56p

The Interdisciplinary Study of Coordination

by

Thomas Malone and Kevin Crowston

CCS WP #157, Sloan School WP #3630-93
November 1993

The Interdisciplinary Study of Coordination

Thomas W. Malone and Kevin Crowston*

November 1993

Forthcoming in *ACM Computing Surveys*.

* Current affiliation: University of Michigan School of Business Administration

Table of Contents

ABSTRACT

This paper characterizes an emerging research area, sometimes 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.

A key insight of the framework presented here is that coordination can be seen as the process of *managing dependencies* between activities. Further progress, therefore, should be possible by characterizing different kinds of dependencies and identifying the coordination processes that can be used to manage them. A variety of processes are analyzed from this perspective and commonalities across disciplines are identified. Processes analyzed include those for managing *shared resources, producer / consumer relationships, simultaneity constraints,* and *task / subtask dependencies*.

A major section of the paper summarizes ways of applying a coordination perspective 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.

The Interdisciplinary Study 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; Rumelhart, et al., 1986; Winograd & Flores, 1986; NSF-IRIS, 1989; NSF, 1991; Bond & Gasser, 1988; Huhns & Gasser, 1989). 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. 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 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. We use the term "theory" with some hesitation because it connotes to some people a degree of rigor and coherence that is not yet present in this field. Instead, the field today is a collection of intriguing analogies, scattered results, and partial frameworks. We use the term "theory", however, in part to signify a provocative goal for this interdisciplinary enterprise, and we hope that the various studies reviewed in this paper will serve as steps along the path toward an emerging theory of coordination.

1.1. A motivating question

We begin with one of the questions that coordination theory may help answer: *How will the widespread use of information technology change the ways people work together?* This is not the only possible focus of coordination theory, but it is a particularly timely question today for two reasons:

(l) 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 3.3).

It now appears likely that there will be a number of 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 trial and error and intuition, 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).

1.2. How can we proceed?

If we believe that new forms of organizing are likely to become more common, how can we understand the possibilities better? What other new kinds of coordination structures will emerge in the electronically connected world of the near future? When are these new structures desirable? What is necessary for them to work well?

To some extent, we can answer these questions by observing innovative organizations as they experiment with new technologies. But to understand the experiences of these organizations, we may need to look more deeply into the fundamental constraints on how coordination can occur. And to imagine new kinds of organizational processes that no organizations have tried yet, we may need to look even further afield for ideas.

One way to do both these things--to understand fundamental constraints and to imagine new possibilities--is to look for analogies in how coordination occurs in very different kinds of systems. For example, could we learn something about tradeoffs between computing and communicating in distributed computer systems that would illuminate possibilities for coordination in human organizations? Might coordination structures analogous to those used in bee hives or ant colonies be useful for certain aspects of human organizations? And could lessons learned about coordination in human systems help understand computational or biological systems, as well?

For these possibilities to be realized, a great deal of cross-disciplinary interaction is needed. It is not enough just to believe that different systems are similar; we also need an intellectual framework for "transporting" concepts and results back and forth between the different kinds of systems.

In the remainder of this paper, we attempt to provide the beginnings of such a framework. We first define coordination in a way that emphasizes its interdisciplinary nature and then suggest an approach for studying it further. Next, we describe examples of how a coordination perspective can be applied in three 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. Finally, we briefly suggest elements of a research agenda for this new area.

2. A FRAMEWORK FOR STUDYING COORDINATION

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 where we thought we had a reservation is fully booked, or when our favorite word processing program stops working in a new version of the operating system, we may become very aware of the effects of poor coordination.

For many purposes, this intuitive meaning is sufficient. However, in trying to characterize a new interdisciplinary area, it is also helpful to have a more precise idea of what we mean by "coordination." Appendix A 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, it is useful to begin with the following simple definition:

Coordination is managing dependencies between activities.[1]

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

As the definition suggests, we believe it is helpful to use the word "coordination" in a fairly inclusive sense. For instance, it is clear that actors performing interdependent activities may have conflicting interests and that what might be called "political processes" are ways of managing them (e.g., Ciborra, 1987; Williamson, 1985; Schelling, 1960; Kling, 1980). Similarly, even though words like "cooperation," "collaboration," and "competition" each have their own connotations, an important part of each of them involves managing dependencies between activities.[2]

It should also be clear that coordination, as we have defined it, can occur in many kinds of systems: human, computational, biological, and others. For instance, questions about how people manage dependencies among their activities are central to parts of organization theory, economics, management science, sociology, social psychology, anthropology, linguistics, law, and political

[1] This definition was particularly influenced by Rockart and Short (1989) and Curtis (1989). The importance of coordination in this very general sense was perhaps first recognized by Holt (1980; 1983).

[2] These terms also, of course, have broader meanings. For instance, cooperation usually implies shared goals among different actors, competition usually implies that one actor's gains are another's losses, and collaboration often connotes peers working together on an intellectual endeavor. However, it is sometimes useful to consider all these terms as describing different approaches to managing dependencies among activities, that is, as different forms of coordination.

science. In computer systems, dependencies between different computational processes must certainly be managed, 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; 1988b; Miller & Drexler, 1988; Smith & Davis, 1981). To give a sense of the approaches different fields have taken to studying coordination, we summarize in Appendix B examples of results about coordination from computer science, organization theory, economics, and biology.

Even though we believe there are more similarities among these different kinds of systems than most people appreciate, there are obviously many differences as well. One of the most important differences is that issues of *incentives, motivations*, and *emotions* are usually of much more concern in human systems than in other kinds of systems. In computer programs, for example, the "incentives" of a program module are usually easy to describe and completely controlled by a programmer. In human systems, on the other hand, the motivations, incentives, and emotions of people are often extremely complex, and understanding them is usually an important part of coordination. Even in human systems, however, analogies with other kinds of systems may help us understand fundamental constraints on coordination and imagine new kinds of organizations that might be especially motivational for people.

2.2. Basic coordination processes

A primary vehicle for facilitating transfer among these different disciplines is identifying and studying the basic *processes* involved in coordination: Are there fundamental coordination processes that occur in all coordinated systems? If so, 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?

One of the advantages of the definition we have used for coordination is that it suggests a direction for addressing these questions. If coordination is defined as managing dependencies, then further progress should be possible by *characterizing different kinds of dependencies and identifying the coordination processes that can be used to manage them.*

Table 1 suggests the beginnings of such an analysis (see Malone, et al., 1993, for more details). For example, one possible kind of dependency between different activities is that they require the same (limited) resources. The table shows that shared resource constraints can be managed by a variety of coordination processes such as "first come / first serve", priority order, budgets, managerial decision, and market-like bidding. If three job shop workers need to use the same machine, for instance, they could use a simple "first come / first serve" mechanism. Alternatively, they could use a form of budgeting with each worker having pre-assigned time slots, or a manager could explicitly decide what to do whenever two workers wanted to use the machine at the same time. In some cases, they might even want to "bid" for use of the machine and the person willing to pay the most would get it.

Dependency	***Examples of coordination processes for managing dependency***
Shared resources	"First come/first serve", priority order, budgets, managerial decision, market-like bidding
Task assignments	(same as for "Shared resources")
Producer / consumer relationships	
Prerequisite constraints	Notification, sequencing, tracking
Transfer	Inventory management (e.g., "Just In Time", "Economic Order Quantity")
Usability	Standardization, ask users, participatory design
Design for manufacturability	Concurrent engineering
Simultaneity constraints	Scheduling, synchronization
Task / subtask	Goal selection, task decomposition

Table 1.
Examples of common dependencies between activities and alternative coordination processes for managing them. (Indentations in the left column indicate more specialized versions of general dependency types.)

The lists of dependencies and coordination processes in Table 1 are by no means intended to be exhaustive. It is important to note, however, that many specific processes that arise in particular kinds of systems (such as "design for manufacturability") can be seen as instances of more generic processes (such as managing "usability" constraints between adjacent steps in a process).

In fact, we believe that one of the most intriguing possibilities for coordination theory is to identify and systematically analyze a wide variety of dependencies and their associated coordination processes. Such a "handbook" of coordination processes could not only facilitate interdisciplinary transfer of knowledge about coordination, it could also provide a guide for analyzing the coordination needs in particular situations and generating alternative ways of fulfilling them (see Malone, et al., 1993).

One question that arises immediately is how to categorize these dependencies and coordination processes. Table 1 provides a start in this direction. Crowston (1991) suggests a more structured taxonomy based on all the possible relationships between "tasks" and "resources."

To illustrate the possibilities for analyzing coordination processes, we will discuss in the remainder of this section the coordination processes listed in Table 1 and how they have been analyzed in different disciplines.

2.2.1. Managing shared resources

Whenever multiple activities share some limited resource (e.g., money, storage space, or an actor's time), a *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, it has received significant attention in economics, organization theory, and computer science.

Economics. Much of economics is devoted to studying resource allocation processes, especially those involving market-like pricing and bidding mechanisms. As economists have observed, for instance, markets have a number of interesting properties as resource allocation mechanisms (Simon, 1981). For one thing, they can be very *decentralized*: many independent decision-makers interacting with each other locally can produce a globally coherent allocation of resources without any centralized controller (e.g., Smith, 1776). For another thing, markets have a built-in set of *incentives*: when all participants in a perfect market try to maximize their own individual benefits, the overall allocation of resources is (in a certain sense) globally "optimal" (e.g., Debreu, 1959).

Organization theory. Organization theory has also paid great attention to resource allocation issues. For instance, control of resources is intimately connected with personal and organizational power: those who control resources have power and vice versa (e.g., Pfeffer & Salancik, 1978). In general, organization theorists emphasize hierarchical resource allocation methods where managers at each level decide how the resources they control will be allocated among the people who report to them (e.g., Burton & Obel, 1980a; 1980b). In practice, however, resource allocation in organizations is much more complex than a simple hierarchical model suggests. For instance, managers may try to increase their own power by attracting resources (e.g., employees and money) away from other possible activities (Barnard, 1964) or by using their resources in a way that is very suboptimal from the point of view of the whole organization.

How can we choose between different resource allocation methods? Recent work in *transaction cost theory* addresses part of this question by analyzing the conditions under which a hierarchy is a better way of coordinating multiple actors than a market (e.g., Williamson, 1975; 1985). For example, if there are extra costs associated with a market transaction (such as extensive legal and accounting work), then the costs of internal transactions within a hierarchical firm may be lower and therefore preferable. A related question involves the conditions under which it is desirable to use market-like resource allocation mechanisms (such as transfer pricing) within a hierarchical organization (Eccles, 1985).

Computer science. Resource allocation issues also arise in computer systems and much work has been done on these topics (e.g., Cytron, 1987; Halstead, 1985). For instance, 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). As we will see below, there have also already been examples of cross-fertilization of ideas about resource allocation between computer science and other fields. For example, in section 2.3.3, we will see how ideas about distributed computer systems helped understand the evolution of human organizations, and in section 3.4, we will see how analogies with human markets have generated novel resource allocation schemes for computer systems.

Task assignment. One very important special case of resource allocation is task assignment, that is, allocating the scarce time of actors to the tasks they will perform. An insight of the approach we are taking here, therefore, is that all the resource allocation methods listed in Table 1 are potentially applicable for task assignment, too.

For instance, in trying to imagine new coordination processes in a human organization, one might consider whether any given situation requiring task assignment could be better managed by managerial decision, by prior assignment according to task type, or by a pricing mechanism. To illustrate the surprising ideas this might lead to, consider Turoff's (1983) suggestion that employees within a large organization should be able to "bid" for the internal projects on which they wish to work, and that teams could be selected using these bids. There are obviously many factors to consider in determining whether such an arrangement would be desirable in a particular situation, but it is interesting to note that one potential disadvantage--the significantly greater communication required--would be much less important in a world with extensive computer networks.

2.2.2. *Managing producer / consumer relationships*

Another extremely common kind of relationship between activities is a "producer / consumer" relationship, that is, a situation where one activity produces something that is used by another activity. This relationship clearly occurs in all physical manufacturing processes, for instance, where the output of one step on an assembly line is the input to the next. It also occurs with information whenever one person in an organization uses information from another or when one part of a computer program uses information produced by another.

Producer / consumer relationships often lead to several kinds of dependencies:

(1) *Prerequisite constraints.* A very common dependency between a "producer" activity and a "consumer" activity is that the producer activity must be completed before the consumer activity can begin. When this dependency exists, there must at least be some *notification* process to indicate to the consumer activity that it can begin. For instance, when an automobile designer delivers a completed drawing of a part to the engineer who will design the manufacturing process for that part, the arrival of the drawing in the engineer's in-box "notifies" the engineer that her activity can begin.

Managing prerequisite dependencies also often involves explicit *sequencing* and *tracking* processes to be sure that producer activities have been completed before their results are needed. For instance, techniques from operations research, such as PERT charts and critical path methods, are often used in human organizations to help schedule large projects with multiple activities and complex prerequisite structures. These and other project tracking systems are also often used by managers to identify activities that are late and then use their authority to "motivate" the people responsible for the late tasks.

What alternatives can we imagine for managing this dependency? One possibility would be computer-based tracking systems that made it easy for everyone in the project to see status information about all the other activities and their dependencies. In this case, late tasks could be visible to everyone throughout the project, and "authoritarian" motivation by managers might become less important.

Sequencing 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 & Culler, 1986; Holt, 1988; Peterson, 1977; 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).

(2) *Transfer.* When one activity produces something that is used by another activity, the thing produced must be transferred from the "producer" activity to the "consumer" activity. Managing this dependency usually involves physical *transportation.* In this sense, physical transportation can be considered a coordination activity, since it involves managing a dependency between a "producer" activity and a "consumer" activity. When the thing transferred is information, we usually call the transfer *"communication",* rather than transportation.

In addition to simply transporting things, managing the transfer dependency also often involves storing things being transferred from one activity to another. For instance, one way of managing this aspect of the transfer dependency is to carefully control the timing of both activities so that items are delivered *"just in time"* to be used, and no storage is needed. This technique, for example, is becoming increasingly common in manufacturing environments (Schonberger, 1982; 1986). A more common approach is maintain an *inventory* of finished items, ready for the second activity to use, as a buffer between the two activities. Operations researchers, for instance, have developed techniques for determining at what stock levels and by how much to replenish an inventory in order to minimize costs (e.g., the *"economic order quantity"* (McClain, et al., 1992)).

Managing this dependency is also important in certain parts of computer science. For example in parallel processing systems, the rate of execution of processes must sometimes be regulated to ensure that the producer does not overwhelm the consumer or vice versa (e.g., Arvind, et al., 1986). As our framework would suggest, a common 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).

(3) *Usability.* Another, somewhat less obvious, dependency that must often be managed in a producer / consumer relationship is that whatever is produced should be usable by the activity that receives it. One common way of managing this dependency is by *standardization,* creating uniformly interchangeable outputs in a form that users already expect. This is the approach on assembly lines, for example. Another approach is to *ask users* what characteristics they want. For instance, in human organizations this might be

done by market research techniques such as surveys and focus groups (Kinnear & Taylor, 1991).

A third, related, alternative is *participatory design*, that is, having the users of a product actively participate in its design (Schuler & Namioka, 1993). This is a widely advocated approach to designing computer systems, for example, and it is interesting to note that the increasingly common practice of "concurrent engineering" (Carter & Baker, 1991) can also be viewed as a kind of "participatory design". In concurrent engineering, people who design a product do not simply hand the design "over the transom" to those who design its manufacturing process. Instead, they work together concurrently to create designs that can be manufactured more easily.

In computer systems, the usability dependency occurs whenever one part of a system must use information produced by another. In general, this dependency is managed by designing various kinds of interchange languages and other standards.

2.2.3. Managing simultaneity constraints.

Another common kind of dependency between activities is that they need to occur at the same time (or cannot occur at the same time). Whenever people schedule meetings, for instance, they must satisfy this constraint.

Another example of this constraint occurs in the design of computer systems in which multiple processes (i.e., instruction streams) can be executed simultaneously. (These systems may have multiple processors or a single processor which is shared between the processes.) In general, the instructions of the different processes can be executed in any order. Permitting this indeterminacy improves the performance of the system (e.g., one process can be executed while another waits for data to be input) but can cause problems when the processes must share data or resources. System designers must therefore provide mechanisms that restrict the possible orderings of the instructions by synchronizing the processes (that is, ensuring that particular instructions from different streams are executed at the same time) (Dubois, et al., 1988).

Synchronization primitives can be used to control sharing of data between a producer and consumer process to ensure that all data is used exactly once (the *producer/consumer* problem) or to prevent simultaneous writes to a shared data item (the *mutual exclusion* problem). For example, if two processes simultaneously read and then update the same data (adding a deposit to an account balance, say), one process might overwrite the value stored by the other.

One example of interdisciplinary transfer involving this concept is the work of Singh and colleagues in using computer science concepts about synchronized interactions to model process in human organizations (Singh & Rein, 1992).

2.2.4. Managing task / subtask dependencies

Top-down goal decomposition. A common kind of dependency among activities is that a group of activities are all "subtasks" for achieving some overall goal. As we discuss in more detail below, there is a sense in which some overall evaluation criteria or "goals" are necessarily implied by the definition of coordination. The most commonly analyzed case of managing this dependency 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 & 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 & 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., Chapman, 1987; Fikes & Nilsson, 1971; Allen, et al., 1990). 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, the steps do not necessarily 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. In human systems, this "bottom-up" process of goal selection can often engender more commitment from the actors involved than a top-down assignment of responsibility.

2.2.5. Managing other dependencies

As noted above, the dependencies discussed so far are only a suggestive list of common dependencies. We believe there are many more dependencies to be identified and analyzed. For instance, when two divisions of a company both deal with the same customer, there is a *shared reputation* dependency between their activities: what one division does affects the customer's perception of the company as a whole, including the other division. As another example, when several people in the same office want to buy a new rug, a key problem is not how to *allocate* the rug, but what color or other *characteristics* it should have. We might call this, therefore, a *shared characteristics* dependency.

More generally, there are many types of dependencies between objects in the world that are managed by coordination processes. For instance, an important part of managing the design of complex manufactured products involves managing the dependencies between different subcomponents. At first glance, our definition of coordination (as managing dependencies *between activities*) might appear to omit dependencies between objects that are not activities. We believe, however, that this focus has the advantage of greatly simplifying the analysis of a coordinated situation. In fact, it appears that all dependencies that require coordination can be treated this way. For example, dependencies between components matter because they, explicitly or implicitly, affect the performance of some activities (e.g., designing or redesigning the components) and they can, therefore, be viewed as a source of dependencies between those activities.

In general, as these examples illustrate, there may be many ways of describing different dependencies, coordination processes, and their relationships to each other (e.g., Crowston, 1991). We believe that there are many opportunities for further work along these lines.

2.2.6. Summary of basic coordination processes

Table 2 loosely summarizes our discussion so far by listing examples of how common coordination processes have been analyzed in different disciplines. The key point of this table, and indeed of much of our discussion, is that the concepts of coordination theory can 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.

2.3. Example: Analyzing the task assignment process

So far, the examples we have described have mostly involved a single field or analogies that have

Coordination process	*Computer Science*	*Economics and Operations Research*	*Organization Theory*
Managing shared resources (including task assignments)	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
Managing producer / consumer relationships (including prerequisites and usability constraints)	data flow and Petri net analyses	PERT charts, critical path methods; scheduling techniques	Participatory design; market research
Managing simultaneity constraints	synchronization techniques, mutual exclusion	scheduling techniques	meeting scheduling; certain kinds of process modeling
Managing task / subtask relationship	modularization techniques in programming; planning in artificial intelligence	economies of scale and scope	strategic planning; management by objectives; methods of grouping people into units

Table 2. Examples of how different disciplines have analyzed coordination processes

been transported from one discipline to another. To illustrate the possibilities for developing abstract theories of coordination that can apply simultaneously 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 & Richartz, 1967; Burton and Obel, 1980a). As we have described in more detail elsewhere (Malone, 1992), these analyses illustrate the kind of interdisciplinary interaction that our search for 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.

2.3.1. 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.

2.3.2. 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 led to high production costs.

2.3.3. *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., March & Simon, 1958; Galbraith, 1973; Williamson, 1985) and with major historical changes in the organizational forms of both human organizations (Chandler, 1962, 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 3.2 of this paper; Crowston, et al., 1987; Malone and Smith, 1988).

2.4. Other processes needed for coordination

In addition to the processes described above for managing specific dependencies, two other processes deserve specific attention: *group decision-making* and *communication*. It is sometimes possible to analyze these processes as ways of managing specific dependencies. For instance, communication can be viewed as a way of managing producer / consumer relationships for information. However, because of the importance of these two processes in almost all instances of coordination, we describe them separately here.

2.4.1. *Group decision-making*

Many coordination processes require making decisions that affect the activities of a group. For instance, in sharing resources a group must somehow "decide" how to allocate the resources; in managing task / subtask dependencies, a group must "decide" how to segment tasks. In all these cases, the alternative ways of making group decisions give rise to alternative coordination processes. For example, any group decision can, in principle, be made by authority (e.g., a "manager" decides), by voting, or by consensus (resulting from negotation).

Because of the importance of group decision-making in coordination, answers to questions about group decision making (e.g., Simon, 1976; Arrow, 1951) will be important for developing coordination theory. For instance, what are the decision-making biases in groups (e.g., Janis & Mann, 1977) as opposed to individuals (Kahneman & Tversky, 1973)? How do computer-based group decision-making tools affect these processes (e.g., Kraemer & King, 1988; Dennis, et al., 1988; Kiesler, et al., 1984)? Can we determine optimal ways of allocating tasks and sharing information for making group decisions (Miao, et al., 1992)? How do (or should) decision-making processes change in situations where both rapid response and high reliability are required (Roberts, et al., in press).

2.4.2. Communication

As with decision-making, there is a already a great deal of theory about communication, both from a technical point of view (e.g., Shannon & Weaver, 1949) and from an organizational point of view (e.g., Allen, 1977; Rogers & Agarwala-Rogers, 1976; Weick, 1969). One obvious way of generating new coordination processes, for example, is by considering alternative forms of communication (synchronous vs. asynchronous, paper vs. electronic) for all the places in a process where information needs to be transferred.

A coordination framework also highlights new aspects of these problems. For example, when we view communication as a way of managing producer / consumer relationships for information, we may be concerned about how to make the information "usable". How, for instance, 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 & Malone, 1990). The process by which standards are developed is also of concern to economists, philosophers, and others (e.g., Farrell & Saloner, 1985; Hirsch, 1987).

A related set of questions arises when we are concerned about how a group of actors can come to have "common knowledge," that is, they all know something, and they also all know *that* they all know it. There is a growing literature about this and related questions in fields as diverse as computer science, economics, and linguistics (Halpern, 1987; Aumann, 1976; Milgrom, 1981; Gray, 1978; Cohen & Levesque, 1991; Shoham, in press).

3. APPLYING A COORDINATION PERSPECTIVE

3.1. Approaches to analyzing coordination in different kinds of systems

Any scientific theory (indeed, any statement about the world) must neglect some things, in order to focus on others. For example, Kling (1980) describes how different perspectives (such as rational, structural, and political) on the use of information systems in organizations each illuminate aspects of reality neglected by the others. In some situations, one or another of these perspectives may be most important, and all of them are involved to some degree in any real situation. In applying coordination theory to any particular 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.

3.1.1. Parametric analysis vs. baseline analysis

There are at least two ways an interdisciplinary theory can help deal with differences like these among systems: (1) *parametric analysis*, and (2) *baseline analysis*.

Parametric 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. Examples of models that have been applied to more than one kind of system in this way are summarized in sections 2.3 and 3.4.2.

Baseline analysis. 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.

3.1.2. Identifying the components of coordination in a si:uation

In order to analyze a situation in terms of coordination, it is sometimes important to explicitly identify the components of coordination in that situation. According to our definition of coordination above, coordination means "managing dependencies between activities." Therefore, since activities must, in some sense, be performed by "actors," the definition implies that all instances of coordination include *actors* performing *activities* that are *interdependent.*[3] It is also often useful to identify *evaluation criteria* for judging how well the dependencies are being "managed." For example, we can often identify some overall "goals" of the activity (such as producing automobiles or printing a report) and other dimensions for evaluating how well those goals are being met (such as minimizing time or costs). Some coordination processes may be faster or more accurate than others, for instance, and the costs of more coordination are by no means always worthwhile.

It is important to realize that there is no single "right" way to identify these components of coordination in a situation. 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." As another example, when we talk about muscular coordination, we implicitly regard different parts of the same person's body as separate "actors" performing separate "activities".

[3] See Baligh & Burton, 1981; Baligh, 1986; Barnard, 1964; Malone, 1987; Malone and Smith, 1988; McGrath, 1984; Mintzberg, 1979 for related decompositions of coordination.

Conflicting goals. One important case of identifying evaluation criteria occurs when there are conflicting goals in a situation. In analyzing coordination in human organizations, it is often useful to simply ask people what their goals are and evaluate their behavior in terms of these criteria. However, some amount of *goal conflict* is nearly always present (e.g., Ciborra, 1987; Williamson, 1985; Schelling, 1960), and people may be unable or unwilling to accurately report their goals, anyway. To understand these situations, it is often useful to both try to identify the conflicting goals and also to analyze the behavior of the system in terms of some overall evaluation criteria. For instance, 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).

Another important example of conflicting goals occurs in market transactions: As we saw above, all participants in a market might have the goal of maximizing their own individual benefits, but we, as observers, can evaluate the market as a coordination mechanism in terms of how well it satisfies overall criteria such as maximizing consumer utilities (e.g., Debreu, 1959) or "fairly" distributing economic resources.

3.1.3. Preview of examples

In the remainder of this section, we describe 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. The early examples use very general notions of coordination; the later ones are more explicit in their identification of specific components of coordination.

This list is not intended to be a comprehensive list of all ways that theories of coordination could be applied. 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.

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

To illustrate how the explicit study of coordination might help with this endeavor, we begin with a very general argument that does not depend on any of the detailed analyses of coordination we have seen so far in this paper.[4] Instead, this argument starts with the simple observation that coordination is itself an activity that has costs. Even though there are many other forces that may affect the way coordination is performed in organizations and markets (e.g., global competition, national culture, government regulation, and interest rates), one important factor is clearly its cost, and that is the focus of this argument. In particular, 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, many

[4] See Malone (1992) and Malone and Rockart (1991) for more detailed versions of the argument in this section.

banks and insurance companies have substituted automated systems for large numbers of human clerks in their back offices. It has also 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). This prediction was not fulfilled for several decades after it was made, but many people believe that it finally began to happen with large numbers of middle management layoffs in the 1980's and 1990's.

(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, Malone and Lin, 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. For example, as noted above, information technology can facilitate what some observers (e.g., Mintzberg, 1979; Toffler, 1970) have called *adhocracies*. 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 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., Malone, et al., 1987; Lotus, 1989) may help make this communication more effective at much larger scales.

What might these new coordination-intensive structure be like? Let us consider recent work on two specific questions about the effects of information technology on organizations and markets: (1) How will IT affect the size of organizations? and (2) How will IT affect the degree of centralization of decision-making in organizations? This work does not focus explicitly on any specific dependencies. Instead, it compares two pairs of general coordination mechanisms that can manage many such dependencies: (1) market transactions vs. internal decision-making with firms, and (2) centralized vs. decentralized managerial decisions.

3.2.1. 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 (Williamson, 1985; Malone, Yates and Benjamin, 1987), 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. 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., in press).

If we extrapolate this trend to a possible long-run extreme, it leads us to speculate that we might see increasing use of "firms" containing only one person. For instance, Malone and Rockart (1991) suggest that there may someday be electronic 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 and then disband. Flexible arrangements like this might appeal especially to people who had a strong desire for autonomy--the freedom to choose their own hours and working situations.

3.2.2. Centralization of decision-making

Gurbaxani and 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 & McFarlan, 1987). Overall, this bidirectional trend for IT and centralization is consistent with empirical studies of this question (Attewell & Rule, 1984).

An alternative approach to this question is provided by (Danziger, et al., 1982). In a sense, this work can be considered a kind of "behavioral coordination theory". In studies of computerization decisions in 42 local governments in the U.S., they found that changes in centralization of power were not best explained any of the formal factors one might have expected. Instead, they found that since people who already have power influence computerization decisions, the new uses of computers tend to reinforce the existing power structure, increasing the power of those who already have it.

3.3. 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., Greif, 1988; Johansen, 1988; Ellis, et al., 1991; 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.

3.3.1. 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 (Nii, 1986; Erman, et al., 1980). 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 & 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, 1981; 1977). 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, using ideas about coordination from other disciplines does not provide any guarantee of developing useful cooperative work tools. However, 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.

3.3.2. 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 part of communication, 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

Process	*Example systems*
Managing shared resources (task assignment and prioritization)	Coordinator (Winograd and Flores, 1986) Information Lens (Malone, et al., 1987)
Managing producer / consumer relationships (sequencing prerequisites)	Polymer (Croft & Lefkowitz, 1988)
Managing simultaneity constraints (synchronizing)	Meeting scheduling tools (e.g., Beard, et al., 1990)
Managing task / subtask relationship (goal decomposition)	Polymer (Croft and Lefkowitz, 1988)
Group decision-making	gIBIS (Conklin and Begeman, 1988) Sibyl (Lee, 1990) electronic meeting rooms (e.g., Stefik, et al., 1987; Dennis, et al., 1988; DeSanctis & Gallupe, 1987)
Communication	Electronic mail, Computer conferencing (e.g., Lotus, 1989) electronic meeting rooms (e.g., Stefik, et al., 1987; Dennis, et al., 1988; DeSanctis and Gallupe, 1987) Information Lens (Malone, et al., 1987) collaborative authoring tools (e.g., Fish, et al., 1988; Ellis, et al., 1990; Neuwirth, et al., 1990)

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

assign tasks), but we have categorized the systems here according to the processes they explicitly emphasize.

Some of the systems also explicitly support several processes. For example, 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 2.3). 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.

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 Oval system (Malone, et al., 1992; Lai, et al., 1988).

3.3.3. Analyzing incentives for using cooperative work tools

Another use for coordination theory in designing cooperative work tools can be 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!

An interesting empirical illustration of this phenomenon occurred in a study of how one large consulting firm used the Lotus Notes group conferencing system. In this study, Orlikowski (Orlikowski, 1992) found that there were surprising inconsistencies between the intended uses of the system and the actual incentives in the organization. For instance, Orlikowski observed that this organization (like many others) was one in which people were rewarded for being the "expert" on something--for knowing things that others did not. Should we be surprised, therefore, that many people were reluctant to spend much effort putting the things they knew into a database where everyone else could easily see them?

These observations do not, of course, mean that conferencing systems like this one cannot be useful in organizations. What they do mean, however, is that we must sometimes be sensitive to very subtle issues about things like incentives and organizational culture in order to obtain the full benefits of such systems. For instance, it might be desirable in this organization to include, as part of an employee's performance appraisal, a record of how often their contributions to the Notes database were used by other people in the organization.

3.4. Designing distributed and parallel processing computer systems

Much recent activity in computer science has involved exploring 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.

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.

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

Competitive bidding markets for resource allocation. 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 (Smith and Davis, 1981; Davis & Smith, 1983) 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 2.3). 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., Stankovic, 1985; Kurose & Simha, 1989). 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 & 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 (Miller and Drexler, 1988; Drexler and Miller, 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" for information routing and resource allocation. Another central problem that arises in distributed and parallel processing systems is how and when to route 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 & Hewitt, 1981; Kornfeld, 1982). 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.

3.4.2. Analyzing stability properties of resource allocation algorithms

Another way of applying coordination concepts is to help evaluate alternative designs of distributed and parallel processing computer systems. For instance, Huberman and his colleagues (Huberman & Hogg, 1988; Lumer & 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).

3.5. Summary of applications

As summarized in Table 5, the examples we have described show how a coordination perspective 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.

Application area	*Examples of analyzing alternative designs*	*Examples of generating new design ideas*
Organizational structures and information technology	Analyzing the effects of decreasing coordination costs on firm size, centralization, and internal structure	Creating temporary "intellectual marketplaces" to solve specific problems.
Cooperative work tools	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
Distributed and parallel computer systems	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 a coordination perspective

4. RESEARCH AGENDA

We have seen how a number of different disciplines can contribute to answering the questions about coordination, 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?

As we suggested above, a central concern of coordination theory should be 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:

4.1. 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?

4.1.1. Characterizing dependencies

What kinds of dependencies are there? Are there ways to organize them that highlight common possibilities for managing them? Are some special cases of others? What causes dependencies? As we modify or alter a process, what techniques will be useful for keeping track of existing dependencies or identifying new ones? What techniques are useful for identifying dependencies in a field study of a particular process?

4.1.2. 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?

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

In analyzing alternatives 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?

4.3. 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 3, 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).

5. 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.

ACKNOWLEDGEMENTS

This work was supported, in part, by Digital Equipment Corporation, the National Science Foundation (Grant Nos. IRI-8805798 and IRI-8903034), and other sponsors of the MIT Center for Coordination Science.

Parts of this paper were included in three previous papers (Malone, 1988; Malone & Crowston, 1990; Malone & Crowston, 1991). We are especially grateful to Deborah Ancona, John Carroll, Michael Cohen, Randall Davis, Rob Kling, 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.

APPENDIX A: PREVIOUS 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 integration and harmonious adjustment of individual work efforts towards the accomplishment of a larger goal." (Singh, 1992)

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

"The act of working together" (Malone and Crowston, 1991)

APPENDIX B: RESULTS ABOUT COORDINATION FROM SELECTED FIELDS

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 appendix, 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.

Computer science

Sharing resources. Much research in computer science focuses on how to manage activities that share resources, such as processors, memory, and access to input/output devices (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., Hoare, 1975; Dijkstra, 1968). 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 & 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 Carrerio (1989) discuss three alternative ways of dividing parallel programs into units: according to the type of work to be done, according to the subparts of the final output, or simply according to which processor is available.

Managing information flows. Another important set of issues involves managing the flow of information. For instance, researchers in artificial intelligence and particularly in distributed artificial intelligence (DAI, e.g., Bond and Gasser, 1988; Huhns and Gasser, 1989) have used "blackboard architectures" to allow processes to share information without having to know precisely which other processes need it (Nii, 1986; Erman, et al., 1980), and "partial global plans" to allow actors to recognize when they need to exchange more information (Durfee & Lesser, 1987).

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). *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 & 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., Marschak & Radner, 1972; Hurwicz, 1973; 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).

Organization theory

Research in organization theory, drawing on disciplines such as sociology and psychology, focuses on how people coordinate their activities in formal organizations. A central theme in this work has involved analyzing general issues about coordination (e.g., Simon, 1976; March and Simon, 1958; Thompson, 1967; Galbraith, 1977; Lawrence and Lorsch, 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 (March and Simon, 1958; Lawrence and Lorsch, 1967). 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 (March and Simon, 1958; Galbraith, 1973; 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.

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 neighbor 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.

REFERENCES

Allen, J., Hendler, J. and Tate, A. (Eds.). (1990). *Readings in Planning* . San Mateo, CA: Morgan Kaufmann.

Allen, T. J. (1977). *Managing the Flow of Technology.* Cambridge, MA: MIT Press.

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.

Arvind, Nikhil, R. S. and Pingali, K. K. (1986). I-Structures: Data structures for parallel computing. In *Proceedings of the Graph Reduction Workshop.* Santa Fe, NM, October.

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, 4,* 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.

Bernstein, P. and Goodman, N. (1981). Concurrency control in distributed database systems. *ACM Computing Surveys, 13*(2), 185-221.

Bond, A. H. and Gasser, L. (Eds.). (1988). *Readings in Distributed Artificial Intelligence* . San Mateo, CA: Morgan Kaufman.

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. (in press). Does Information Technology Lead to Smaller Firms? *Management Science.*

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*, 401-417.

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.

Carter, D. E. and Baker, B. S. (1991). *Concurrent Engineering: The Product Development Environment for the 1990's*. Reading, MA: Addison-Wesley.

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.

Cohen, P. and Levesque, H. J. (1991). *Teamwork* (Technote No. 504). Menlo Park, CA: SRI International.

Conklin, J. and Begeman, M. L. (1988). gIBIS: A hypertext tool for exploratory policy discussion. In *Proceedings of CSCW '88 Conference on Computer-Supported Cooperative Work* (pp. 140–152). Portland, OR: ACM Press, Sept. 26-28.

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.* Ph.D. 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 *Proceedings of Conference on 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). Pennsylvania State University, State College, PA.

Dantzig, G. B. (1963). *Linear Programming and Extensions.* Princeton, NJ: Princeton University Press.

Danziger, J. N., Dutton, W. H., Kling, R. and Kraemer, K. L. (1982). *Computers and Politics: High Technology in American Local Governments.* New York: Columbia 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.

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.

Dertouzos, M. L. (1991). Building the Information Marketplace. *Technology Review, January,* 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.

Dubois, M., Scheurich, C. and Briggs, F. A. (1988). Synchronization, coherence, and event ordering in multiprocessors. *IEEE Computer, 21*(2), 9-21.

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).

Eccles, R. G. (1985). *The Transfer Pricing Problem: A Theory for Practice.* Lexington, MA: Lexington Books.

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, November 6-9.

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). Napa Valley, CA: North-Holland / Elsevier.

Ellis, C. A., Gibbs, S. J. and Rein, G. L. (1991). Groupware, some issues and experiences. *Communications of the ACM*, *34*(1), 38-57.

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, compatibiity, 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, March 23-25.

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.

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.

Gray, J. (1978). *Notes on data base operating systems* (Research Report RJ 2188). IBM.

Greif, I. (Eds.). (1988). *Computer Supported Cooperative Work* . Los Altos, CA: Morgan Kaufmann Publishers.

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, Sept. 26-28.

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.

Halstead, R. H. (1985). Multilisp: A language for concurrent symbolic computation. *ACM Transactions on Programming Languages and Systems*, *7*(4), 501-538.

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, March 25-27.

Hewitt, C. (1986). Offices are open systems. *ACM Transactions on Office Systems*, *4*(3), 271–287.

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. (1980). *Coordinator Programs* (Unpublished technical report). Wakefield, MA: Massachusetts Computer Associates, Inc.

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.

Holt, A. W., Ramsey, H. R. and Grimes, J. D. (1983). Coordination system technology as the basis for a programming environment. *Electrical Communication*, *57*(4), 307-314.

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.

Kinnear, T. C. and Taylor, J. R. (1991). *Marketing Research: An Applied Approach*. New York: McGraw Hill.

Kling, R. (1980). Social analyses of computing: Theoretical perspectives in recent empirical research. *ACM Computing Surveys*, *12*(1), 61–110.

Kohler, W. (1981). A survey of techniques for synchronization and recovery in decentralized computer systems. *ACM Computing Surveys, 13*(2), 149-183.

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. (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.

Lumer, E. and Huberman, B. A. (1990). *Dynamics of resource allocation in distributed systems*SSL-90-05). Palo Alto, CA: Xerox PARC.

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 (Ed.), *Cognition, Computation, and Cooperation* (pp. 56–83). Norwood, NJ: Ablex.

Malone, T. W. (1992). 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, October 7 - 10.

Malone, T. W., Crowston, K., Lee, J. and Pentland, B. (1993). Tools for inventing organizations: Toward a handbook of organizational processes. In *Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises.* Morgantown, WV, April 20-22.

Malone, T. W. and Crowston, K. G. (1991). *Toward an interdisciplinary theory of coordination* (Technical report #120). Cambridge, MA: Massachusetts Institute of Technology, Center for Coordination Science.

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., Lai, K.-Y. and Fry, C. (1992). Experiments with Oval: A radically tailorable tool for cooperative work. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW '92).* Toronto, Ontario, October 31 - November 4.

Malone, T. W. and Rockart, J. F. (1991). Computers, networks, and the corporation. *Scientific American, 265*(3 (Sept.)), 128-136.

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, October 7 - 10.

Marschak, J. and Radner, R. (1972). *Economic Theory of Teams.* New Haven: Yale University.

McClain, J., Thomas, L. J. and Mazola, J. (1992). *Operations Management* (3rd edition ed.). Englewood Cliffs, N.J.: Prentice Hall.

McGrath, J. E. (1984). *Groups: Interaction and Performance.* Englewood Cliffs, NJ: Prentice-Hall.

Miao, X., Luh, P. B. and Kleinman, D. L. (1992). A normative-descriptive approach to hierarchical team resource allocation. *IEEE Transactions on Systems, Man and Cybernetics, 22*(3), 482-497.

Milgrom, P. (1981). An axiomatic characterization of common knowledge. *Econometrica, 49*(1), 17-26.

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.

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.

Neuwirth, C. M., Kaufer, D. S., Chandhok, R. and Morris, J. H. (1990). Issues in the design of computer-support for co-authoring and commenting. In *Proceedings of the Third Conference on Computer Supported Cooperative Work (CSCW 90)* (pp. 183-195). Los Angeles, CA: ACM Press, October 7 - 10.

Nii, P. (1986). The blackboard model of problem solving. *AI Magazine*, (Spring), 38–53.

NSF. (1991). *Coordination Theory and Collaboration Technology Workshop Summary (June 3-5, 1991).* Washington, D.C.: National Science Foundation.

NSF-IRIS. (1989). *A report by the NSF-IRIS Review Panel for Research on Coordination Theory and Technology.* Available from NSF Forms & Publications Unit.

Orlikowski, W. (1992). Learning from Notes: Organizational Issues in Groupware Implementation. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW 92).* Toronto, Canada: ACM, November.

Peterson, D. (Eds.). (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 Assocation 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 fur Grundlagen der Planung I.A.

Roberts, K. H. and Gargano, G. (1989). Managing a high reliability organization: A case for interdependence. In M. A. V. Glinow and S. Mohrmon (Ed.), *Managing complexity in high technology industries: Systems and people* (pp. 147-159). New York: Oxford University Press.

Roberts, K. H., Stout, S., K. and Halpern, J. J. (in press). Decision dynamics in two high reliability military organizations. *Management Science*, .

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.

Rumelhart, 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.

Schonberger, R. (1982). *Japanese Manufacturing Techniques.* New York: Free Press.

Schonberger, R. (1986). *World Class Manufacturing.* New York: Free Press.

Schuler, D. and Namioka, A. (Eds.). (1993). *Participatory Design: Principles and Practices* . Hillsdale, N. J.: Lawrence Erlbaum.

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.

Shoham, Y. (in press). Agent Oriented Programming. *Journal of Artificial Intelligence*, .

Simon, H. A. (1976). *Administrative Behavior* (3rd ed.). New York: Free Press.

Simon, H. A. (1981). *Sciences of the Artificial* (2nd Edition ed.). Cambridge, MA: MIT Press.

Singh, B. (1992). *Interconnected Roles (IR): A coordination model* (Technical Report No. CT-084-92). Austin, TX: Microelectronics and Computer Technology Corp. (MCC).

Singh, B. and Rein, G. L. (1992). *Role Interaction Nets (RIN): A process description formalism* (Technical Report No. CT-083-92). Austin, TX: MCC.

Smith, A. (1776). *The Wealth of Nations* (1986 edition ed.). London: Penguin Books.

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 applicaiton of Bayesian decision theory to decentralized control of job scheduling. *IEEE Transactions on Computers, C-34*(2), 117-130.

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.

Stoddard, D. (1986). *OTISLINE* (Case 9-186-304). Harvard Business School.

Tannenbaum, A. S. (1981). *Computer Networks*. Englewood Cliffs, NJ: Prentice-Hall.

Tatar, D. (Eds.). (1988). *Proceedings of the Second Conference on Computer-supported Cooperative Work* . Portland, OR: ACM Press.

Tatar, D. (Eds.). (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.

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.

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.