegation to current computer agents differ from delegation to humans. These include the role of incentives and the long-term, relationship-like nature of delegation when done between humans. It is possible that these too will become areas of similarity as agents become more sophisticated.
References


Axley, S. R. Delegate: why we should, why we don’t and how we can. Industrial Management, 1992, 16-19.1992


