

TOWARD AN INTERDISCIPLINARY THEORY OF
COORDINATION

Thomas W. Malone
Kevin Crowston

CCS TR# 120
SS WP# 3294-91-MSA

April, 1991

Table of Contents

1. INTRODUCTION.....	1
2. COORDINATION THEORY	2
2.1. What is coordination?.....	2
2.2. What is coordination theory?	3
2.3. Foundations of coordination theory.....	6
2.3.1. Computer science.....	6
2.3.2. Economics and operations research.....	7
2.3.3. Organization theory	8
2.3.4. Biology.....	8
3. TOWARD A FRAMEWORK FOR COORDINATION THEORY	9
3.1. Components of coordination.....	9
3.2. Basic coordination processes.....	12
3.2.1. Goal selection and goal decomposition	13
3.2.2. Resource allocation (managing resource interdependencies).....	14
3.2.3. Sequencing and synchronizing (managing timing interdependencies).....	14
3.2.4. Managing other interdependencies.....	15
3.3. Processes underlying coordination.....	16
3.3.1. Group decision-making.....	17
3.3.2. Communication.....	17
3.3.3. Perception of common objects.....	18
3.4. Example: Analyzing the task assignment process	18
3.4.1. A generic task assignment problem.....	19
3.4.2. Possible coordination mechanisms.....	19
3.4.3. Applying these models to various kinds of systems	20
3.5. Summary of coordination processes	20
4. APPLICATIONS OF COORDINATION THEORY	20
4.1. Understanding the effects of information technology on organizations and markets.....	20
4.1.1. Firm size.....	23
4.1.2. Centralization of decision-making.....	23
4.1.3. Internal structure of organizations.....	24
4.2. Designing cooperative work tools.....	25
4.2.1. Using coordination concepts from other disciplines to suggest design ideas.....	25
4.2.2. A taxonomy of cooperative work tools.....	26
4.2.3. Analyzing incentives for using cooperative work tools	28
4.3. Designing distributed and parallel processing computer systems	28
4.3.1. Analogies with social and biological systems as a source of design ideas.....	28
4.3.2. Analyzing stability properties of load sharing algorithms.....	30
4.4. Summary of applications.....	30
5. RESEARCH AGENDA	30
5.1.1. Representing and classifying coordination processes.....	31
5.1.2. How general are coordination processes?	31
5.1.3. Analyzing specific processes.....	31
5.1.4. Applications and methodologies.....	32
6. CONCLUSIONS	32
7. ACKNOWLEDGEMENTS.....	33
8. APPENDIX:.....	35
References.....	37

ABSTRACT

This paper characterizes a new research area, called *coordination theory*, that focuses on the interdisciplinary study of coordination. Research in this area uses and extends ideas about coordination from disciplines such as computer science, organization theory, operations research, economics, linguistics, and psychology.

In the framework presented here, coordination is analyzed in terms of *actors* performing *interdependent activities* that achieve *goals*. A variety of processes are analyzed from this perspective and commonalities across disciplines are identified. Processes analyzed include *goal decomposition*, *resource allocation*, *synchronization*, *group decision-making*, *communication*, and the *perception of common objects*.

A major section of the paper summarizes recent applications of coordination theory in three different domains: (1) understanding the effects of information technology on human organizations and markets, (2) designing cooperative work tools, and (3) designing distributed and parallel processing computer systems. In the final section of the paper, elements of a research agenda in this new area are briefly outlined.

Toward an Interdisciplinary Theory of Coordination

1. INTRODUCTION

In recent years, there has been a growing interest in questions about how the activities of complex systems can be coordinated (e.g., Huberman, 1988b; Johansen, 1988; NSF-IRIS, 1989; Rumerhart, et al., 1986; Winograd and Flores, 1986). In some cases, this work has focused on coordination in parallel and distributed computer systems; in others, on coordination in human systems; and in many cases, on complex systems that include both people and computers.

Our goal in this paper is to summarize and stimulate development of theories that can help with this work. This new research area—the interdisciplinary study of coordination—draws upon a variety of different disciplines including computer science, organization theory, management science, economics, linguistics, and psychology. There is still no widely accepted name for this area, so we will use the term *coordination theory* to refer to theories about how coordination can occur in diverse kinds of systems. Many of the researchers whose efforts can contribute to and benefit from this new area are not yet aware of each other's work. Therefore, by summarizing this diverse body of work in a way that emphasizes its common themes, we hope to help define a community of interest and to suggest useful directions for future progress.

There are at least three reasons why work in this area is timely:

- (1) In recent years, large numbers of people have acquired direct access to computers, primarily for individual tasks like spreadsheet analysis and word processing. These computers are now beginning to be connected to each other. Therefore, we now have, for the first time, an opportunity for vastly larger numbers of people to use computing and communications capabilities to help coordinate their work. For example, specialized new software has been developed to (a) support multiple authors working together on the same document, (b) help people display and manipulate information more effectively in face-to-face meetings, and (c) help people intelligently route and process electronic messages (see detailed references in section 4.2).

It already appears likely that there will be commercially successful products of this new type (often called "computer-supported cooperative work" or "groupware"), and to some observers these applications herald a paradigm shift in computer usage as significant as the earlier shifts to time-sharing and personal computing. It is less clear whether the continuing development of new computer applications in this area will depend solely on the intuitions of successful designers or whether it will also be guided by a coherent underlying theory of how people coordinate their activities now and how they might do so differently with computer support.

- (2) In the long run, the dramatic improvements in the costs and capabilities of information technologies are changing—by orders of magnitude—the constraints on how certain kinds of communication and coordination can occur. At the same time, there is a pervasive feeling in businesses today that global interdependencies are becoming more critical, that the pace of change is accelerating, and that we need to create more flexible and adaptive organizations. Together, these changes may soon lead us across a threshold where entirely new ways of organizing human activities become desirable.

For example, new capabilities for communicating information faster, less expensively, and more selectively may help create what some observers (e.g., Toffler, 1970) have called “adhocracies”—rapidly changing organizations with highly decentralized networks of shifting project teams. As another example, lowering the costs of coordination between firms may encourage more market transactions (i.e., more “buying” rather than “making”) and, at the same time, closer coordination across firm boundaries (such as “just in time” inventory management). To understand and take advantage of these new possibilities will almost certainly require major extensions or reformulations of our current theories of organizations, of markets, and of management.

- (3) Much recent activity in computer science has involved the exploration of a variety of distributed and parallel processing computer architectures. In many ways, physically connecting the processors to each other is easy compared to the difficulty of coordinating the activities of many different processors working on different aspects of the same problem. An intriguing possibility here is that lessons learned about coordination in social and biological systems can be applied to coordinating networks of computer processors.

For these possibilities to be realized, a great deal of cross-disciplinary interaction is needed. In the remainder of this paper, we define coordination theory in a way that emphasizes its interdisciplinary content, suggest a framework for developing it further, and then describe how coordination theory has already been applied to the three domains just described: (1) understanding the effects of information technology on human organizations and markets, (2) designing cooperative work tools, and (3) designing distributed and parallel processing computer systems. In the final section of the paper, we will briefly suggest elements of a research agenda for this new area.

2. COORDINATION THEORY

2.1. What is coordination?

We all have an intuitive sense of what the word “coordination” means. When we attend a well-run conference, when we watch a winning basketball team, or when we see a smoothly functioning assembly line we may notice how well coordinated the actions of a group of people seem to be. Often, however, good coordination is nearly invisible, and

we sometimes notice coordination most clearly when it is lacking. When we spend hours waiting on an airport runway because the airline can't find a gate for our plane, when the hotel room we thought had been reserved for us is sold out, or when a company fails repeatedly to capitalize on innovative ideas its researchers develop we may become very aware of the effects of poor coordination.

In order to proceed, it is helpful to have a more precise idea of what we mean by "coordination." The appendix lists a number of definitions that have been suggested for this term. The diversity of these definitions illustrates the difficulty of defining coordination, and also the variety of possible starting points for studying the concept. For our purposes here, however, we believe it is most useful to start with the following simple definition:

Coordination is the act of working together.

We will refer to this as the "broad" definition of coordination, and we will suggest a more restrictive "narrow" definition below. In both cases, however, we believe it is useful to use the word "coordination" in a fairly inclusive sense. For instance, even though the words "cooperation" and "collaboration" each have their own connotations, much of what they describe falls within this broad definition of coordination. It is also often possible to analyze "conflict" and "competition" from the point of view of some higher level goals and see that conflict at one level leads to coherent results at another level. Metaphorically speaking, for instance, two hands clapping may seem to be opposed to each other, but their combined activity results in a clapping sound. As an organizational example, different groups in a company may compete for resources and people, but this very competition may contribute to the company's overall ability to produce useful products (e.g., Kidder, 1981). Even though it is often important to distinguish between concepts like cooperation, collaboration, and competition, including them all in the same general framework allows us to examine their relationships and their relative advantages and disadvantages.

2.2. What is coordination theory?

We define *coordination theory* as a body of principles about how activities can be coordinated, that is, about how actors can work together. It is important to realize that there is not yet a coherent body of theory in this domain. However, there are theories, concepts, and results from many different fields that can both contribute to and benefit from the development of such general theories.

For instance, it is clear that questions about how people coordinate their activities are central to parts of organization theory, sociology, social psychology, anthropology, linguistics, law, and political science. Important parts of economics and management science also analyze how people can coordinate their work with a special focus on rational ways of allocating resources. Computer science does not deal primarily with people, but different computational processes must certainly "work together," and, as numerous observers have pointed out, certain kinds of interactions among computational processes resemble interactions among people (e.g., Fox, 1981; Hewitt, 1986; Huberman, 1988a; Huberman, 1988b; Miller and Drexler, 1988; Smith and Davis, 1981).

These potential overlaps suggest that coordination theory will be like other interdisciplinary fields (such as cybernetics (Wiener, 1961), and cognitive science (Gardner, 1985)) that arise from the recognition of commonalities in problems that have previously been considered separately in different fields. For instance, the field of cognitive science grew out of the recognition by researchers in several different fields (e.g., psychology, computer science, and linguistics) that they were dealing separately with similar problems: how can information processing systems (people or computers) do things like use language, learn, plan, remember, and solve problems (e.g., see Gardner, 1985; Norman, 1980)?

In coordination theory, the common problems have to do with coordination: How can overall goals be subdivided into actions? How can actions be assigned to groups or to individual actors? How can resources be allocated among different activities? How can information be shared among different actors to help achieve the overall goals?

Identifying and analyzing coordination processes. One of the most intriguing possibilities of coordination theory is that for a given kind of coordination problem, we should be able to list a variety of alternative coordination processes that could solve the problem. For example, a resource allocation problem can, in principle, be solved by various kinds of markets, hierarchies, and other structures. Realizing this can sometimes help think of new design alternatives that would not otherwise have arisen. We will see below, for instance, how analogies with human markets have generated some novel resource allocation schemes for computer systems (section 4.3.1).

Of course, merely thinking of new schemes is not as useful as also having a way of judging which schemes are likely to be most appropriate in a given situation. Therefore, a fully developed coordination theory should also provide tools for evaluating alternative coordination schemes. For instance, we will see examples below of formally analyzing the tradeoffs in several generic schemes for task assignment (section 3.4), and the stability properties of several task allocation schemes in computer networks (section 4.3.2).

Relation to systems theory and cybernetics. In its attempts to find generalizations that apply across disciplines and across levels of analysis, coordination theory resembles earlier work on systems theory and cybernetics (e.g., Beer, 1967; Boulding, 1956; Emery, 1969; Forrester, 1980; von Bertalanffy, 1950; Wiener, 1961). Coordination theory, however, does not attempt to analyze everything that happens in complex systems. Instead, as we will see below, it focuses only on the goal-directed processes of coordination. Furthermore, we are also significantly better equipped for the task of identifying and analyzing coordination processes now than systems theorists were several decades ago. For example, in addition to the quantitative tools (such as linear systems theory) used in cybernetics, we now have a new set of qualitative languages (such as Petri nets (Peterson, 1981) and object inheritance networks (Stefik and Bobrow, 1986)) that seem promising for formalizing a new class of theories (e.g., Winograd and Flores' (1986) "conversations for action").

What isn't coordination theory? If coordination theory can draw upon so many different fields, is it any more than just the union of these fields? How can we look at a theory and decide whether it is or is not an example of

coordination theory? While it is certainly not helpful to include everything in coordination theory, neither do we think it essential to draw sharp boundaries between what is and is not coordination theory. Instead, as in cognitive science and many other fields, we think certain characteristic questions and approaches will come to typify central examples of coordination theory. For example, theories that apply to only one kind of actor will probably be less important to coordination theory than theories that can be applied to several kinds of actors. In this sense, then, coordination theory will be more like the intersection of the fields from which it draws, rather than the union. By highlighting a novel set of common questions, however, coordination theory may lead to new insights and results that would not have arisen if work in each field had been pursued independently.

Applying coordination theory. Any scientific theory (indeed, any statement about the world) must neglect some things, in order to focus on others. Since coordination theory focuses on principles that apply to many different kinds of systems, it necessarily deemphasizes things that may be extremely important in particular systems. In applying coordination theory to any particular kind of system, therefore, it may be necessary to consider many other factors as well.

For instance, in designing a new computer system to help people coordinate their work, “details” about screen layout and response time may sometimes be as important as the basic functionality of the system, and the reputation of the manager who introduces the system in a particular organization may have more effect on the motivation of people to use it in that organization than any incentive structures designed into the system. Similarly, in designing a distributed processing computer system, the failure rates for different kinds of communications media and processors may be the primary design consideration, overwhelming any other considerations about how tasks are allocated among processors.

There are at least two ways an interdisciplinary theory like coordination theory can deal with differences like these among systems: (1) *parametric analysis*, and (2) *baseline analysis*. In parametric analysis, the abstract theories include parameters which may be different for different kinds of systems. For instance, the principles of aerodynamics apply to both birds and airplanes, even though parameters such as size, weight, and energy expenditure are very different in the two kinds of systems. Similarly, abstract models of coordination may include parameters for things like incentives, cognitive capacities, and communication costs which are very different in human, computational, and biological systems.

In baseline analysis, one theory is used as a baseline for comparison to the actual behavior of a system, and deviations from the baseline are then explained with other theories. For example, in behavioral decision theory (e.g., Kahneman and Tversky, 1973), mathematical decision theory is used to analyze the ways people actually make decisions. In the cases where people depart from the prescriptions of the normative mathematical theory, new theories are developed to explain the differences. Even though the original mathematical theory does not completely explain people’s actual behavior, the anomalies explained by the new theories could not even have been recognized

without a baseline theory for comparison. This suggests that an important part of coordination theory will be “behavioral coordination theory” in which careful observations of actual coordination in human systems are used to develop, test, and augment abstract models of coordination.

2.3. Foundations of coordination theory

Even though use of the term “coordination theory” is quite recent, a great deal of previous work in various fields can contribute to the interdisciplinary understanding of coordination. In this section, we briefly describe examples of such work from several different disciplines. These examples focus on cases where coordination has been analyzed in ways that appear to be generalizable beyond a single discipline or type of actor. We have not, of course, attempted to list all such cases; we have merely tried to pick illustrative examples from several disciplines.

2.3.1. Computer science

Sharing resources. Much research in computer science focuses on how to manage activities that share resources. For example, operating systems require algorithms for allocating resources—such as processors and memory—to different processes and for scheduling accesses to input/output devices, such as disks (e.g., Deitel, 1983).

Other mechanisms have been developed to enforce resource allocations. For example, semaphores, monitors, and critical regions for mutual exclusion are programming constructs that can be used to grant a process exclusive access to a resource (e.g., Dijkstra, 1968; Hoare, 1975). Researchers in database systems have developed numerous other mechanisms, such as locking or timestamping, to allow multiple processes to concurrently access shared data without interference (e.g., Bernstein and Goodman, 1981).

Managing unreliable actors. In addition, protocols have been developed to ensure the reliability of transactions comprising multiple reads or writes on different processors (e.g., Kohler, 1981). In particular, these protocols ensure that either all a transaction’s operations are performed or none are, even if some of the processors fail.

Segmenting and assigning tasks. One of the important problems in allocating work to processors is how to divide up the tasks. For example, Gelernter and Carrero (1989) discuss three alternative ways of dividing parallel programs into units: according to the type of work to be done, according to the final results, or simply according to which processor is available.

Managing information flows. Another important set of issues involves managing the flow of information between activities (e.g., Dennis, 1974). For example, in the producer/consumer problem, the rate of execution of each process must be regulated to ensure that the producer does not overwhelm the consumer or vice versa. One approach to this problem is to place a buffer between the two processes and allocate space in the buffer to one process or the other. Network protocols manage similar problems between communicating processes that do not share any memory (Tannenbaum, 1981).

Other researchers in artificial intelligence and particularly in distributed artificial intelligence (DAI, e.g., Bond and Gasser, 1988; Huhns and Gasser, 1989) have addressed many of these issues. For example, the blackboard architecture allows processes to share information without having to know precisely which other processes need it (Erman, et al., 1980; Nii, 1986), and exchanging partial global plans allows nodes in a distributed sensor network to combine their local actions to achieve global goals (Durfee and Lesser, 1987).

2.3.2. *Economics and operations research*

In a sense, almost all of economics involves the study of coordination, with a special focus on how incentives and information flows affect the allocation of resources among actors. For example, *classical microeconomics* analyzes how different sources of supply and demand can interact locally in a market in ways that result in a globally coherent allocation of resources. Among the major results of this theory are formal proofs that (under appropriate mathematical conditions) if consumers each maximize their individual “utilities” and firms each maximize their individual profits, then the resulting allocation of resources will be globally “optimal” in the sense that no one’s utilities can be increased without decreasing someone else’s (e.g., Debreu, 1959).

Some more recent work in economics has focused on the limitations of markets and contracts for allocating resources. For instance, *transaction cost theory* analyzes the conditions under which a hierarchy is a better way of coordinating multiple actors than a market (e.g., Williamson, 1975). For instance, if extensive legal and accounting work are necessary for market transactions, then the costs of internal transactions within a hierarchical firm may be lower and therefore preferable. *Agency theory* focuses on how to create incentives for some actors (“agents”) to act in a way that advances the interests of other actors (“principals”) even when the principals cannot observe everything their agents are doing (Ross, 1973). One result of this theory is that there are some situations where no incentives can motivate an agent to perform optimally from the principal’s point of view (Jensen and Meckling, 1976).

Finally, some parts of economics focus explicitly on information flows. For example, *team theory* and its descendants analyze how information should be exchanged when multiple actors need to make interdependent decisions but when all agents have the same ultimate goals (e.g., Hurwicz, 1973; Marschak and Radner, 1972; Reiter, 1986). *Mechanism design theory* also analyzes how to provide incentives for actors to reveal information they possess, even when they have conflicting goals. For example, this theory has been applied to designing and analyzing various forms of auctions. In a “second price auction,” for instance, each participant submits a sealed bid, and the highest bidder is only required to pay the amount of the second highest bid. It can be shown that this mechanism motivates the bidders to each reveal the true value they place on the item being sold, rather than trying to “game the system” by bidding only enough to surpass what they expect to be the next highest bid (Myerson, 1981).

Operations research analyzes the properties of various coordination mechanisms, but operations research also includes a special focus on developing optimal techniques for coordination decisions. For instance, operations research

includes analyses of various scheduling and queueing policies and techniques such as linear programming and dynamic programming for making resource allocation decisions optimally (e.g., Dantzig, 1963).

2.3.3. *Organization theory*

Research in organization theory, drawing on disciplines such as sociology and psychology, focuses on how people coordinate their activities in formal organizations. Some of this work deals specifically with the emotions, motivations, and cognitive limitations of people. But another central theme in this work has involved analyzing general issues about coordination (e.g., Galbraith, 1977; Lawrence and Lorsch, 1967; March and Simon, 1958; Simon, 1976; Thompson, 1967; summarized by Mintzberg, 1979, and Malone, 1990). We can loosely paraphrase the key ideas of this work as follows:

All activities that involve more than one actor require (1) some way of dividing activities among the different actors and (2) some way of managing the interdependencies between the different activities (Lawrence and Lorsch, 1967; March and Simon, 1958). Interdependencies between activities can be of (at least) three kinds: (a) *pooled*, where the activities share or produce common resources but are otherwise independent; (b) *sequential*, where some activities depend on the completion of others before beginning, and (c) *reciprocal*, where each activity requires inputs from the other (Thompson, 1967). These different kinds of interdependencies can be managed by a variety of coordination mechanisms, such as: *standardization*, where predetermined rules govern the performance of each activity; *direct supervision*, where one actor manages interdependencies on a case-by-case basis, and *mutual adjustment*, where each actor makes on-going adjustments to manage the interdependencies (Galbraith, 1973; March and Simon, 1958; Mintzberg, 1979).

These coordination mechanisms can be used to manage interdependencies, not only between individual activities, but also between groups of activities. One criterion for grouping activities into units is to minimize the difficulties of managing these inter-group interdependencies. For example, activities with the strongest interdependencies are often grouped into the smallest units, then these units are grouped into larger units with other units with which they have weaker interdependencies. Various combinations of the coordination mechanisms, together with different kinds of grouping, give rise to the different organizational structures common in human organizations, including functional hierarchies, product hierarchies, and matrix organizations. For instance, sometimes all activities of the same type (e.g., manufacturing) might be grouped together in order to take advantage of economies of scale; at other times, all activities for the same product (e.g., marketing, manufacturing, and engineering) might be grouped together to simplify managing the interdependencies between the activities.

2.3.4. *Biology*

Many parts of biology involve studying how different parts of living entities interact. For instance, human physiology can be viewed as a study of how the activities of different parts of a human body are coordinated in order

to keep a person alive and healthy. Other parts of biology involve studying how different living things interact with each other. For instance, ecology can be viewed as the study of how the activities of different plants and animals are coordinated to maintain a “healthy” environment.

Some of the most intriguing studies of biological coordination involve coordination between different animals in a group. For example Mangel (1988) discusses the optimal hunting pack size for lions, who trade the benefit of an increased chance of catching something against the cost of having to share what they catch. Deneubourg (1989) point out that the interaction between simple rules—such as “do what my neighbour is doing”—and the environment may lead to a variety of collective behaviors.

The most striking examples of such group behaviors are in social insects, such as honey bees or army ants, where the group displays often quite complex behavior, despite the simplicity of the individuals (e.g., Franks, 1989; Seeley, 1989). Using a variety of simple rules, these insects “allocate” individual workers at economically efficient levels to a variety of tasks—including searching for new food sources, gathering nectar or pollen from particular sources (bees), carrying individual food items back to the bivouac (ants), guarding the hive (bees) and regulating the group temperature. For example, in honey bees, the interaction of two simple local rules controls the global allocation of food collectors to particular food sources. First, nectar storing bees unload nectar from foraging bees returning to the hive at a rate that depends on the richness of the nectar. Second, if bees are unloaded rapidly, they recruit other bees to their food source. The result of these two rules is that more bees collect food from better sources. Seeley (1989) speculates that this decentralized control may occur because it provides faster responses to local stresses (Miller, 1978), or it may be simply because bees have not evolved any more global means of communication.

3. TOWARD A FRAMEWORK FOR COORDINATION THEORY

So far, we have claimed that many different disciplines can contribute to our understanding of coordination and given examples where this seems possible. But is it really sensible to use the term “coordination” in describing all the different kinds of phenomena to which we have alluded? For that matter, is there anything in common among these different phenomena, other than some occasional similarities in terminology? As a first step toward answering these questions, this section provides a preliminary framework for analyzing coordination. This framework is not a “theory of everything;” it is only one approach that helps identify the relationships between different views of coordination and suggests directions for future progress.

3.1. Components of coordination

According to our broad definition of coordination above, coordination means “the act of working together.” What does this broad definition of coordination imply? First of all, what does the word “work” imply? The dictionary

<i>Components of coordination</i>	<i>Associated coordination processes</i>
Goals	Identifying goals (e.g., goal selection)
Activities	Mapping goals to activities (e.g., goal decomposition)
Actors	Mapping activities to actors (e.g., task assignment)
Interdependencies	“Managing” interdependencies (e.g., resource allocation, sequencing, and synchronizing)

Table 1. Components of coordination.

defines “work” as “physical or mental effort or activity directed toward the production or accomplishment of something” (American Heritage Dictionary, 1981). Thus there must be one or more *actors*, performing some *activities* which are directed toward some ends. In what follows, we will refer to the ends toward which the activities are directed as *goals* or evaluation criteria. By using the word “together,” the definition implies that the activities are not independent. Instead, they must be performed in a way that helps achieve the goals. We will refer to these goal-relevant relationships between the activities as *interdependencies*. These components and the coordination processes associated with them are summarized in Table 1. (See Baligh, 1986; Baligh and Burton, 1981; Barnard, 1964; Malone, 1987; Malone and Smith, 1988; McGrath, 1984; Mintzberg, 1979 for related decompositions of coordination.)

We believe that all four of these components—goals, activities, actors and interdependencies—are necessary for a situation to be analyzed in terms of coordination. That is, in order to sensibly use the word “coordination” to refer to a situation you must, at least implicitly, have in mind all four components. For instance, it does not make sense to refer to a system as being coordinated if there are no activities performed or if the activities are completely independent. Furthermore, activities can only be “coordinated” with respect to some goals or evaluation criteria.

For example, an automobile manufacturing company might be thought of as having a set of goals (producing several different lines of automobiles) and a set of actors (people) who perform activities (designing and assembling cars) that achieve these goals. These activities may have various kinds of interdependencies such as using the same resources (a station on an assembly line) or needing to be done in a certain order (a car must usually be designed before it is built). As another example, a computer network can be thought of as having a set of goals (computations to be performed) and a set of actors (computer processors of various types) that perform the tasks necessary to achieve these goals, subject to various interdependencies (such as using the same data).

Coordination is attributed to a situation by observers. It is important to realize that the actors involved in a situation may or may not all agree on the identification of all these components. Instead, one or more of these components may be attributed by an observer in order to analyze the situation in terms of coordination. For instance, we may sometimes analyze everything that happens in a manufacturing division as one “activity,” while at other times, we may want to analyze each station on an assembly line as a separate “activity.” Sometimes, we might even regard different processes in the brain of a single person as separate activities performed by separate actors (e.g., Minsky, 1987).

One important case of this attribution occurs in attributing goals to actors. For instance, it is often useful to analyze biological systems in terms of the goal of survival and to assume that the process of natural selection will result in an optimal set of traits (Stephens and Krebs, 1986).

In analyzing human actors, it is often useful to simply ask people what their goals are and analyze their behavior in terms of these stated goals. In some cases, however, people may be unable or unwilling to accurately report their goals. And even if people think they are accurately reporting their goals, it may also be useful to analyze their behavior with respect to other evaluation criteria. For instance, the behavior of managers is often better explained by assuming that they try to maximize their own power and status (e.g., by increasing revenues) rather than trying to maximize the profits of the shareholders in their firm. One form of this view is what Dennet (1987) calls the “intentional stance:” since there is no completely reliable way to determine someone’s goals (or if, indeed they have goals at all), we, as observers, can only impute goals to the actors. The key issue, then, is how useful the imputed goals are in understanding the behavior of the system.

In cases where individual actors in a system have *conflicting goals*, it is often useful to analyze the behavior of the system in terms of some overall goals (or evaluation criteria) in which we are interested. For instance, even though two designers on a project team may have strongly opposing views about how a product should be designed, we can evaluate their collective behavior in terms of the quality of the final design. In human organizations, situations in which actors have at least partly conflicting goals are nearly universal, and mixtures of cooperation and conflict are quite common (e.g., Ciborra, 1987; Schelling, 1960; Williamson, 1985). When we analyze the coordination in these situations, we must (at least implicitly) evaluate the actors’ collective behavior in terms of how well it achieves some overall goals (which may or may not be held by the actors themselves).

Another important example of conflicting goals occurs in market transactions: All participants in a market might have the goal of maximizing their own benefits, but we, as observers, can evaluate the market as a coordination mechanism in terms of how well it achieves global goals such as maximizing consumer utilities (e.g., Debreu, 1959) or “fairly” distributing economic resources.

A narrow definition of coordination. The decomposition of coordination into components also suggests another definition for coordination. The broad definition of coordination (discussed in section 2.1) includes almost

everything that happens when actors work together: setting goals, selecting actors, and performing all the other activities that need to be done. For some purposes, it is useful to be able to focus explicitly on the elements that are unique to coordination, that is, on the aspects of “working together” that are not simply part of “working.” In our analysis of the broad definition above, the element of coordination that was implied by the word “together” was interdependencies. Therefore, when we want to focus specifically on the aspects of a situation that are unique to coordination, we will use the following narrow definition:

Coordination is the act of managing interdependencies between activities.

This emphasis on the importance of interdependence is consistent with a long history in organization theory of emphasizing the same theme (e.g., Galbraith, 1973; Hart and Estrin, 1990; Lawrence and Lorsch, 1967; Pfeffer, 1978; Rockart and Short, 1989; Thompson, 1967). It is also consistent with the simple intuition that, if there is no interdependence, there is nothing to coordinate.

Are multiple actors necessary for coordination to occur? In our previous work (e.g., Malone, 1988), we defined coordination as something that occurs only when multiple actors are involved. Since then, however, we have become convinced that the essential elements of coordination listed above arise whenever multiple, interdependent activities are performed to achieve goals—even if only one actor performs all of them. For instance, if a single person is cooking a meal, that person must coordinate the preparation of multiple dishes, including resource conflicts, timing interdependencies, and so forth.

3.2. Basic coordination processes

A central concern for coordination theory is understanding the basic processes involved in coordination: Are there fundamental coordination processes that occur in all coordinated systems? How can we represent and analyze these processes? Is it possible to characterize situations in a way that helps generate and choose appropriate coordination mechanisms for them?

Taxonomies for coordination processes. One question that arises immediately is how to distinguish between different coordination processes. For instance, one taxonomy is suggested in Table 1: Each component of coordination (goals, activities, actors, and interdependencies) is associated with processes that determine or manage it. Crowston (1991) suggests a more structured taxonomy based on the possible relationships between “tasks” and “objects.” The further development of such taxonomies is an important goal for coordination theory. For our purposes here, however, we simply describe several examples of pervasive coordination processes without claiming that all these processes are necessary in every case or that the categories in this list are mutually exclusive or exhaustive.

3.2.1. *Goal selection and goal decomposition*

Top-down goal decomposition. As we noted above, the definition of coordination implies multiple activities related to some goal(s). Therefore, in order for coordination to occur, these goals and activities must somehow be identified. The most commonly analyzed case of this process occurs when an individual or group decides to pursue a goal, and then decomposes this goal into activities (or subgoals) which together will achieve the original goal. In this case, we call the process of choosing the goal *goal selection*, and the process of choosing the activities *goal decomposition*.

For example, the strategic planning process in human organizations is often viewed as involving this kind of goal selection and goal decomposition process. Furthermore, an important role for all managers in a traditionally conceived hierarchy is to decompose the goals they are given into tasks that they can, in turn, delegate to people who work for them. There are, in general, many ways a given goal can be broken into pieces, and a long-standing topic in organization theory involves analyzing different possible decompositions such as by function, by product, by customer and by geographical region (Mintzberg, 1979). Some of these different goal decompositions for human organizations are analogous to ways computer systems can be structured (e.g., Malone and Smith, 1988).

In computer systems, we usually think of the goals as being predetermined, but an important problem involves how to break these goals into activities that can be performed separately. In a sense, for example, the essence of all computer programming is decomposing goals into elementary activities. For instance, programming techniques such as subroutine calls, modular programming, object oriented programming, and so forth can all be thought of as techniques for structuring the process of goal decomposition (Liskov and Guttag, 1986). In these cases the goal decomposition is performed by a human programmer. Another example of goal decomposition in computer systems is provided by work on planning in artificial intelligence (e.g., Allen, et al.; Chapman, 1987; Fikes and Nilsson, 1971). In this case, goals are decomposed by a planning program into a sequence of elementary activities, based on knowledge of the elementary activities available, their prerequisites, and their effects.

In some cases, techniques for goal decomposition used in computer systems may suggest new ways of structuring human organizations. For example, Moses (1990) suggests that human organizations might sometimes be better off not as strict hierarchies but as multi-layered structures in which any actor at one level could direct the activities of any actor at the next level down. This multi-layered structure is analogous to successive layers of languages or "virtual machines" in a computer system (see Malone, 1990).

Bottom-up goal identification. Even though the most commonly analyzed cases of coordination involve a sequential process of goal selection and then goal decomposition, our definition of coordination does not require that things happen in this order. Another possibility, for instance, is that several actors realize that the things they are already doing (with small additions) could work together to achieve a new goal. For example, the creation of a new interdisciplinary research group may have this character. It is also possible, as we noted above, for an external

observer to analyze the behavior of a group of actors in terms of a “goal” that none of the individual actors actually has. In order to include these different possibilities for who determines goals and activities and the order in which they are determined, we intentionally used the general terms “identifying goals” and “mapping goals to activities” in Table 1.

3.2.2. *Resource allocation (managing resource interdependencies)*

Whenever multiple activities share some limited resource (e.g., money, storage space, or an actor’s time), then some resource allocation process is needed to manage the interdependencies among these activities. Resource allocation is perhaps the most widely studied of all coordination processes. For example, much of economics and operations research is devoted to studying various resource allocation processes. As we saw above, resource allocation issues also arise in computer processor scheduling and memory management, and substantial work has been done on both these topics (e.g., Cytron, 1987; Halstead, 1985). Organization theory has investigated the extensive effects resource dependencies often have on entire organizational structures (e.g., Pfeffer and Salancik, 1978) and has studied various hierarchical resource allocation processes (e.g., Burton and Obel, 1980a; Burton and Obel, 1980b). From the point of view of individual human organizations, a key aspect of the resource allocation problem involves attracting resources (e.g., employees and money) away from other possible activities (Barnard, 1964).

One important special case of resource allocation is task assignment, that is, allocating the scarce time of actors to the tasks they will perform. Part of this process can be viewed as *selecting actors* to perform tasks based on their capabilities, other possible tasks they could do, and so forth (see section 3.4). Another part of this process involves establishing agreements between task requestors and task performers about who will perform the task, what the task consists of, and whether it has been satisfactorily completed. Winograd and Flores (1986) analyze this *mutual agreeing* part of the process using concepts from linguistics and computer science (see section 4.2.1).

3.2.3. *Sequencing and synchronizing (managing timing interdependencies)*

Another pervasive coordination problem involves getting a group of activities to occur at the right times with respect to each other. Sometimes one activity must finish before others can begin. We call managing this kind of interdependency *sequencing*. In other cases, several activities must all occur at the same time. We call managing this kind of interdependency *synchronizing*. Sequencing and synchronizing are both common problems in human organizations, and a number of techniques have been developed to deal with them. For instance, techniques from operations research, such as PERT charts and critical path methods, are designed to help manage large projects with multiple activities and complex prerequisite structures. Sequencing and synchronizing problems arise frequently in computer systems, as well. For instance, one of the key issues in taking advantage of parallel processing computers is determining which activities can be done in parallel and which ones must wait for the completion of others (Arvind and Culler, 1986; Holt, 1988; Peterson, 1977; Peterson, 1981). Some of these ideas from computer science

have also been used to help streamline processes in human organizations by taking advantage of their latent parallelism (e.g., Ellis, et al., 1979).

In practice, timing interdependencies often arise in the context of scarce resources, and many of the formal techniques for managing timing interdependencies also involve resource allocation. For example, scheduling techniques developed in operations research often allocate scarce resources to the most important activities while simultaneously satisfying prerequisite constraints (Conway, et al., 1967). As another example, the *mutual exclusion* or *synchronization* problem in computer science involves preventing two different processes from simultaneously accessing the same resource (e.g., Dijkstra, 1968). For example, if two processes simultaneously read and then add to the same value in a database, the results are unpredictable. It is clear that this problem has important timing aspects. However, it can also be viewed as a resource allocation problem since the data being updated is a kind of resource that can only be used by one process at a time.

3.2.4. *Managing other interdependencies*

The three kinds of interdependencies discussed above are quite common, as reflected by the large number of techniques developed for managing them. It seems clear, however, that there are many other kinds of possible interdependencies.

Generic interdependencies. Some of these other interdependencies are generic. For example, one possible relationship between two tasks is that they *duplicate* each other. In this case, it may be possible to merge the two tasks or eliminate one of them. For example, if two engineers are unintentionally designing different parts that do the same thing, effort might be saved by eliminating one of the parts. In many situations, it is also important to manage *interdependencies between different objects*. For example, mechanical engineers must manage interdependencies between parts that touch each other or that interact physically in other ways.

Specific interdependencies. Other kinds of interdependencies are specific to the particular actions involved. For instance, people from different parts of a company who deal with the same customer are interdependent to the degree that they each affect the customer's perception of the company. As another example, the actions of designing and manufacturing a part are interdependent in that the manufacturing process constrains how the part can be designed. In this case, for instance, the designer and the manufacturer can manage the interdependence by negotiating the design in joint meetings. Alternatively, some of the knowledge about the constraints of one task can be moved to the other engineer. For example, some of the manufacturer's knowledge about manufacturing constraints can be made available to the designer by training the designer in methodologies such as Design For Manufacturing (DFM).

A catalogue of interdependencies and the processes for managing them? One intriguing possibility here is that it may be possible to develop a "catalogue" or "engineering handbook" of common kinds interdependencies (such as those just described) and various processes for managing them. For example, such a handbook might include a

<i>Process Level</i>	<i>Components</i>	<i>Examples of Generic Processes</i>
Coordination	goals, activities, actors, interdependencies	identifying goals, ordering activities, assigning activities to actors, allocating resources, synchronizing activities
Group decision-making	goals, actors, alternatives, evaluations, choices	proposing alternatives, evaluating alternatives, making choices (e.g., by authority, consensus, or voting)
Communication	senders, receivers, messages, languages	establishing common languages, selecting receiver (routing), transporting message (delivering)
Perception of common objects	actors, objects	seeing same physical objects, accessing shared databases

Table 2. Processes underlying coordination.

number of common alternative processes for managing the interdependencies involved in designing parts to be easily manufactured (e.g., see Crowston, 1991). One of our hopes is that identifying these additional interdependencies may also lead to the development of new formal techniques for managing some of them analogous to the formal techniques that have already been developed for resource allocation and timing.

3.3. Processes underlying coordination

In attempting to characterize more precisely different coordination processes, we have found it useful to describe them in terms of successively deeper levels of underlying processes, each of which depends on the levels below it. Table 2 shows a diagram of these levels. For instance, most of the coordination processes listed above in the last column of Table 1 require that some decision be accepted by a group (e.g., what goal will be selected or which actors will perform which activities). Group decisions, in turn, often require members of the group to communicate in some form about the goals to be achieved, the alternatives being considered, the evaluations of these alternatives, and the choices that are made. This communication requires that some form of "messages" be transported from senders to receivers in a language that is understandable to both. Finally, the establishment of this common language and the transportation of messages depends, ultimately, on the ability of actors to perceive common objects such as physical objects in a shared situation or shared information in a computer database. These layers are analogous to abstraction levels in other systems, such as protocol layers for network communications.

The order of these layers reflects the observation that each layer uses processes from the layers below it. For instance, group decision-making requires that the members be able to communicate in some way. However, unlike most network protocols, there are also times when a layer uses processes from the layers *above* it. For instance, a group may sometimes use decision-making processes to extend the common language it uses to communicate (e.g. see Lee and Malone, 1990), or a group may use coordination processes to assign decision-making activities to actors.

3.3.1. *Group decision-making*

By group decision making, we mean the process by which a group of actors choose a consistent set of alternatives. For example, in a task assignment process, actors must agree on an assignment of actors to tasks, possibly subject to constraints (e.g., all tasks must be assigned to a single actor capable of doing the task; the load on actors should be balanced). Obvious possible assignment processes include authority (i.e., the “boss” decides and tells everyone), consensus (all actors agree on an assignment) or voting. Note that this model of group decision-making does not require that all actors agree on a single “best” alternative or even be aware that a decision was made. For example, in a decentralized organization, an actor may independently elect to perform a task, making it unnecessary for other actors to even consider doing it.

There is already a large body of literature on decision theory (e.g., Kahneman and Tversky, 1973; Luce and Raiffa, 1957) and problem solving (e.g., Newell and Simon, 1972; Nilsson, 1971) that applies primarily to *individual* decision making. Some work also focuses on problems of *group* decision-making (e.g., Simon, 1976). For instance, the “Arrow impossibility theorem” (Arrow, 1951) examines the difficulty of reconciling the preferences of different individuals using any of these methods. It shows that there is no non-dictatorial way to always reconcile individual preferences without violating other sensible conditions such as that introducing irrelevant alternatives should not change the group’s choice.

An intriguing question suggested by this framework is how the existing work on decision-making should be augmented to include group decision-making. For instance, are there decision-making biases in groups (e.g., Janis and Mann, 1977) that are analogous to the individual decision-making biases described by Kahneman & Tversky (1973)? How do computer-based group decision-making tools affect these processes (e.g., Dennis, et al., 1988; Kiesler, et al., 1984; Kraemer and King, 1988)? Can we determine optimal ways of allocating tasks and sharing information for making group decisions (Miao, et al., in press).

3.3.2. *Communication*

As with decision-making, there is already a great deal of theory about communication, both from a technical point of view (e.g., Shannon and Weaver, 1949) and from an organizational point of view (e.g., Allen, 1977; Rogers and Agarwala-Rogers, 1976; Weick, 1969). However, a coordination framework highlights several somewhat novel aspects to these problems. For one thing, it emphasizes questions about why the communication is occurring in the first place (e.g., to aid decision-making or to allocate resources). For another thing, it suggests the importance of several questions that have not yet been heavily studied.

For instance: How can actors establish common languages that allow them to communicate in the first place? This question of developing standards for communication is of crucial concern in designing computer networks in general (Dertouzos, 1991) and cooperative work tools in particular (e.g., Lee and Malone, 1988). The process by which

standards are developed is also of concern to economists, philosophers, and others (e.g., Farrell and Saloner, 1985; Hirsch, 1987).

Another set of questions highlighted by our framework is: How can information be routed to the right actors (e.g., Allen, 1977; Hewitt, 1977; Malone, et al., 1987)? More specifically, what do we have to know about actors' goals and task assignments to determine which actors should know which kinds of information (e.g., Marschak and Radner, 1972)?

3.3.3. Perception of common objects

In a sense, the ability of actors to communicate or work together at all depends ultimately on their ability to perceive the same things. For instance, Lynch and Fischer (1981) base their analysis of distributed computing on the premise that different processes have some shared variables. Similarly, there is growing literature in computer science, economics, and linguistics on how distributed actors can come to have "common knowledge" of a fact (Aumann, 1976; Halpern, 1987; Milgrom, 1981). For instance, in the "coordinated attack" problem, two generals are overlooking a common enemy from opposite hilltops and can communicate only via messengers who may be delayed or captured by the enemy. If the generals attack together, they will defeat the enemy; if either attacks alone, he will be defeated. Surprisingly, it can be shown that if the messengers are unreliable, the generals can never be sure they will both attack at the same time (Gray, 1978; Halpern and Moses, 1986).

At a higher level, computer-based communication among people ultimately depends on either some kind of shared database or some kind of messages which are transported from one person to another. At a different level, Suchman (1987), Winograd and Flores (1986), and others observe that all human action is grounded in shared situations. Thus ambiguities can be resolved and "breakdowns" can be "repaired" by appealing to the always present—but usually implicit—features of these shared situations.

3.4. Example: Analyzing the task assignment process

To illustrate the possibilities for developing theories of coordination that apply to many different kinds of systems, let us consider the task assignment process as analyzed by Malone and Smith (Malone, 1987; Malone and Smith, 1988; see also related work by Baligh and Richartz, 1967; Burton and Obel, 1980a). As we have described in more detail elsewhere (Malone, in press), these analyses illustrate precisely the kind of interdisciplinary interaction that coordination theory encourages: the models grew originally out of designing distributed computer systems, they drew upon results from operations research, and they led eventually to new insights about the evolution of human organizations.

3.4.1. *A generic task assignment problem*

Consider the following task assignment problem: A system is producing a set of “products,” each of which requires a set of “tasks” to be performed. The tasks are of various types, and each type of task can only be performed by “server” actors specialized for that kind of task. Furthermore, the specific tasks to be performed cannot be predicted in advance; they only become known during the course of the process and then only to actors we will call “clients.” This description of the task assignment problem is certainly not universally applicable, but it is an abstract description that can be applied to many common task assignment situations. For instance, the tasks might be (1) designing, manufacturing, and marketing different kinds of automobiles, or (2) processing steps in different jobs on a computer network.

As suggested by Table 2, the *coordination* problem in this situation involves a *group decision* about which actors will perform which tasks. In order to make this decision rationally, two kinds of information must be considered: (1) information about what tasks need to be done, and (2) information about which actors are qualified and available to do the tasks. Furthermore, once the decision has been made, the server actors have to be notified what tasks they will perform. The coordination mechanisms we will consider for solving this task assignment problem involve various ways of *communicating* this information (e.g., via messages), and in order for this communication to occur, both the senders and receivers must share some representations of *common objects* (such as the descriptions of tasks and actors’ capabilities).

3.4.2. *Possible coordination mechanisms*

One (highly centralized) possibility for solving this task assignment problem is for all the clients and servers to send all their information to a central decision maker who decides which servers will perform which tasks and then notifies them accordingly. Another (highly decentralized) possibility is suggested by the competitive bidding scheme for computer networks formalized by Smith and Davis (1981). In this scheme, a client first broadcasts an *announcement* message to all potential servers. This message includes a description of the activity to be performed and the qualifications required. The potential servers then use this information to decide whether to submit a *bid* on the action. If they decide to bid, their bid message includes a description of their qualifications and their availability for performing the action. The client uses these bid messages to decide which server should perform the activity and then sends an *award* message to notify the server that is selected.

Malone and Smith (Malone, 1987; Malone and Smith, 1988) analyzed several alternative coordination mechanisms like these, each of which is analogous to a mechanism used in human organizations. In particular, they developed formal models to represent various forms of markets (centralized and decentralized) and various forms of hierarchies (based on products or functions). Then they used techniques from queueing theory and probability theory to analyze tradeoffs among these structures in terms of production costs, coordination costs, and vulnerability costs. For instance, they showed that the centralized schemes had lower coordination costs, but were more vulnerable to

processor failures. Decentralized markets, on the other hand, were much less vulnerable to processor failures but had high coordination costs. And decentralized hierarchies (“product hierarchies”) had low coordination costs, but they had unused processor capacity which lead to high production costs.

3.4.3. Applying these models to various kinds of systems

Even though these models omit many important aspects of human organizations and computer systems they help illuminate a surprisingly wide range of phenomena. For instance, as Malone and Smith (1988) show, the models are consistent with a number of previous theories about human organizational design (e.g., Galbraith, 1973; March and Simon, 1958; Williamson, 1985) and with major historical changes in the organizational forms of both human organizations (Chandler, 1962; Chandler, 1977) and computer systems. These models also help analyze design alternatives for distributed scheduling mechanisms in computer systems, and they suggest ways of analyzing the structural changes associated with introducing new information technology into organizations (section 4.1 of this paper; Crowston, et al., 1987; Malone and Smith, 1988).

3.5. Summary of coordination processes

Table 3 loosely summarizes our discussion of coordination processes by listing examples of analyses of coordination processes in different disciplines. The key point of this table, and indeed of much of our discussion, is that the concepts of coordination theory help identify similarities among concepts and results in different disciplines. These similarities, in turn, suggest how ideas can be transported back and forth across disciplinary boundaries and where opportunities exist to develop even deeper analyses.

4. APPLICATIONS OF COORDINATION THEORY

We describe in this section examples of how concepts about coordination have been applied in three different areas: (1) understanding the new possibilities for human organizations and markets provided by information technology, (2) designing cooperative work tools, and (3) designing distributed and parallel computer systems. This list is not intended to be a comprehensive list of all coordination theory applications. In fact, most of the work we describe here did not explicitly use the term “coordination theory.” We have chosen examples, however, to illustrate the wide range of applications for interdisciplinary theories about coordination.

4.1. Understanding the effects of information technology on organizations and markets

Managers, organization theorists, and others have long been interested in how the widespread use of information technology (IT) may change the ways human organizations and markets will be structured (e.g., Leavitt and Whisler,

<i>Level</i>	<i>Process</i>	<i>Computer Science</i>	<i>Economics and Operations Research</i>	<i>Organization Theory</i>
Coordination	Goal decomposition	modularization techniques in programming; planning in artificial intelligence	economies of scale and scope	strategic planning; management by objectives; methods of grouping people into units
	Resource allocation (including task assignment)	techniques for processor scheduling and memory allocation	analyses of markets and other resource allocation mechanisms; scheduling algorithms and other optimization techniques	analyses of different organizational structures; budgeting processes, organizational power, and resource dependence
	Sequencing and synchronizing	synchronization techniques; data flow and Petri net analyses	PERT charts, critical path methods; scheduling techniques	
Group decision-making	Proposing, evaluating, and selecting alternatives	search techniques in artificial intelligence	decision theory and other optimization techniques	behavioral decision theory; analyses of organizational decision-making
Communication	Establishing common languages	communication standards; polymorphism in object-oriented programming	analyses of standards formation processes	studies of shared languages and culture
	Routing	message passing languages; analyzing data flows; network name servers	analyses of the value of information	studies of organizational communication patterns
Perception of common objects	Seeing and manipulating shared information	shared variables; common knowledge; shared databases	theories of common knowledge	theories of situated action

Table 3. Examples of previous analyses of coordination processes in different disciplines

1958; Simon, 1976). One of the most important contributions of coordination theory may be to help understand these possibilities better.

To illustrate this approach, we begin with the simple observation that coordination is itself an activity that has costs and benefits.¹ Even though there are many other forces that may affect the way coordination is performed in organizations and markets (e.g., global competition, government regulation, interest rates), one important factor is

¹ See Malone (in press) for another version of this discussion.

clearly its cost. Furthermore, it seems quite plausible to assume that information technology is likely to significantly reduce the costs of certain kinds of coordination (e.g., Crawford, 1982).

Now, using some elementary ideas from microeconomics about substitution and elasticity of demand, we can make some simple predictions about the possible effects of reducing coordination costs. It is useful to illustrate these effects by analogy with similar changes in the costs of transportation induced by the introduction of trains and automobiles:

- (1) A “first order” effect of reducing transportation costs with trains and automobiles was simply some substitution of the new transportation technologies for the old: people began to ride on trains more and in horse-drawn carriages less.
- (2) A “second order” effect of reducing transportation costs was to increase the amount of transportation used: people began to travel more when this could be done more cheaply and conveniently in trains than on foot.
- (3) Finally, a “third order” effect was to allow the creation of more “transportation-intensive” structures: people eventually began to live in distant suburbs and use shopping malls—both examples of new structures that depended on the widespread availability of cheap and convenient transportation.

Similarly, we can expect several effects from using new information technologies to reduce the costs of coordination:

- (1) A “first order” effect of reducing coordination costs with information technology may be to substitute information technology for some human coordination. For instance, it has long been commonplace to predict that computers will lead to the demise of middle management because the communication tasks performed by middle managers could be performed less expensively by computers (e.g., Leavitt and Whisler, 1958). In fact, it is still not clear that this prediction has been fulfilled, perhaps, in part, because of the following second and third order effects.
- (2) A “second order” effect of reducing coordination costs may be to increase the overall amount of coordination used. In some cases, this may overwhelm the first order effect. For instance, in one case we studied, a computer conferencing system was used to help remove a layer of middle managers (see Crowston, et al., 1987). Several years later, however, almost the same number of new positions (for different people at the same grade level) had been created for staff specialists in the corporate staff group, many of whom were helping to develop new computer systems. One interpretation of this outcome is that the managerial resources no longer needed for simple communication tasks could now be applied to more complex analysis tasks that would not previously have been undertaken.

- (3) A “third order” effect of reducing coordination costs may be to encourage a shift toward the use of more “coordination-intensive” structures. In other words, coordination structures that were previously too “expensive” will now become more feasible and desirable. What might these new coordination-intensive structures be like? Let us consider recent work on three specific questions about these new structures: (a) How will IT affect the size of organizations? (b) How will IT affect the degree of centralization of decision-making in organizations? and (c) How will IT affect the internal structure of organizations?

4.1.1. *Firm size*

Malone, Yates, and Benjamin (1987) have used ideas from transaction cost theory to systematically analyze how information technology will affect firm size and, more generally, the use of markets as a coordination structure. They conclude that by reducing the costs of coordination, information technology may lead to an overall shift toward smaller firms and proportionately more use of markets—rather than internal decisions within firms—to coordinate economic activity.

This argument has two parts. First, since market transactions often have higher coordination costs than internal coordination (Malone, et al., 1987; Williamson, 1985), an overall reduction in the “unit costs” of coordination should lead to markets becoming more desirable in situations where internal transactions were previously favored. This, in turn, should lead to less vertical integration and smaller firms.

For example, after the introduction of computerized airline reservation systems, the proportion of reservations made through travel agents (rather than by calling the airline directly) went from 35% to 70%. Thus, the function of selling reservations was “disintegrated” from the airlines and moved to a separate firm—the travel agents. Preliminary econometric analyses of the overall U.S. economy in the period 1975-1985 are also consistent with these predictions: the use of information technology appears to be correlated with decreases in both firm size and vertical integration (Brynjolfsson, et al., 1989).

4.1.2. *Centralization of decision-making*

Gurbaxani & Whang (1991) have used ideas from agency theory to systematically analyze the effects on centralization of the reductions in coordination costs enabled by IT. They conclude that IT can lead to either centralization or decentralization, depending on how it is used. While this conclusion may not be surprising, the structure of their analysis helps us understand the factors involved more clearly: (1) When IT primarily reduces *decision information costs*, it leads to more centralization. For instance, the Otis elevator company used IT to centralize the reporting and dispatching functions of their customer service system, instead of having these functions distributed to numerous remote field offices (Stoddard, 1986). (2) On the other hand, when IT primarily reduces *agency costs*, it leads to more decentralization. As used here, agency costs are the costs of employees not acting in

the interests of the firm. For instance, when one insurance company developed a system that more effectively monitored their salespeople's overall performance, they were able to decentralize to the salespeople many of the decisions that had previously been made centrally (Bruns and McFarlan, 1987). Overall, this bidirectional trend for IT and centralization is consistent with empirical studies of this question (Attewell and Rule, 1984).

4.1.3. *Internal structure of organizations*

In addition to the changes already noted, we should also expect the internal structures of organizations to become more coordination-intensive. For example, Milgrom and Roberts (1990) show mathematically how increases in computer-supported coordination are consistent with a large cluster of other organizational attributes such as: closer integration between process and product engineering, extensive use of external suppliers, shorter product cycle times, and very low inventories.

As another example, Malone and Smith (1988) suggest that information technology can facilitate what some observers (e.g., Mintzberg, 1979; Toffler, 1970) have called *adhocracies*. As noted above, adhocracies are very flexible organizations, including many shifting project teams and highly decentralized networks of communication among relatively autonomous entrepreneurial groups. One of the disadvantages of adhocracies as a coordination structure is that they require large amounts of unplanned communication and coordination throughout an organization. However, technologies such as electronic mail and computer conferencing can help reduce the costs of this communication, and advanced information sharing tools (e.g., Lotus, 1989; Malone, et al., 1987) may help make this communication more effective at much larger scales.

Imagining new kinds of human organizations. As these last examples suggests, one of the most important questions that coordination theory may help answer is what new kinds of coordination structures will emerge in the electronically connected world of the near future. What are the organizational equivalents of suburbs and shopping malls that information technology may make possible?

An intriguing possibility is that we may be able to use analogies with coordination structures in other kinds of systems to generate ideas for new kinds of human organizations. For example, would coordination structures like those used in bee hives or ant colonies be useful for certain kinds of human organizations? Might it be possible, for instance, to create knowledge structures that many people could "swarm" around, filling in things they know, until a complete picture of what to do emerges? Or could there be marketplaces of "intellectual mercenaries" in which it is possible to electronically assemble "overnight armies" of thousands of people who work for a few hours or days to solve a particular problem? Generating and understanding these new possibilities will require precisely the kind of theories that coordination theory should be able to help us develop.

4.2. Designing cooperative work tools

There has recently been a great deal of interest in designing computer tools to help people work together more effectively (e.g., Ellis, et al., 1991; Greif, 1988; Johansen, 1988; Peterson, 1986; Tatar, 1988; Tatar, 1990; additional references in Table 4). Using terms such as “computer-supported cooperative work” and “groupware” these systems perform functions such as helping people collaborate on writing the same document, managing projects, keeping track of tasks, and finding, sorting, and prioritizing electronic messages. Other systems in this category help people display and manipulate information more effectively in face-to-face meetings and represent and share the rationales for group decisions.

In this section, we will describe how ideas about coordination have been helpful in suggesting new systems, classifying systems, and analyzing how these systems are used.

4.2.1. *Using coordination concepts from other disciplines to suggest design ideas*

One way of generating new design ideas for cooperative work tools is to look to other disciplines that deal with coordination. For instance, even though the following authors did not explicitly use the term “coordination theory,” they each used ideas about coordination from other disciplines to help develop cooperative work tools.

Using ideas from linguistics and philosophy about speech acts. Winograd and Flores (Flores, et al., 1988; Winograd, 1987; Winograd and Flores, 1986) have developed a theoretical perspective for analyzing group action based heavily on ideas from linguistics (e.g., Searle, 1975). This perspective emphasizes different kinds of speech acts, such as requests and commitments. For example, Winograd and Flores analyzed a generic “conversation for action” in terms of the possible states and transitions involved when one actor performs a task at the request of another. An actor may respond to a request, for instance, by (1) promising to fulfill the request, (2) declining the request, (3) reporting that the request has already been completed, or (4) simply acknowledging that the request has been received. The analysis of this conversation type (and several others) provided a primary basis for designing the Coordinator, a computer-based cooperative work tool. For example, the Coordinator helps people make and keep track of requests and commitments to each other. It thus supports what we might call the “mutual agreeing” part of the task assignment process.

Using ideas from artificial intelligence and organization theory about blackboards and adhocracies. Malone (1990) describes how ideas from artificial intelligence and organization theory combined to suggest a new tool for routing information within organizations. In the “blackboard architecture,” program modules interact by searching a global blackboard for their inputs and posting their outputs on the same blackboard (Erman, et al., 1980; Nii, 1986). This provides very flexible patterns of communication between different program modules: any module can communicate with any other module, even when this interaction is not explicitly anticipated by the program designer. In adhocracies, as we saw above, just this kind of unplanned, highly decentralized communication is essential for

rapidly responding to new situations (Mintzberg, 1979; Toffler, 1970). Stimulated, in part, by this need for an “organizational blackboard,” Malone and colleagues designed the Information Lens system (Malone, et al., 1987). A central component of this system is an “anyone server” that lets people specify rules about what kinds of electronic messages they are interested in seeing. The system then uses these rules to route all non-private electronic messages to everyone in the organization who might want to see them. (To help people deal with large numbers of messages, another part of the system uses a different set of rules to sort and prioritize the messages people receive.)

Using ideas from philosophy and rhetoric about decision-making. Two cooperative work tools, gIBIS (Conklin and Begeman, 1988) and Sibyl (Lee, 1990) are designed to help groups of people make decisions more effectively. To do this, they explicitly represent the arguments (and counter arguments) for different alternatives a group might choose. Both these systems are based on ideas from philosophy and rhetoric about the logical structure of decision-making. For example, the basic elements in the gIBIS system (issues, positions, and arguments) are taken from a philosophical analysis of argumentation by Rittel (1970). The constructs for representing arguments in Sibyl are based on the work of philosophers like Toulmin (1958) and Rescher (1977).

Using ideas from computer science about parallel processes. Holt (Holt, 1988) describes a theoretical language used for designing coordination tools that is based, in part, on ideas about Petri nets, a formalism used in computer science to represent process flows in distributed or parallel systems (Peterson, 1977; Peterson, 1981). This language is part of a larger theoretical framework called “coordination mechanics” and has been used to design a “coordination environment” to help people work together on computer networks.

Summary of examples. Clearly, drawing a line around these examples and calling them “coordination theory” does not, in itself, provide any benefit. Nor does using ideas about coordination from other disciplines provide any guarantee of developing useful cooperative work tools. Nevertheless, we feel that considering these examples within the common framework of coordination theory provides two benefits: (1) it suggests that no one of these perspectives is the complete story, and (2) it suggests that we should look to previous work in various disciplines for more insights about coordination that could lead to new cooperative work tools.

4.2.2. *A taxonomy of cooperative work tools*

As shown in Table 4, the framework we have suggested for coordination provides a natural way of classifying existing cooperative work systems according to the coordination processes they support. Some of these systems primarily emphasize a single coordination-related process. For instance, electronic mail systems primarily support the message transport process, and meeting scheduling tools primarily support the synchronization process (i.e., arranging for several people to attend a meeting at the same time). There is a sense, of course, in which each of these systems also support other processes (e.g., a simple electronic mail system can be used to assign tasks), but we have categorized the systems here according to the processes they explicitly emphasize.

Many of the systems explicitly support several processes. For example, the gIBIS and Sibyl systems explicitly support at least two group decision-making processes: proposing alternatives and evaluating alternatives. The Information Lens system supports both the communication routing process (by rules that distribute messages to interested people) and a form of resource allocation process (by helping people prioritize their own activities using rules that sort messages they receive). And the Polymer system helps people decompose goals into tasks and sequence the tasks (e.g., to prepare a monthly report, first gather the project reports and then write a summary paragraph).

One possibility raised by this framework is that it might help identify new opportunities for cooperative work tools. For instance, the Coordinator focuses on supporting one part of the task assignment process (mutual agreement on commitments). However, it does not provide much help for the earlier part of the process involving selecting an actor to perform the task in the first place (see section 3.4). New tools, such as an “electronic yellow pages” or bidding schemes like those suggested by Turoff (1983) and Malone (1987) might be useful for this purpose.

<i>Level</i>	<i>Process</i>	<i>Example systems</i>
Coordination	Goal decomposition	Polymer (Croft and Lefkowitz, 1988)
	Task assignment (agreeing on commitments)	Coordinator (Winograd and Flores, 1986)
	Resource allocation (task prioritization)	Information Lens (sorting rules) (Malone, et al., 1987)
	Synchronizing	Meeting scheduling tools (e.g., Beard, et al., 1990)
	Sequencing	Polymer (Croft and Lefkowitz, 1988)
Group decision-making	Proposing alternatives	gIBIS (Conklin and Begeman, 1988), Sibyl (Lee, 1990), electronic brainstorming tools (e.g., Dennis, et al., 1988; Stefik, et al., 1987)
	Evaluating alternatives	gIBIS (Conklin and Begeman, 1988), Sibyl (Lee, 1990)
Communication	Transporting messages	Electronic mail
	Routing	Information Lens (“Anyone” rules) (Malone, et al., 1987), Computer conferencing (e.g., Lotus, 1989)
Perception of common objects	Seeing and manipulating shared information	Computer conferencing (e.g., Lotus, 1989), collaborative authoring tools (e.g., Ellis, et al., 1990; Fish, et al., 1988), electronic meeting rooms (e.g., Dennis, et al., 1988; DeSanctis and Gallupe, 1987; Stefik, et al., 1987)

Table 4. A taxonomy of cooperative work tools based on the processes they support

Another intriguing possibility suggested by this framework is that it might be possible to implement “primitives” for a number of different coordination-related processes in the same environment, and then let people combine these primitives in various ways to help solve particular coordination problems. This is one of the goals of the Object Lens system (Lai, et al., 1988).

4.2.3. Analyzing incentives for using cooperative work tools

Another use for coordination theory in designing cooperative work tools is to help systematically evaluate proposed (or actual) systems. For example, Markus and Connolly (1990) systematically analyze how the payoffs to individual users of a cooperative work system depend on how many other people are using the system. They do this by using an economic model from Schelling (1978) to extend Grudin’s (1988) insights about the incentives to use cooperative work systems. For instance, on-line calendars and many other cooperative work applications involve “discretionary databases” which users can view or update as they see fit. For each individual user, however, the benefits of viewing the database can be obtained without contributing anything. Thus, it is often in the interests of each individual user to use the database without making the effort required to contribute to it. Unfortunately, the equilibrium state of a system like this is for no one to ever contribute anything! One possible solution to this kind of problem is to change the payoff structure so that, for example, users who do not contribute to the database have to pay more to use it.

4.3. Designing distributed and parallel processing computer systems

As we noted above, coordination issues are central to the design of distributed and parallel processing computer systems. In this section, we describe examples of work that has addressed these issues in an explicitly interdisciplinary way, drawing on insights from other disciplines or kinds of systems to design or analyze distributed or parallel computer systems. In particular, we consider examples of (1) analogies with social and biological systems as a source of design ideas, and (2) quantitative tools for analyzing alternative designs.

4.3.1. Analogies with social and biological systems as a source of design ideas

Competitive bidding markets. One of the basic problems in designing distributed or parallel computer systems is how to assign tasks to processors, and several distributed computer systems have addressed this problem with competitive bidding mechanisms based on analogies with human markets. For example, the Contract Nets protocol (Davis and Smith, 1983; Smith and Davis, 1981) formalizes a sequence of messages to be exchanged by computer processors sharing tasks in a network. The “contracts” are arbitrary computational tasks that can potentially be performed by any of a number of processors on the network, the “clients” are machines at which these tasks originate, and the “contractors” are machines that might process the tasks (i.e., the servers). The sequence of announcement, bid, and award messages used by this protocol was described above in our analysis of the task assignment process (section 3.4.2). One of the desirable features of this system is its great degree of decentralization

and the flexibility it provides for how both clients and contractors can make their decisions. For instance, clients may select contractors on the basis of estimated completion time or the presence of specialized data; contractors may select tasks to bid on based on the size of the task or how long the task has been waiting.

Using these or similar ideas, a number of other bidding systems have been developed (e.g., Kurose and Simha, 1989; Stankovic, 1985). For instance, several bidding systems have been developed to allow personal workstations connected by a local area network to share tasks (Malone, et al., 1988; Waldspurger, et al., 1988). In this way, users can take advantage of the unused processing capacity at idle workstations elsewhere on the network. Furthermore, the local bidding "negotiations" can result in globally coherent processor scheduling according to various priorities (e.g., Malone, et al., 1988). (For a review of several related systems and an analysis of a variety of bidding algorithms, see Drexler and Miller, 1988; Miller and Drexler, 1988).

The notion of competitive bidding markets has also been suggested as a technique for storage management by Miller and Drexler (Drexler and Miller, 1988; Miller and Drexler, 1988). In their proposal, when object A wishes to maintain a pointer to object B, object A pays "rent" to the "landlord" of the space in which object B is stored. These rents are determined by competitive bidding, and when an object fails to pay rent, it is "evicted" (that is, garbage collected). Their proposal includes various schemes for how to determine rents, how to pass rents along a chain of references, and how to keep track of the various costs and payments without excessive overhead. They conclude that this proposal is not likely to be practical for small scale storage management (such as garbage collection of individual Lisp cells), but that it may well be useful for sharing large objects in complex networks that cross "trust boundaries" (e.g., inter-organizational networks). The scheme also appears useful for managing local caching and the migration of objects between different forms of short-term and long-term storage.

Scientific communities. Another central problem that arises in distributed and parallel processing systems is how and when to share information between processors. For instance, one interesting example of this problem arises in artificial intelligence programs that search a large space of possibilities, the nature of which is not well known in advance. It is particularly useful, in this case, for processors to exchange information about intermediate results in such a way that each processor can avoid performing work that is rendered unnecessary by work already done elsewhere.

One solution to this problem is suggested by the Scientific Community Metaphor embodied in the Ether system (Kornfeld, 1982; Kornfeld and Hewitt, 1981). In this system, there are a number of "sprites," each analogous to an individual scientist, that operate in parallel and interact through a global database. Each sprite requires certain conditions to be true in the global database before it is "triggered." When a sprite is triggered, it may (1) compute new results that are added to the global database, (2) create new sprites that await conditions that will trigger them, or (3) stifle a collection of sprites whose work is now known to be unnecessary. In one example use of this system, Kornfeld (1982) shows how sharing intermediate results in this way can dramatically improve the time

performance of an algorithm (even if it is executed by time-sharing a single processor). He calls this effect “combinatorial implosion.”

This system also uses the scientific community metaphor to suggest a solution to the resource allocation problem for processors. Each sprite is “supported” by a “sponsor,” and without a sponsor, a sprite will not receive any processing time to do its work. For instance, a sponsor may sometimes support both work directed toward proving some proposition and also work directed toward proving the negation of the proposition. Whenever one of these lines of work is successful, support is withdrawn from the other.

4.3.2. Analyzing stability properties of load sharing algorithms

Another use of coordination theory ideas is to help evaluate alternative designs of distributed and parallel processing computer systems. For instance, Huberman and his colleagues (Huberman and Hogg, 1988; Lumer and Huberman, 1990) have applied mathematical techniques like those used in chaos theory to analyze the dynamic behavior of distributed computer networks. In one case they analyze, for example, heavily loaded processors in a network transfer tasks to more lightly loaded processors according to a probabilistic process. When any processor in such a system can exchange tasks with any other processor, the behavior of the system is unstable for large numbers of processors (e.g., more than 21 processors in a typical example). However, when the processors are grouped hierarchically into clusters that exchange tasks frequently among themselves and only occasionally with other clusters, the system remains stable for arbitrarily large numbers of processors. This hierarchical arrangement has the disadvantage that it takes a long time to reach stability. In an intriguing analogy with human organizations, however, Huberman and his colleagues find that this disadvantage can be eliminated by having a few “lateral links” between different clusters in the hierarchy (Lumer and Huberman, 1990).

4.4 Summary of applications

As summarized in Table 5, the examples we have described show how coordination theory can help (1) analyze alternative designs, and (2) suggest new design ideas. In each case, these applications depended upon interdisciplinary use of theories or concepts about coordination.

5. RESEARCH AGENDA

We have seen how a number of different disciplines can contribute to answering the questions of coordination theory, and how theories of coordination can, in turn, be applied to the concerns of several different disciplines. What is needed to further develop this interdisciplinary study of coordination?

<i>Application area</i>	<i>Examples of analyzing alternative designs</i>	<i>Examples of generating new design ideas</i>
<i>Organizational structures and information technology</i>	Analyzing the effects of decreasing coordination costs on firm size, centralization, and internal structure	Creating temporary “intellectual marketplaces” to solve specific problems. Human analogs of beehives or dataflow computers
<i>Cooperative work tools</i>	Analyzing how the payoffs to individual users of a system depend on the number of other users	Designing new tools for task assignment, information routing, and group decision-making
<i>Distributed and parallel computer systems</i>	Analyzing stability properties of load sharing algorithms in computer networks	Using competitive bidding mechanisms to allocate processors and memory in computer systems. Using a scientific community metaphor to organize parallel problem-solving.

Table 5. Sample applications of coordination theory

As we suggested above, a central concern of coordination theory is identifying and analyzing specific coordination processes and structures. Therefore, a critical item on the agenda for coordination research should be developing these analyses. For example, the following kinds of questions arise:

5.1.1. Representing and classifying coordination processes

How can we represent coordination processes? When should we use flowcharts, Petri nets, or state transition diagrams? Are there other notations that are even more perspicuous for analyzing coordination? How can we classify different coordination processes? For instance, can we usefully regard some coordination processes as “special cases” of others? How are different coordination processes combined when activities are actually performed?

5.1.2. How general are coordination processes?

Another set of questions has to do with how generic coordination processes really are: How far can we get by analyzing very general coordination processes, and when will we find that most of the important factors are specific to coordinating a particular kind of task? For example, are there general heuristics for coordination that are analogous to the general problem-solving heuristics studied in cognitive science and artificial intelligence?

5.1.3. Analyzing specific processes

At least as important as these general questions are analyses of specific processes. For example, how far can we go in analyzing alternative coordination processes for problems such as resource allocation? Can we characterize an entire “design space” for solutions to this problem and analyze the major factors that would favor one solution over another in specific situations? Could we do the same thing for other processes such as goal selection or managing

timing interdependencies? Are there other processes (such as managing other kinds of interdependencies) that could be analyzed systematically in ways that have not yet been done?

In analyzing alternative processes for specific problems, we might consider various kinds of properties: Which processes are least “expensive” in terms of production costs and coordination costs? Which processes are fastest? Which processes are most stable in the face of failures of actors or delays of information? Which processes are most susceptible to incentive problems? For instance, how does the presence of significant conflicts of interest among actors affect the desirability of different resource allocation methods? How do information processing limitations of actors affect the desirability of different methods? For example, are some methods appropriate for coordinating people that would not be appropriate for coordinating computer processors, and vice versa? What new methods for coordinating people become desirable when human information processing capacities are augmented by computers?

5.1.4. Applications and methodologies

A critical part of the research agenda for this area is developing coordination theory in the context of various different kinds of systems. For instance, in section 4, we suggested numerous examples of these possibilities for human organizations and computer systems.

In some cases, this work may involve applying previously developed theories to these application areas. In many cases, however, we expect that new systems or new observations of these systems will stimulate the development of new theories. For example, all of the following methodologies appear likely to be useful in developing coordination theory: (1) empirically studying coordination in human or other biological systems (e.g., field studies, laboratory studies, or econometric studies), (2) designing new technologies for supporting human coordination, (3) designing and experimenting with new methods for coordinating distributed and parallel processing computer systems, and (4) formal modeling of coordination processes (e.g., mathematical modeling or computer simulation).

6. CONCLUSIONS

Clearly, the questions we have just listed are only the beginning of a set of research issues in the interdisciplinary study of coordination. However, we believe they illustrate how the notion of “coordination” provides a set of abstractions that help unify questions previously considered separately in a variety of different disciplines and suggests avenues for further exploration.

While much work remains to be done, it appears that this approach can build upon much previous work in these different disciplines to help solve a variety of immediate practical needs, including: (1) designing computer and communication tools that enable people to work together more effectively, (2) harnessing the power of multiple computer processors working simultaneously on related problems, and (3) creating more flexible and more satisfying ways of organizing collective human activity.

7. ACKNOWLEDGEMENTS

This work was supported, in part, by Digital Equipment Corporation, the National Science Foundation (Grant Nos. IRI-8805798 and IRI-8903034), the MIT International Financial Services Research Center, and General Motors/Electronic Data Systems.

Parts of this paper were included in two previous papers (Malone, 1988; Malone and Crowston, 1990). We are especially grateful to Deborah Ancona, John Carroll, Michael Cohen, Randall Davis, John Little, and Wanda Orlikowski for comments on earlier versions of the paper, and to participants in numerous seminars and workshops at which these ideas have been presented.

**8. APPENDIX:
PREVIOUSLY SUGGESTED DEFINITIONS OF COORDINATION**

“The operation of complex systems made up of components.” (NSF-IRIS, 1989)

“The emergent behavior of collections of individuals whose actions are based on complex decision processes.” (NSF-IRIS, 1989)

“Information processing within a system of communicating entities with distinct information states.” (NSF-IRIS, 1989)

“The joint efforts of independent communicating actors towards mutually defined goals.” (NSF-IRIS, 1989)

“Networks of human action and commitments that are enabled by computer and communications technologies.” (NSF-IRIS, 1989)

“Composing purposeful actions into larger purposeful wholes.” (A. Holt, personal communication, 1989)

“Activities required to maintain consistency within a work product or to manage dependencies within the workflow.” (Curtis, 1989)

“The additional information processing performed when multiple, connected actors pursue goals that a single actor pursuing the same goals would not perform.” (Malone, 1988)

REFERENCES

- Allen, J., Hendler, J. and Tate, A. (Eds.). *Readings in Planning*. San Mateo, CA: Morgan Kaufmann.
- Allen, T. J. (1977). *Managing the Flow of Technology*. Cambridge, MA: MIT Press.
- American Heritage Dictionary. (1981). Boston: Houghton Mifflin.
- Arrow, K. J. (1951). *Social Choice and Individual Value* (Vol. 12). New York: John Wiley & Sons.
- Arvind and Culler, D. E. (1986). Dataflow architectures. In *Annual Reviews in Computer Science* (Vol. 1). Palo Alto, CA: Annual Reviews, INC.
- Attewell, P. and Rule, J. (1984). Computing and organizations: What we know and what we don't know. *Communications of the ACM*, 27, 1184-1192.
- Aumann, R. J. (1976). Agreeing to disagree. *Annals of Statistics*, 1236-1239.
- Baligh, H. H. (1986). Decision rules and transactions, organizations and markets. *Management Science*, 32, 1480-1491.
- Baligh, H. H. and Burton, R. M. (1981). Describing and designing organizational structures and processes. *International Journal of Policy Analysis and Information Systems*, 5, 251-266.
- Baligh, H. H. and Richartz, L. (1967). *Vertical Market Structures*. Boston, MA: Allyn and Bacon.
- Barnard, C. I. (1964). *The Functions of the Executive*. Cambridge, MA: Harvard University.
- Beard, D., Murugappan, P., Humm, A., Banks, D., Nair, A. and Shan, Y.-P. (1990). A Visual Calendar for Scheduling Group Meetings. In D. Tatar (Ed.), *Proceedings of the Third Conference on Computer-supported Cooperative Work* (pp. 279-290). Los Angeles, CA: ACM Press.
- Beer, S. (1967). *Cybernetics and Management* (2nd ed.). London: English Universities Press.
- Bernstein, P. and Goodman, N. (1981). Concurrency control in distributed database systems. *ACM Computing Surveys*, (June).
- Bond, A. H. and Gasser, L. (Eds.). (1988). *Readings in Distributed Artificial Intelligence*. San Mateo, CA: Morgan Kaufman.
- Boulding, K. E. (1956). *The Image*. Ann Arbor, MI: University of Michigan.
- Bruns, W. J. and McFarlan, F. W. (1987). Information technology puts power in control systems. *Harvard Business Review*, (Sept.-Oct.), 89-94.
- Brynjolfsson, E., Malone, T., Gurbaxani, J. and Kambil, A. (1989). *Does Information Technology Lead to Smaller Firms?* (Technical report 106). Center for Coordination Science, MIT.
- Burton, R. M. and Obel, B. (1980a). A computer simulation test of the M-form hypothesis. *Administrative Science Quarterly*, 25, 457-466.
- Burton, R. M. and Obel, B. (1980b). The efficiency of the price, budget, and mixed approaches under varying *a priori* information levels for decentralized planning. *Management Science*, 26.

- Carriero, N. and Gelernter, D. (1989). *Coordination languages and their significance* (Working paper YALEU/DCS/RR-716). New Haven, CT: Department of Computer Science, Yale University.
- Chandler, A. D., Jr. (1962). *Strategy and Structure: Chapters in the History of the American Industrial Enterprise*. Cambridge, MA: MIT Press.
- Chandler, A. D. (1977). *Administrative coordination, allocation and monitoring: Concepts and comparisons* (Working paper 77-21). Brussels: European Institute for Advanced Studies in Management.
- Chapman, D. (1987). Planning for conjunctive goals. *Artificial Intelligence*, 32, 333–377.
- Ciborra, C. U. (1987). Reframing the role of computers in organizations: The transaction costs approach. *Office: Technology and People*, 3, 17–38.
- Conklin, J. and Begeman, M. L. (1988). gIBIS: A hypertext tooling for exploratory policy discussion. In D. Tatar (Ed.), *Proceedings of the Second Conference on Computer-supported Cooperative Work* (pp. 140–152). Portland, OR: ACM Press.
- Conway, R. W., Maxwell, W. L. and Miller, L. W. (1967). *Theory of Scheduling*. Reading, MA: Addison-Wesley.
- Crawford, A. B., Jr. (1982). Corporate electronic mail—A communication-intensive application of information technology. *MIS Quarterly*, 6(September), 1–13.
- Croft, W. B. and Lefkowitz, L. S. (1988). Using a planner to support office work. In R. B. Allen (Ed.), *Proceedings of the ACM Conference on Office Information Systems* (pp. 55-62). Palo Alto, CA.
- Crowston, K. (1991). *Towards a Coordination Cookbook: Recipes for Multi-Agent Action*. Doctoral Dissertation, MIT Sloan School of Management, Cambridge, MA.
- Crowston, K., Malone, T. W. and Lin, F. (1987). Cognitive science and organizational design: A case study of computer conferencing. *Human Computer Interaction*, 3, 59-85.
- Curtis, B. (1989). Modeling coordination from field experiments. In *Organizational Computing, Coordination and Collaboration: Theories and Technologies for Computer-Supported Work*. Austin, TX.
- Cytron, R. (1987). Limited processor scheduling of doacross loops. In *Proceedings of the 1987 International Conference on Parallel Processing* (pp. 226-234).
- Dantzig, G. B. (1963). *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press.
- Davis, R. and Smith, R. G. (1983). Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20, 63–109.
- Debreu, G. (1959). *Theory of value: An axiomatic analysis of economic equilibrium*. New York: Wiley.
- Deitel, H. M. (1983). *An Introduction to Operating Systems*. Reading, MA: Addison-Wesley.
- Deneubourg, J. L. and Gross, S. (1989). Collective patterns and decision-making. *Ethology, Ecology & Evolution*, 1, 295–311.
- Dennett, D. C. (1987). *The Intentional Stance*. Cambridge, MA: MIT Press.

- Dennis, A. R., Joey, F. G., Jessup, L. M., Nunamaker, J. F. and Vogel, D. R. (1988). Information Technology to Support Electronic Meetings. *MIS Quarterly*, 12:4(December), 591-619.
- Dennis, J. B. (1974). First version of a data flow procedure language. In B. Robinet (Ed.), *Proceedings, Colloque sur la Programmation, Lecture Notes in Computer Science* (Vol. 54) (pp. 231-242). Berlin: Springer-Verlag.
- Building the Information Marketplace. (1991, January). *Technology Review*, pp. 29-40.
- DeSanctis, G. and Gallupe, B. (1987). A foundation for the study of group decision support systems. *Management Science*, 33(5), 589-609.
- Dijkstra, E. W. (1968). The structure of the T. H. E. operating system. *Communications of the ACM*, 11(5), 341-346.
- Drexler, K. E. and Miller, M. S. (1988). Incentive engineering for computational resource management. In B. A. Huberman (Ed.), *The Ecology of Computation* (pp. 231-266). Amsterdam: Elsevier.
- Durfee, E. D. and Lesser, V. R. (1987). Using partial global plans to coordinate distributed problem solvers. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)* (pp. 875-883).
- Ellis, C. A., Gibbons, R. and Morris, P. (1979). Office streamlining. In *Proceedings of the International Workshop on Integrated Offices*. Versailles, France. Institut de Recherche d'Informatique de d'Automatique.
- Ellis, C. A., Gibbs, S. J. and Rein, G. L. (1990). Design and use of a group editor. In G. Cockton (Ed.), *Engineering for Human-Computer Interaction* (pp. 13-25).
- Ellis, C. A., Gibbs, S. J. and Rein, G. L. (1991). Groupware, some issues and experiences. *Communications of the ACM*, 34(1), 38-57.
- Emery, J. C. (1969). *Organizational Planning and Control Systems: Theory and Technology*. New York: MacMillan.
- Erman, L. D., Hayes-Roth, F., Lesser, V. R. and Reddy, D. R. (1980). The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, 12(2), 213-253.
- Farrell, J. and Saloner, G. (1985). Standardization, compatibility, and innovation. *Rand Journal of Economics*, 16(Spring), 70-83.
- Fikes, R. E. and Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2, 198-208.
- Fish, R., Kraut, R., Leland, M. and Cohen, M. (1988). Quilt: A collaborative tool for cooperative writing. In *Proceedings of the Conference on Office Information Systems* (pp. 30-37). Palo Alto, CA.
- Flores, F., Graves, M., Hartfield, B. and Winograd, T. (1988). Computer systems and the design of organizational interaction. *ACM Transactions on Office Information Systems*, 6(2), 153-172.
- Forrester, J. W. (1980). *Systems Dynamics*. New York: North-Holland.
- Fox, M. S. (1981). An organizational view of distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1), 70-79.

- Franks, N. R. (1989). Army ants: A collective intelligence. *American Scientist*, 77(March-April), 139–145.
- Galbraith, J. R. (1973). *Designing Complex Organizations*. Reading, MA: Addison-Wesley.
- Galbraith, J. R. (1977). *Organization Design*. Reading, MA: Addison-Wesley.
- Gardner, D. (1985). *The Mind's New Science: A History of the Cognitive Revolution*. New York: Basic.
- Gray, J. (1978). *Notes on data base operating systems* (Research Report RJ 2188). IBM.
- Greif, I. (1988). Computer-supported cooperative work: Breakthroughs for user acceptance (Panel Description). In D. Peterson (Ed.), *Proceedings of the CSCW '86 Conference on Computer-supported Cooperative Work*. Austin, TX: ACM.
- Grudin, J. (1988). Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. In D. Tatar (Ed.), *Proceedings of the Second Conference on Computer-supported Cooperative Work* (pp. 85-93). Portland, OR: ACM Press.
- Gurbaxani, V. and Whang, S. (1991). The Impact of Information Systems on Organizations and Markets. *Communications of the ACM*, 34(1), 59-73.
- Halpern, J. Y. (1987). Using reasoning about knowledge to analyze distributed systems. In *Annual Review of Computer Science* (Vol. 2) (pp. 37-68). Palo Alto, CA: Annual Reviews Inc.
- Halpern, J. Y. and Moses, Y. (1986). *Knowledge and common knowledge in a distributed environment* (Report RJ 4421 (47909) Computer Science). San Jose, CA: IBM Almaden Research Center.
- Halstead, R. H. (1985). Multilisp: A language for concurrent symbolic computation. *ACM Transactions on Programming Languages and Systems*.
- Hart, P. and Estrin, D. (1990). Inter-organization computer networks: Indications of shifts in interdependence. In *Proceedings of the ACM Conference on Office Information Systems*. Cambridge, MA.
- Hewitt, C. (1986). Offices are open systems. *ACM Transactions on Office Systems*, 4(3), 271–287.
- Hewitt, C. E. (1977). Viewing control structures as patterns of message passing. *Journal of Artificial Intelligence*, 8(3), 323–364.
- Hirsch, E. D. (1987). *Cultural Literacy: What Every American Needs to Know*. Boston, MA: Houghton Mifflin.
- Hoare, C. A. R. (1975). Monitors: An operating systems structuring concept. *Communications of the ACM*, 17(10), 549–557. Corrigendum, *CACM*, 18(2): 95.
- Holt, A. W. (1988). Diplans: A new language for the study and implementation of coordination. *ACM Transactions on Office Information Systems*, 6(2), 109–125.
- Huberman, B. A. (Ed.). (1988a). *The Ecology of Computation*. Amsterdam: North-Holland.
- Huberman, B. A. (1988b). *Open systems: The ecology of computation* (PARC working paper P88-00074). Palo Alto, CA: Xerox PARC.
- Huberman, B. A. and Hogg, T. (1988). The behaviour of computational ecologies. In B. A. Huberman (Ed.), *The Ecology of Computation* (pp. 77-116). Amsterdam: Elsevier.

- Huhns, M. N. and Gasser, L. (Eds.). (1989). *Distributed Artificial Intelligence* (Vol. 3). San Mateo, CA: Morgan Kaufmann.
- Hurwicz, L. (1973). The design of resource allocation mechanisms. *American Economic Review Papers and Proceedings*, 58(May), 1-30.
- Janis, I. L. and Mann, L. (1977). *Decision-making*. New York: Free Press.
- Jensen, M. C. and Meckling, W. H. (1976). Theory of the firm: Managerial behavior, agency costs and ownership structure. *Journal of Financial Economics*, 3, 305-360.
- Johansen, R. (1988). *Groupware: Computer Support for Business Teams*. New York: The Free Press.
- Kahneman, D. and Tversky, A. (1973). On the psychology of prediction. *Psychological Review*, 80, 237-251.
- Kidder, T. (1981). *The Soul of a New Machine*. Boston: Little, Brown.
- Kiesler, S., Siegel, J. and McGuire, T. W. (1984). Social psychological aspects of computer-mediated communication. *American Psychologist*, 39, 1123-1134.
- Kohler, W. (1981). A survey of techniques for synchronization and recovery in decentralized computer systems. *ACM Computing Surveys*, (June).
- Kornfeld, W., A. (1982). Combinatorially Implosive Algorithms. *Communications of the ACM*, 25(10), 734-738.
- Kornfeld, W. A. and Hewitt, C. (1981). The scientific community metaphor. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11, 24-33.
- Kraemer, K. and King, J. L. (1988). Computer-based systems for cooperative work and group decision-making. *ACM Computing Surveys*, 20(2), 115-146.
- Kurose, J. F. and Simha, R. (1989). A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transaction on Computers*, 38(5), 705-717.
- Lai, K. Y., Malone, T. and Yu, K.-C. (1988). Object Lens: A spreadsheet for cooperative work. *ACM Transactions on Office Information Systems*, 6(4), 332-353.
- Lawrence, P. and Lorsch, J. (1967). *Organization and Environment*. Boston, MA: Division of Research, Harvard Business School.
- Leavitt, H. J. and Whisler, T. L. (1958). Management in the 1980's. *Harvard Business Review*, 36(November/December), 41-48.
- Lee, J. (1990). Sibyl: A qualitative decision management system. In P. Winston (Ed.), *Artificial Intelligence at MIT: Expanding Frontiers* (Vol. 1). Cambridge, MA: MIT Press.
- Lee, J. and Malone, T. W. (1988). How can groups communicate when they use different languages? Translating between partially shared type hierarchies. In R. B. Allen (Ed.), *Proceedings of the ACM Conference on Office Information Systems* (pp. 22-29). Palo Alto, CA.
- Lee, J. and Malone, T. W. (1990). Partially shared views: A scheme for communicating among groups that use different type hierarchies. *ACM Transactions on Information Systems*, 8(1), 1-26.

- Liskov, B. and Guttag, J. (1986). *Abstraction and Specification in Program Development*. Cambridge, MA: MIT Press.
- Lotus. (1989). *Lotus Notes Users Guide*. Cambridge, MA: Lotus Development Corp.
- Luce, R. D. and Raiffa, H. (1957). *Games and Decisions*. New York: Wiley.
- Lumer, E. and Huberman, B. A. (1990). *Dynamics of resource allocation in distributed systems* (SSL-90-05). Palo Alto, CA: Xerox PARC.
- Lynch, N. A. and Fischer, M. J. (1981). On describing the behavior and implementation of distributed systems. *Theoretical Computer Science*, 13(1), 17-43.
- Malone, T. W. (1987). Modeling coordination in organizations and markets. *Management Science*, 33, 1317-1332.
- Malone, T. W. (1988). *What is coordination theory?* (Working paper #2051-88). Cambridge, MA: MIT Sloan School of Management.
- Malone, T. W. (1990). Organizing information processing systems: Parallels between organizations and computer systems. In W. Zachary, S. Robertson and J. Black (Eds.), *Cognition, Computation, and Cooperation* (pp. 56-83). Norwood, NJ: Ablex.
- Malone, T. W. (in press). Analogies Between Human Organization and Artificial Intelligence Systems: Two Examples and Some Reflections. In M. Masuch (Ed.), *Distributed Intelligence: Perspectives of Artificial Intelligence on Organization and Management Theory*. Amsterdam: Elsevier.
- Malone, T. W. and Crowston, K. (1990). What is coordination theory and how can it help design cooperative work systems? In D. Tatar (Ed.), *Proceeding of the Third Conference on Computer-supported Cooperative Work* (pp. 357-370). Los Angeles, CA: ACM Press.
- Malone, T. W., Fikes, R. E., Grant, K. R. and Howard, M. T. (1988). Enterprise: A market-like task scheduler for distributed computing. In B. Huberman (Ed.), *The Ecology of Computation*. New York: North-Holland.
- Malone, T. W., Grant, K. R., Turbak, F. A., Brobst, S. A. and Cohen, M. D. (1987). Intelligent information-sharing systems. *Communications of the ACM*, 30, 390-402.
- Malone, T. W. and Smith, S. A. (1988). Modeling the performance of organizational structures. *Operations Research*, 36(3), 421-436.
- Malone, T. W., Yates, J. and Benjamin, R. I. (1987). Electronic markets and electronic hierarchies. *Communications of the ACM*, 30, 484-497.
- Mangel, M. and Clark, C. W. (1988). *Dynamic Modeling in Behavioral Ecology*. Princeton, NJ: Princeton University Press.
- March, J. G. and Simon, H. A. (1958). *Organizations*. New York: John Wiley and Sons.
- Markus, M. L. and Connolly, T. (1990). Why CSCW applications fail: Problems in the adoption of interdependent work tools. In D. Tatar (Ed.), *Proceedings of the Third Conference on Computer-supported Cooperative Work* (pp. 371-380). Los Angeles, CA: ACM Press.

- Marschak, J. and Radner, R. (1972). *Economic Theory of Teams*. New Haven: Yale University.
- McGrath, J. E. (1984). *Groups: Interaction and Performance*. Englewood Cliffs, NJ: Prentice-Hall.
- Miao, X., Luh, P. B. and Kleinman, D. L. (in press). A normative-descriptive approach to hierarchical team resource allocation. *IEEE Transactions on Systems, Man and Cybernetics*.
- Milgrom, P. (1981). An axiomatic characterization of common knowledge. *Econometrica*, 49(1), 17-26.
- Milgrom, P. and Roberts, J. (1990). The Economics of Modern Manufacturing: Technology, Strategy, and Organization. *The American Economic Review*, 80(3), 511-528.
- Miller, J. G. (1978). *Living Systems*. New York: McGraw-Hill.
- Miller, M. S. and Drexler, K. E. (1988). Markets and computation: Agoric open systems. In B. A. Huberman (Ed.), *The Ecology of Computation* (pp. 133-176). Amsterdam: North-Holland.
- Minsky, M. (1987). *The Society of the Mind*. New York: Simon and Schuster.
- Mintzberg, H. (1979). *The Structuring of Organizations*. Englewood Cliffs, NJ: Prentice-Hall.
- Moses, J. (1990). *Organization and Ideology* (Unpublished manuscript). Cambridge, MA: Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.
- Myerson, R. B. (1981). Optimal auction design. *Mathematics of Operations Research*, 6, 58-73.
- Newell, A. and Simon, H. A. (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall, Inc.
- Nii, P. (1986). The blackboard model of problem solving. *AI Magazine*, (Spring), 38-53.
- Nilsson, N. (1971). *Problem-Solving Methods in Artificial Intelligence*. New York: McGraw-Hill.
- Norman, D. A. (1980). Twelve issues for cognitive science. *Cognitive Science*, 4, 1-32.
- NSF-IRIS. (1989). *A report by the NSF-IRIS Review Panel for Research on Coordination Theory and Technology*. Available from NSF Forms & Publications Unit.
- Peterson, D. (1986). *Proceedings of the Conference on Computer-Supported Cooperative Work*. Austin, TX: ACM.
- Peterson, J. L. (1977). Petri nets. *ACM Computing Surveys*, 9(3), 223-252.
- Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems*. Englewood Cliff, NJ: Prentice-Hall.
- Pfeffer, J. (1978). *Organizational Design*. Arlington Heights, IL: Harlan Davidson.
- Pfeffer, J. and Salancik, G. R. (1978). *The External Control of Organizations: A Resource Dependency Perspective*. New York: Harper & Row.
- Reiter, S. (1986). Informational incentive and performance in the (new)² welfare economics. In S. Reiter (Ed.), *Studies in Mathematical Economics (Studies in Mathematics, Volume 25)*. Mathematical Association of America.
- Rescher, N. (1977). *Dialectics: A Controversy-Oriented Approach to the Theory of Knowledge*. Buffalo, NY: State University of New York Press.

- Rittel, H. and Kunz, W. (1970). *Issues as elements of information systems* (Working paper 131). Stuttgart, Germany: University of Stuttgart, Institut für Grundlagen der Planung I.A.
- Rockart, J. F. and Short, J. E. (1989). IT and the networked organization: Toward more effective management of interdependence. In M. S. Scott Morton (Ed.), *Management in the 1990s Research Program Final Report*. Cambridge, MA: Massachusetts Institute of Technology.
- Rogers, E. and Agarwala-Rogers, R. (1976). *Communication in organizations*. New York: Free Press.
- Ross, S. (1973). The economic theory of agency. *American Economic Review*, 63, 134–139.
- Rumerhart, D. E., McClelland, J. L. and PDP Research Group. (1986). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*. Cambridge: MIT Press.
- Schelling, T., C. (1978). *Micromotives and Macrobehavior*. New York, N.Y.: Norton.
- Schelling, T. C. (1960). *Strategy of Conflict*. Cambridge, MA: Harvard University Press.
- Searle, J. R. (1975). A taxonomy of illocutionary acts. In K. Gunderson (Ed.), *Language, Mind and Knowledge* (pp. 344–369). Minneapolis, MN: University of Minnesota.
- Seeley, T. D. (1989). The honey bee colony as a superorganism. *American Scientist*, 77(November-December), 546–553.
- Shannon, C. E. and Weaver, W. (1949). *The mathematical theory of communication*. Urbana: University of Illinois Press.
- Simon, H. A. (1976). *Administrative Behavior* (3rd ed.). New York: Free Press.
- Smith, R. G. and Davis, R. (1981). Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1), 61–70.
- Stankovic, J. (1985). An application of Bayesian decision theory to decentralized control of job scheduling. *IEEE Transactions on Computers*, C-34(2), 117-130.
- Stefik, M. and Bobrow, D. G. (1986). Object-oriented programming: Themes and variations. *AI Magazine*, (Spring), 40–62.
- Stefik, M., Foster, G., Bobrow, D. G., Kahn, K., Lanning, S. and Suchman, L. (1987). Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *Communications of the ACM*, 30(1), 32-47.
- Stephens, D. W. and Krebs, J. R. (1986). *Foraging Theory*. Princeton, NJ: Princeton University Press.
- Stoddard, D. (1986). *OTISLINE* (Case 9-186-304). Harvard Business School.
- Suchman, L. A. (1987). *Plans and Situated Actions: The Problem of Human-Machine Communication*. New York: Cambridge University Press.
- Tannenbaum, A. S. (1981). *Computer Networks*. Englewood Cliffs, NJ: Prentice-Hall.
- Tatar, D. (Ed.). (1988). *Proceedings of the Second Conference on Computer-supported Cooperative Work*. Portland, OR: ACM Press.

- Tatar, D. (Ed.). (1990). *Proceedings of the Third Conference on Computer-supported Cooperative Work*. Los Angeles, CA: ACM Press.
- Thompson, J. D. (1967). *Organizations in Action: Social Science Bases of Administrative Theory*. New York: McGraw-Hill.
- Toffler, A. (1970). *Future Shock*. New York: Bantam Books.
- Toulmin, S. (1958). *The Uses of Argument*. Cambridge, England: Cambridge University Press.
- Turoff, M. (1983). *Information, value, and the internal marketplace* (Unpublished manuscript). New Jersey Institute of Technology.
- von Bertalanffy, L. (1950). The theory of open systems in physics and biology. *Science*, 111.
- Waldspurger, C. A., Hogg, T., Huberman, B. A., Kephart, J. O. and Stornetta, S. (1988). *Spawn: A distributed computational ecology* (Unpublished working paper). Palo Alto, CA: Xerox PARC.
- Weick, K. E. (1969). *The social psychology of organizing*. Reading, MA: Addison-Wesley.
- Wiener, N. (1961). *Cybernetics: Or Control and Communication in the Animal and the Machine* (2nd ed.). Cambridge, MA: MIT Press.
- Williamson, O. (1985). *The Economic Institutions of Capitalism*. New York: Free Press.
- Williamson, O. E. (1975). *Markets and Hierarchies*. New York: Free Press.
- Winograd, T. (1987). A language/action perspective on the design of cooperative work. *Human Computer Interaction*, 3, 3-30.
- Winograd, T. and Flores, F. (1986). *Understanding computers and cognition: A new foundation for design*. Norwood, NJ: Ablex.