Explicit vs. Implicit Coordination Mechanisms and Task Dependencies:
One Size Does Not Fit All

Alberto Espinosa
Javier Lerch
Robert Kraut

Carnegie Mellon University

January 2002
Forth Coming in

Team Cognition: Process and Performance at the Inter- and Intra-individual Level
Eduardo Salas, Stephen M. Fiore, and Janis A. Cannon-Bowers (Editors)

Team Cognition, Task Dependencies and Coordination: One Size Does Not Fit All

<u>Introduction</u>

Teams are important units of organizational work (Hackman 1987, Sproull & Kiesler 1991). They bring diverse expertise, skills and resources to complex tasks that may be too large or complex for a single individual to undertake. However, as projects and teams grow in size and complexity, tasks and member dependencies become more numerous, diverse and complex, thus increasing the need for team coordination. Coordinated teams manage these dependencies effectively using a number of explicit and implicit mechanisms and processes. Teams coordinate explicitly using task programming mechanisms (e.g., schedules, plans, procedures, etc.) or by communicating (e.g., orally, in writing, formally, informally, interpersonally, in groups). We call these mechanisms “explicit” because team members use them purposely to coordinate. However, teams can also coordinate “implicitly” (i.e., without consciously trying to coordinate) through team cognition, which is based on shared knowledge team members have about the task and about each other. This shared knowledge helps team members understand what is going on with the task, and also helps them anticipate what is going to happen next, and which actions team members are likely to take, thus helping them become more coordinated.

In this chapter we discuss the interplay between explicit and implicit coordination mechanisms and how they jointly affect team coordination and performance. Because dependencies can be managed in more than one way, teams will make decisions on which explicit coordination mechanisms to use. However, the use of such explicit mechanisms both, influence (e.g., through team interaction, use of common documents, tools and procedures, etc.) and are influenced by the existing level of team cognition (e.g., members may not communicate as often once they know what to expect from each other). Consequently, it is important that we understand how explicit and implicit coordination mechanisms complement and interact with each other. We also discuss in this chapter the need to understand the nature of the multiple dependencies involved in a task in order to figure out which mechanism can be more effective in helping teams coordinate. Different mechanisms, whether explicit or implicit, will have varying degrees of effectiveness for different tasks and for different stages of a given task. For example team communication may be very important for: (a) complex intellective tasks in which task dependencies are somewhat uncertain; and (b) early stages of other tasks when team members don't know much about each other or the task. On the other hand, team communication may not be so important for: (c) more mechanical tasks (e.g., assembly line) in which dependencies are more predictable and their management can be programmed; and (d) later stages of other tasks when team members know each other well or have implemented effective division of labor.

Throughout our discussions, we focus on asynchronous (i.e., non real-time) and geographically dispersed task contexts. This is an important distinction because most of the literature on team cognition has focused on synchronous (i.e., same-time) and co-located teams (e.g., flight crews, medical emergency units, etc.). However, work arrangements and the nature of the resulting dependencies will vary substantially depending on whether the team is separated by time or by distance. Thus, our focus on different-time and different-place contexts provides a unique contribution to the team cognition literature, and a good complement to the existing theories and empirical evidence.

In the following section we describe a unified framework of team coordination that incorporates both, explicit and implicit coordination mechanisms. The next section discusses the need to identify key dependencies and coordination types for a given task, and to investigate which are the most effective mechanisms to manage them. The final section summarizes our conclusions. Throughout our discussion, we present examples and results from our empirical studies with: (1) decision [asynchronous] teams who managed simulated companies for Carnegie Mellon's Management Game course; and (2) large-scale [dispersed] software development teams from a Fortune 500 telecommunications company. We refer in this chapter to the first teams as the "decision teams" and the second ones as the "software teams". More complete descriptions of these two studies can be found in appendices A and B respectively.

Theoretical Framework

In this section we propose and describe a theoretical framework to study the effects of team cognition on team coordination and performance. This framework has three important properties. First, it follows an input-process-output (IPO) model, which has been suggested by many for the study of teams (Kraemer & Pinsonneault 1990, McGrath 1991, McGrath & Hollingshead 1994). Input variables in this framework represent task, team and context factors that give rise to different work arrangements and dependencies. Teams manage these dependencies using mix of coordination mechanisms. The extent to which these dependencies are effectively managed will influence the team's state of coordination, which may in turn affect team performance. Second, the framework includes both, explicit and implicit coordination mechanisms, which, as we discuss later on, need to be modeled jointly because they may complement, affect or interact with each other in their effect on team coordination. Finally, the framework ties each of its elements to how task dependencies originate (i.e., the need to coordinate) and how these dependencies are managed (i.e., team coordination). We discuss the first two properties of the framework in this section as we provide the theoretical foundations for the framework. We begin by defining coordination, which is the central piece of the framework.

We then proceed to develop the framework in a backward fashion, starting with the effect of coordination on team performance. We then discuss how implicit and explicit coordination mechanisms affect team coordination. Next we discuss how input variables (i.e., task, team and context) influence the mix of coordination mechanisms employed by a team. We then put together all the building blocks to illustrate the integrated framework. We discuss the third property in the following section in which we relate the different elements of the framework to the underlying task dependencies.

Coordination Defined: Managing Dependencies

We draw from the research literature on coordination theory to define coordination as the effective management of dependencies among sub-tasks, resources (e.g., equipment, tools, etc.) and people (Malone & Crowston 1990, 1994). If things can be done independently, then there is no need for coordination. Conversely, when multiple individuals, sub-tasks and resources need to interact in a synchronized fashion to carry out a joint task, it gives rise to dependencies among them. Team members can perform their individual responsibilities competently and still be very uncoordinated with the rest of the team if the respective dependencies among their sub-tasks are not properly managed. We use this view of coordination from the perspective of managing dependencies throughout this chapter to explain how different elements of the proposed framework fit together. Furthermore, this definition of coordination is very useful when conducting research because it helps define concepts and develop measures. For example, coordination can be viewed as both, a process (i.e., coordinating) and an outcome (i.e., state of coordination). The process of "coordinating" can be defined as the activities carried out by team members when managing dependencies. For example, when a software team convenes a debriefing meeting so that all parties can share the status of their respective tasks, they are managing the dependencies involved in developing software as a team by communicating with each other about what is going on with their individual task assignments. Coordination as an outcome (i.e., state of coordination or coordination success) can be defined as the extent to which dependencies have been effectively managed. For example, a software team will be highly coordinated if all relevant software parts for a given project integrate and work well together, are delivered on schedule, and are produced according to the established software process (i.e., technical, temporal and software process dependencies have been effectively managed).

Coordination: An Antecedent of Team Performance

Performance is an outcome of utmost interest in team research. Team coordination is useful to the extent that it leads to team performance, but coordination is not always important for performance. For example, coordination may not be so important for team problem-solving tasks in which a "eureka" type solution exists, which is evident to all members when one member figures it out. If members of a team can work independently and if there are no task dependencies, then there is nothing to coordinate and, consequently, coordination will not affect performance (Malone et al. 1994, Thompson 1967, VanDeVen et al. 1976). In contrast, more complex tasks with substantial dependencies can indeed benefit from coordination. However, even if coordination is necessary for performance, it may not always be sufficient to ensure performance. First of all, some task dependencies are more critical to performance than others. A team may be doing a great job at managing task dependencies that are not so important for task performance (e.g., producing software in a timely manner) while not paying much attention to other dependencies that are very important for task performance (e.g., delivering high quality and error-free software for a mission-critical application). For example, in the study with software teams some technical and project managers described situations in which they had implemented software product features in a very effective and productive manner, to find out later on that these features had been removed from the product because they were not what the client wanted or what the market needed. Similar "feature churn" problems have been documented in other studies (Crowston & Kammerer 1998). Similarly, in our study with decision teams we found that the effect of task coordination (e.g., avoiding duplication of work, having clear assignment of tasks, etc.) on performance is indirect, mediated by sub-strategy coordination (e.g., finance strategy well coordinated with production strategy). Second, there may be other antecedents of performance that are unrelated to coordination, which may hinder or enhance performance. For example, in our study with decision teams we found some evidence that strategy coordination has a positive effect on team performance, but it only explains a small percentage of variance. Instructors of the Management Game simulation course indicated that in addition to having coordinated strategies, successful teams also have sound functional sub-strategies (i.e., finance, marketing and operations) and a good understanding of the competition. This relationship between coordination and performance is illustrated in Figure 1.

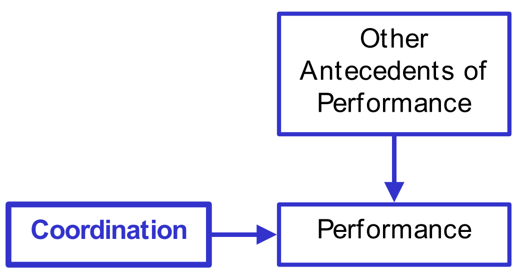

Figure 1
Coordination and Performance

Explicit Coordination Mechanisms: The Classic Organizational Theory View

As we discussed above, a highly coordinated team is one in which task dependencies have been managed effectively. The management of these dependencies is accomplished via coordination mechanisms. Thus, a coordination mechanism can be defined as one that helps teams manage dependencies. For example, simple things in our daily lives like a traffic light and a flight schedule can be viewed as coordination mechanisms that help us manage our dependencies with other drivers and the airlines, respectively. Consequently, explicit coordination mechanisms (or processes) can be defined as those mechanisms (or processes) explicitly employed by a team to help manage task dependencies. Explicit coordination mechanisms and processes have been studied in the classical organizational research literature for several years. This literature suggests that team coordinate explicitly by using task organization mechanisms or by communicating. March and Simon suggested that teams use task organization mechanisms (i.e., "task programming") for the most routine aspects of the task because the respective dependencies are more predictable, and thus can be more easily managed in a programmed way (March & Simon 1958). Others have used different terms for this type of coordination like "impersonal mechanisms" (VanDeVen et al. 1976) and "administrative coordination" (Faraj & Sproull 2000). Examples of these mechanisms include thinks like division of labor, tools, schedules, plans, manuals and specifications.

When routines change, when routines are no longer applicable to the task, or when the task has very little or no routine aspects, task organization mechanisms are less effective because dependencies can no longer be managed in a programmed way. March and Simon suggested that teams resort to communication (i.e., "coordination by feedback") in such cases. For example, a software project schedule may be rendered useless if several delivery deadlines have already been missed or if there has been a crisis (e.g., a major hardware failure) that calls for re-scheduling. When these things happen the team needs to communicate to cope with the changing situation, or to adopt new task organization mechanism more suitable to the new situation. The literature has also used different terms for this type of coordination like coordination by "mutual adjustment" (Thompson 1967) and "personal" and "group" mechanisms (VanDeVen et al. 1976). As the last two terms imply, coordination by communication can be interpersonal or in groups, but it can also be formal or informal. For example, an empirical study with software teams at an organization found that teams not only coordinate by communicating formally via meetings and documents, but that they also do a substantial amount of coordination via more informal communication when team members spontaneously encounter each other in public places like cafeterias, coffee rooms, and hallways (Kraut & Streeter 1995).

This discussion underscores the fact that different coordination mechanisms may be more suitable for different tasks. Furthermore, the same task may require the use of different coordination mechanisms over time. Studies conducted at Carnegie Mellon have analyzed software student teams developing small applications for external clients and found that these teams changed the mix of coordination mechanisms they used over time (Kiesler et al. 1994, Wholey et al. 1996). For example, the study found that teams communicate more intensely at the beginning. Interestingly, the study also found that successful teams only communicated moderately towards the end. Unsuccessful novice teams (i.e., from introductory course) communicated too little while unsuccessful expert teams (i.e., from follow up course) communicated too much. Similarly, the study found that division of labor was an effective task organization mechanism only towards the end of the task, once team members knew each other's skills well. This traditional view of coordination based on explicit mechanisms is depicted in Figure 2.

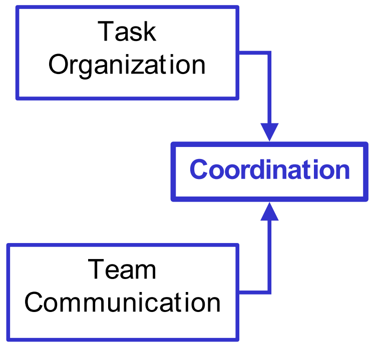

Figure 2
Explicit Coordination Mechanisms

Implicit Coordination Mechanisms: A More Recent View from Team Cognition Research

More recent research in team cognition suggests that as team members interact with each other and gain expertise with the joint task, they develop knowledge about the task and the team, which helps them coordinate implicitly (Cannon-Bowers et al. 1993, Klimoski & Mohammed 1994, Levesque et al. 2001). Such implicit coordination has been referred to as the "synchronization of member actions based on unspoken assumptions about what others in the group are likely to do" (Wittenbaum & Stasser 1996). Following these concepts and our prior definition of coordination, we define implicit coordination mechanisms as those mechanisms that are available to team members from shared cognition, which enable them to explain and anticipate task states and member actions, thus helping them manage task dependencies. We refer to these mechanisms as implicit because they are available to the team in the form of shared cognition and as such they are not consciously employed for the purpose of coordinating.

There are a number of different streams of research that have studied these implicit coordination mechanisms. For example, a recent survey study on shared cognition research (Cannon-Bowers & Salas 2001) identified over 20 team cognition labels and constructs described and studied in this literature, including shared (or team) mental models (Cannon-

Bowers et al. 1993, Klimoski et al. 1994, Kraiger & Wenzel 1997, Rouse & Morris 1986), team situation awareness (Endsley 1995, Wellens 1993), transactive memory (Liang et al. 1995, Wegner 1986), and collective mind (Weick & Roberts 1993). These constructs are conceptually distinct, but they all share the commonality that some form of knowledge similarity or complement helps team members explain other members' actions, understand what is going on with the task, and develop accurate expectations about future member actions and task states, thus helping them coordinate implicitly. For example, the effect of shared mental models is evident in real-time teams performing in high-paced contexts like sports competitions and medical emergency rooms in which members act in a highly coordinated fashion with very little communication because of their prior experience working and/or training together. These models are perhaps more evident when absent because activities become noticeably uncoordinated, sometimes even causing accidents (Helmreich 1997, Weick 1990, 1993, Weick et al. 1993). While the lack of shared mental models in asynchronous and/or distributed collaborative work like software development or management decision making may not necessarily lead to accidents, it can lead to uncoordinated activity, low productivity and substantial financial losses due to things like duplicate work, missed deadlines, misunderstandings and priority conflicts.

Team cognition research has focused primarily on developing theoretical foundations, but there has been a paucity of empirical studies. However, this is beginning to change. For example, a recent study of dyads working in a flight simulation task found that shared mental models have a positive, but indirect effect on team performance, mediated by team process (e.g., coordination) (Mathieu et al. 2000). Also, most of the research in shared cognition has focused on real-time and co-located teams. Because time and distance separation provides fewer opportunities for interaction and acquisition of shared knowledge, team cognition is much more difficult to develop in asynchronous and geographically dispersed environments, thus the importance of conducting research in these contexts. Fortunately, new research is beginning to emerge in these contexts. For example, a recent study of large-scale software developers found that "administrative" (i.e., explicit) coordination positively affects team performance, but it also found that knowing where expertise is located in the team has a positive and significant effect on team effectiveness and efficiency, above and beyond the effects of administrative coordination (Faraj et al. 2000). Another study with consulting teams composed of MBA students working on three-month long projects in actual client organizations found preliminary evidence that transactive memory is positively correlated with team coordination (Lewis 2000). Yet another study with software requirements analysis teams at two separate organizations found that explicit management of dependencies helps understand why teams are coordinated, but only up to a certain point, and that teams that exhibit a "collective mind" (Weick et al. 1993) tend to be more coordinated (Crowston et al. 1998).

Our research with decision teams has also found that shared mental models have a positive effect on task coordination (i.e., management of task dependencies—e.g., avoiding duplication of work, tasks are clearly assigned, etc.) and strategy coordination (i.e., management of functional strategy dependencies—e.g., finance, operations and marketing sub-strategies) (Espinosa et al. 2001b). A related qualitative study with two distributed decision teams (i.e., students from a Mexican university and from Carnegie Mellon working in the same team) suggested that these teams were uncoordinated at the beginning, but that they became progressively more coordinated over time as got to know the task and each other better (i.e., students from each site visited the other site at two times during the semester and also had video conference meetings). Qualitative analysis from our research with software teams has also found evidence that organized shared knowledge (i.e., shared mental models) of key concepts, processes and products, knowing who knows what in the team (i.e., transactive memory), and knowing what is happening with the task (i.e., task awareness) and who is around (i.e., presence awareness) help teams coordinate, particularly when teams are geographically distributed (Espinosa et al. 2000).

In sum, as team members develop experience with the task and interact with each other they develop implicit coordination mechanisms based on team cognition that help them coordinate implicitly. Consequently, the use of task organization mechanisms and team communication will affect how this team cognition develops. In turn, as we discussed before, the strength and type of team implicit coordination mechanism developed will influence which task organization mechanisms used by the team (and how they are used) and how they communicate. This shared cognition view of coordination through implicit mechanisms is depicted in Figure 3.

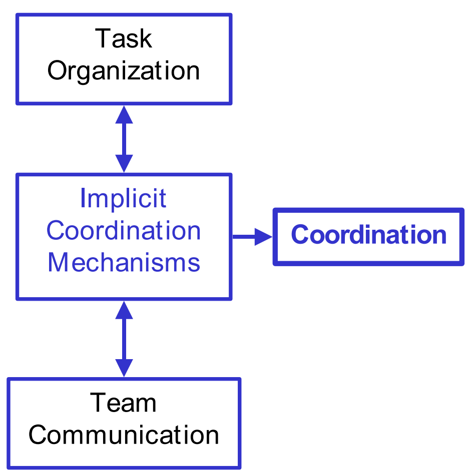

Figure 3
Implicit Coordination Mechanisms

Input Variables: Effect on Mix of Implicit and Explicit Coordination Mechanisms

As the discussion in this section suggests, a team will employ a mix of coordination mechanisms that deems most suitable to manage dependencies present in the task. Task, team and context variables will influence the number, complexity and types of dependencies that will be present in the task, and consequently, which coordination mechanisms will be used by team members to manage these dependencies. For example, decision and software teams in our studies faced different types of dependencies. Decision teams had sub-task dependencies (e.g., avoiding duplication of work, sub-task assignments) and functional sub-strategy dependencies (e.g., synchronizing finance and operations strategies), whereas software teams faced technical (i.e., software parts working well together), temporal (i.e., parallel activities completed on time) and software process dependencies (i.e., activities carried out according to the formal process). Team variables will also influence the types of dependencies present. For example, larger teams are likely to have more numerous and complex dependencies than smaller teams. Also, teams with a long history working together will be more experienced at managing team member dependencies. Finally, context variables will also influence the types of dependencies present in the task. Things like technology and organizational processes can impose structure and restrictions that affect how teams interact and carry out their tasks. For example, teams that are co-located are likely to use a different mix of coordination mechanisms than teams who are geographically distributed whose interaction is mediated by communication technologies. Similarly, teams working on real-time contexts (e.g., flying an airplane, operating a patient) are likely to use a different mix of coordination mechanisms than teams working on asynchronous tasks (e.g., strategic planning, large-scale software development). How input variables change the nature of dependencies is discussed in more detail in the next section of this chapter.

An Integrated Framework: Explicit and Implicit Coordination

Our discussion up to this point suggests the integration of the classical organizational view of team coordination (i.e., explicit coordination) with newer views from team cognition research (i.e., implicit coordination). As discussed in the next section, every task is different, and the nature of dependencies present will influence which mix of coordination mechanisms can better help teams coordinate. Teams in familiar and routine tasks and larger teams may benefit more from task organization mechanisms (i.e., plans, schedules, programs, tools, etc.). Teams in less familiar and unstable conditions and smaller teams may be more effective at coordinating via communication. In addition, as the task progresses and as team members interact over time, their team cognition will gradually become stronger and, while team members may continue to use some explicit coordination mechanisms, they will probably begin to substitute other explicit mechanisms with implicit mechanisms. Explicit mechanisms that consume more time and effort (e.g., formal meetings, writing reports) are more likely to be substituted, particularly when under time pressure.

Consequently, studies of team coordination focusing on explicit mechanisms need to take into account the team's use of implicit coordination mechanisms. Furthermore, the interplay between explicit and implicit coordination mechanisms may give rise to interesting complementary and interaction effects. For example, the effect that a new software version management system may have on coordination in a large-scale software development project may be different for a new team than for one who has been working together in the task domain for several years. In fact, the introduction of a new CASE tool for software development at two different sites for the software teams we studied had noticeably different effects. One site had a long history of working together with little turnover, and had developed paradigms and methods deeply rooted on tradition. This group had severe difficulties adapting to the new CASE tool and many commented that it did not help their coordination very much. The other site had only been in existence for approximately three years and had a high turnover rate, and many of the people interviewed welcomed the implementation of new tools and methods because it helped them coordinate their work.

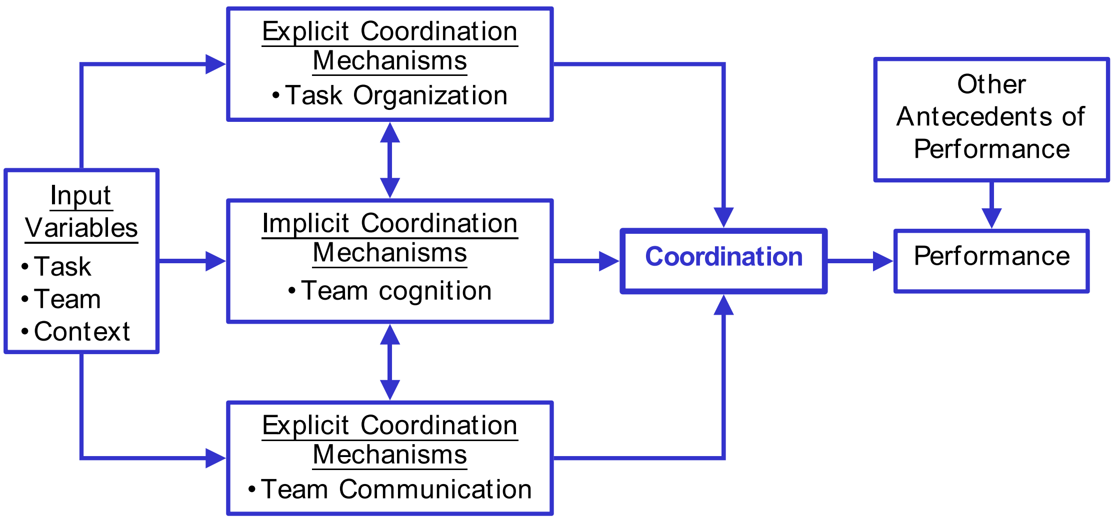

Figure 4

An Integrated Framework of Team Coordination and Performance

Similarly, studies of team coordination focusing on team cognition and implicit coordination mechanisms need to take into account the team's use of explicit coordination mechanisms. Since the task organization mechanisms and team communication processes used by the team affect coordination, the use of such mechanisms needs to be considered. For example, a software team that does not use a configuration management system for software version control may need to rely more on shared task knowledge than a team that does. In sum, this section suggests the use of an integrated framework of team coordination that incorporates both, implicit and explicit coordination mechanisms. This integrated framework is illustrated in

Figure 4. It contains inputs, processes and output (IPO), as suggested in the team research literature (Kraemer et al. 1990, McGrath 1991, McGrath et al. 1994).

Different Types of Dependencies and Coordination: One Size Does Not Fit All

In a nutshell, the framework proposed above suggests that task, team and context variables influence the nature of dependencies present in a task, which in turn affect the mix of coordination mechanisms used by the team to effectively manage these dependencies. To the extent that these dependencies are important for performance, coordination will affect team performance. However, as suggested in other studies, figuring out which coordination mechanisms are most effective in managing the various dependencies requires an understanding of these dependencies (Crowston et al. 1998). While some argue that shared mental models or transactive memory may improve coordination and performance, we argue that there may be important complementarities, tradeoffs and interactions between implicit and explicit mechanisms. Consequently, if key dependencies can be managed effectively with explicit coordination mechanisms alone, then team cognition may not improve coordination. For example, configuration management systems (i.e., version control systems) in large-scale software projects use a library metaphor in which software is "checked out", updated and then "checked in" thus facilitating parallel development by multiple programmers, who may not even know each other. Modern configuration management systems automatically protect all parts of the software code that are affected by the software part that was checked out so that developers don't interfere with each other, and also contain features to manage workflow (Grinter et al. 1999). Such systems reduce the need for communication and team cognition to coordinate small modifications to the software. Larger modifications, on the other hand, may have more complex dependencies that can benefit from more communication and stronger team cognition.

This discussion illustrates the importance of understanding the nature of the dependencies involved in a task before we can truly learn which mechanisms help teams coordinate and perform. This is particularly important for complex tasks like management decision and large-scale software development in which dependencies are numerous and complex. For example, our study with software teams found that there are three main types of coordination problems in this domain, stemming from the specific dependencies involved: technical, temporal and software process. Technical coordination problems arise because of inadequately management of technical dependencies among software parts that need to work well together when integrated. Temporal coordination problems arise because temporal dependencies are inadequate managed when software parts, or related sub-tasks, are not completed according to schedule in a synchronized manner. Software process coordination problems arise when process dependencies are inadequately managed because of team members who don't adhere to the established software process or to procedures agreed upon at meetings.

One of the interesting things that we found in this study was that software developers and other technical staff (e.g., testing engineers, integrators, etc.) were more concerned with technical coordination because they were severely affected when technical dependencies among software parts were not properly addressed in the software code since it led to errors, repairs and re-work. These groups seemed to rely on implicit coordination mechanisms like shared knowledge of the software product being developed and of the specific telecommunications function being implemented, and on task organization mechanisms like software development tools, technical web sites and the software configuration management system. Formal meetings and other related communication by technical groups tended to be brief, informational, and to the point. Interestingly, further analysis of software production data in our studies suggests that technical coordination and software process coordination are associated with reduced time intervals for small modifications, but that temporal coordination is not. Conversely, results from interviews with managers suggest that time intervals for integration of larger subsystems are severely affected by process and temporal coordination. Project and technical managers were more concerned with process and temporal coordination because the dependencies that they had to manage had to do with meeting milestones and delivering software features to clients on time. These groups seemed to rely more on implicit coordination mechanisms like shared knowledge of (and close attention to) software processes and plans, and task organization mechanisms like schedules, shared databases and plans. Also, these groups had more active and frequent communication via formal meetings, discussion boards, and electronic mail.

Because of the importance of understanding dependencies when studying team cognition and coordination, the framework previously presented in Figure 4 has been slightly re-formulated in Figure 5 to better illustrate the relationship between dependencies, coordination and performance, and to bring attention to the fact that different coordination mechanisms will have different effects on different types of coordination, which in turn will have different effects on team performance—i.e., ***"one size does not fit all"***. The following sub-sections discuss the relationship between dependencies and the different elements of the framework. Once again, we start the discussion with the dependent variable of interest (i.e., team performance) and proceed backwards through the framework.

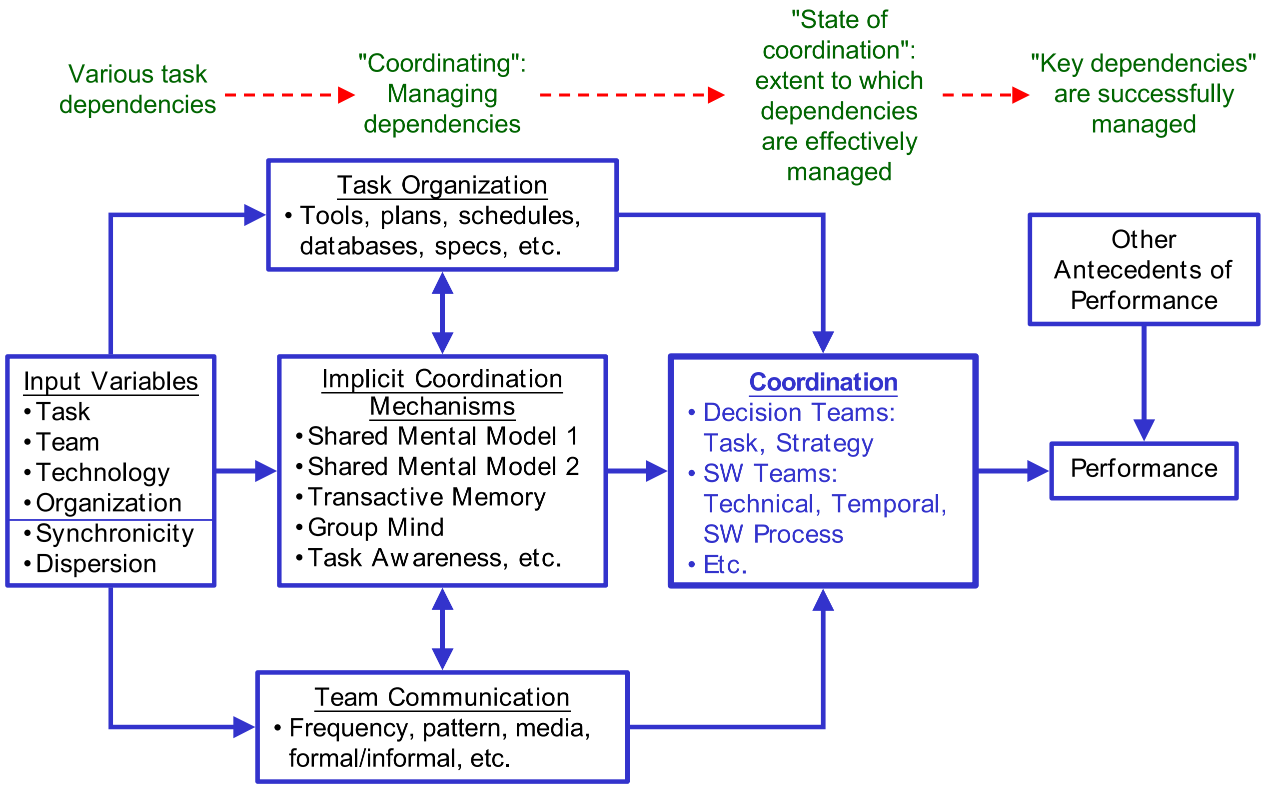

Figure 5

Dependencies, Team Coordination and Performance

Effect of (which type of) Coordination on Performance: Key Dependencies Well Managed

If the purpose of coordination mechanisms is to improve coordination to achieve higher levels of team performance, it is imperative that we understand, not only which dependencies are present in the task, but also which ones are more critical to performance. Teams may be highly coordinated and still perform poorly. This could be because of two reasons. First, there may be other antecedents of performance that are unrelated to coordination. For example, a well-coordinated software team that does not know very well the application domain for which they are writing software will end up developing poor quality software (Curtis et al. 1988). Second, some dependencies may be more important to performance than others and, while many dependencies may be adequately managed, some of the key dependencies may not be managed well. In other words, the team may be coordinated on aspects of the task that are not that important for performance. For example, a software team with high technical and temporal coordination will deliver error free software on time, but will not be an effective team if important software features needed by clients are not being properly implemented (i.e., dependencies with clients are not managed effectively). In other words, the team may be delivering timely and error free software that is useless to the client.

For example, our research with decision teams has shown that teams with poor task coordination don't perform well (i.e., poor financial performance and low board evaluations), but that good task coordination is not sufficient for performance. In other words, managing task dependencies (e.g., avoiding duplication of work, having clear work assignments, etc.) is important, but it is not critical to performance. On the other hand, we did find that good strategy coordination was a strong predictor of team performance. Because the decision task involved in this study required the successful integration of functional sub-strategies (i.e., finance, marketing, operations) into a cohesive and sound overall company strategy, managing dependencies effectively among theses sub-strategies was important to performance. Nevertheless, the performance variance explained by strategy coordination was small, suggesting that other types of coordination and or other antecedents of performance (e.g., sound functional strategies, good understanding of the competition) were important too. Furthermore, there may be different aspects of performance, which may be affected differently by different types of coordination. For example, in our studies with decision teams, we found that strategy coordination affects financial performance (e.g., profits, rate of return on investment, stock price) but it did not directly affect performance measured by external board evaluation. The effect of strategy coordination on board evaluation was mediated by financial performance.

Effect of Coordination Mechanisms on Coordination: Managing Dependencies Effectively

As discussed before, complex tasks have multiple dependencies, and teams have a choice of coordination mechanisms to help them manage them. Managing these different types of dependencies represent different types of coordination required for the task. For example, the management of technical dependencies among different software modules represents the technical coordination necessary for these modules need to have to interoperate properly. Different mechanisms will have varying degrees of effectiveness in helping manage the various types of dependencies present. Consequently, well-coordinated teams will not necessarily be those that have strong team cognition, but those who find an effective mix of mechanisms for the coordination needs of the task they are engaged in. As we discussed, and as suggested in the organizational literature, task organization mechanisms will be generally preferred for routine aspects of the tasks, and for larger teams in which it is more difficult to communicate (VanDeVen et al. 1976). Team communication will be generally preferred for tasks without many routine aspects, or for tasks in which routines often change, and also for smaller teams in which it is easier to communicate (VanDeVen et al. 1976). At the same time, teams with prior histories working together, in similar tasks, and/or in similar contexts, will have stronger team cognition and thus more implicit coordination mechanisms available to them, which will reduce their reliance on explicit coordination mechanisms. Consequently, when we study the effect of team cognition on team coordination, it is important that we ask and find answers to these questions: (1) Which specific type of team cognition are we studying? (2) Which types of coordination are involved? And, more importantly, (3) which other types of explicit coordination mechanisms are being used by the team?

The answer to these questions lies in a thorough understanding of how to best manage task dependencies. Consistent with the classical view of explicit coordination, our research with decision teams shows that teams with high task coordination make heavy use of a central file sharing facility to routinely exchange documents that contain information on simulation inputs and presentations for their board of directors. They also rely on face-to-face and electronic communications for less routine aspects of the task, like discussion of work assignments and key decisions to be made. Interestingly, the same explicit coordination factors were used by teams with high strategy coordination, except for the file sharing system, probably because it plays no role in helping manage the dependencies involved in integrating functional strategies, which are largely non-routine in nature, and are thus managed via communication. Shared mental models of the task and shared mental models of the team were concurrently correlated with both task and strategy coordination, but when the proper regression models were formulated, the effect of shared mental model of the team disappeared. This suggests that once the use of all explicit coordination mechanisms is accounted for, only the shared mental model of the task has an effect on coordination in this domain (i.e. decision task, co-located and asynchronous), thus highlighting the importance to consider the effects of all coordination mechanisms used. In contrast, both shared mental models had significant and positive effects in another study with flight simulation teams (Mathieu et al. 2000), suggesting that managing both, task and member dependencies are important for coordination in such domains (i.e., co-located, synchronous).

Similarly, our experiences from our research with software teams suggest that technical personnel (e.g., software developers, testing engineers, etc.) are more concerned with technical coordination, because they want to ensure that the different parts of software they are developing integrate well. So they rely heavily on software tools (e.g., CASE tools, change management system, etc.) and formal documents (e.g., application program interfaces, specifications, etc.), and communicate heavily when they encounter unusual situations or errors that need to be repaired. In addition, technical coordination problems tend to appear when members of the team don't have organized shared knowledge (i.e., shared mental models) about the products being developed and the functions being implemented. In contrast, technical and project managers are more concerned with temporal and software process coordination, because they want to ensure that things are done in a timely manner, per project schedules, and that things are carried out as specified by the formal software process established. These managers rely more on planning, project management and software process documents. They also have more frequent meetings because the nature of the dependencies they have to manage is mostly non-routine. Also, process coordination problems tend to appear when managers don't have organized shared knowledge about the established software process, which leads to confusion, duplication of work, and priority conflicts, among other things.

Effect of Input Variables on Use of Coordination Mechanisms: Nature of Dependencies

As discussed in the prior section, task, team and context variables influence the types of dependencies a team needs to manage and the mix of coordination mechanisms employed for this purpose. When conducting research on team cognition and coordination, it is important to understand how task, team and context variables affect the nature of dependencies present in a task for a number of reasons. First, some findings may be limited within the particular boundaries defined by the type of task, team and/or context of the used in the study because the nature of dependencies will be different outside of those boundaries. Second, different types of coordination mechanisms may be more effective at managing different types of dependencies. For example, a team experienced with the joint task and with a long history of working together will be more likely to rely on team cognition (e.g., shared mental models, transactive memory) than a team without any experience with each other or the task, whose members can only rely on task organization mechanisms (e.g., manuals, specifications) and communication. Finally, some task, team and context variables cannot be changed easily (e.g., type of task, organizational culture), while some others may be more easily changed through interventions (e.g., training, new member recruitment, technology, geographic location). Consequently, it may be easier to solve some coordination problems by changing the nature of dependencies via task, team and context interventions, than by forcing teams to use different coordination mechanisms.

In this subsection we discuss how task, team and four context variables can affect the nature of dependencies present. Two of the context variables we discuss, technology and organization, are important input variables used in team research because they have substantial effects on how teams interact. The other two context variables, synchronicity and dispersion, are essential in the study of teams mediated by information technologies because time separation (i.e., real time vs. different time) or distance separation (co-located vs. dispersed) have a substantial effect on how teams organize their work and interact (Bullen & Bennett 1993).

(a) The task. The nature of the task really matters when studying team interaction and performance (McGrath 1991) and it will affect which types of dependencies are present. For example, a team assembling gadgets in a mechanical task will face different dependencies than a medical team from different areas of expertise trying to diagnose a patient. For example, research on problem solving teams has shown that when team members believe that a demonstrable solution exists (i.e., an intellective task) it changes how information is sampled from team members, compared to when they believe that a demonstrable solution does not exist (i.e., a judgment task) (Stasser & Stewart 1992).

(b) The team. Many team variables can affect the nature of dependencies present. For example, how long or how often has the team worked or trained together in the past, the team composition (e.g., homogeneity, expertise, etc.), member acquisition practices (e.g., labor markets, socialization, etc.), and team size will affect team cognition development (Cannon-Bowers et al. 1993, Klimoski et al. 1994, Levesque et al. 2001, Rentsch & Klimoski 2001). Teams that have worked together for a long time may have well-developed team cognition mechanisms and work practices that help them minimize dependencies. Team variables can, not only affect team cognition development, but also which explicit mechanisms they employ. For example, the number and complexity of dependencies in software development increases exponentially as the size of the team increases and it makes it difficult to coordinate and communicate (Brooks 1995). So, larger teams are more likely to rely on task organization mechanisms (e.g., software tools, manuals, specifications, schedules, etc.), while smaller teams are more likely to rely on team communication (VanDeVen et al. 1976). Other studies with software teams have also found that project size and task uncertainty make coordination more difficult (Kraut et al. 1995), but that teams that employ formal routines reduce some of the resulting negative effects (Sproull et al. 1991).

(c) Context: Technology. Communications and other technologies can affect how teams interact (McGrath et al. 1994). For example, a seminal study of relationships between technologists and radiologists at two hospitals found that new technologies changed the nature of individual roles, which in turn had a substantial effect on how technologists and radiologists interacted (Barley 1990). Other research studies have also suggested that information technologies affect dependencies, information flow, and workflow among collaborators (Grinter et al. 1999, Sproull et al. 1991). For example, smaller software projects don't always utilize configuration management systems, which create member dependencies when different developers need to work on the same part of the software code. In contrast, large-scale software projects use such systems, which enable multiple developers to work on the same part of the software code simultaneously without having to interact with each other (i.e., the system handles this for them), which reduces member dependencies.

(d) ***Context: Organization.*** Organizational factors (e.g., culture, structure, standard procedures) also affect how teams interact. In fact, some research studies suggest that organizations are social systems that affect (and are affected by) how technologies are used (DeSanctis & Poole 1994, Orlikowski 1992, 1996), all of which can affect the types of dependencies present in a task.

(e) ***Context: Synchronicity and Geographic Dispersion.*** Teams can operate synchronously (i.e., same time or real time) or asynchronously (i.e., different time) and can be either co-located (i.e., same place) or geographically dispersed (i.e., different place), thus creating four possible modes of team interaction that need to be considered when researching teams (Bullen et al. 1993). Consequently, synchronicity and geographic dispersion are two task context factors that need to be considered when studying team cognition because they can generate different work arrangements with different resulting sets and types of dependencies. Asynchronous and geographically dispersed teams have fewer opportunities to interact, communicate less spontaneously, and use less rich media (e.g., electronic mail, telephone). Consequently, time and distance separation will not only make it more difficult for team members to develop and use shared cognition to coordinate, but it will also affect the mix of explicit and implicit mechanisms that they will employ to coordinate. Most of the team cognition research literature has focused on same-time/same-place contexts (e.g. lab studies with flight simulators or assembly kits), partly because of the importance of shared cognition to real-time, fast-paced team tasks like formation flights, medical emergency surgery, and sports competitions. Research in this context has produced very useful theoretical foundations, and its related empirical research is beginning to show evidence of the positive effects of team cognition (Liang et al. 1995, Mathieu et al. 2000, Stout et al. 1999). However, we need to be careful before generalizing these findings to different time and different place contexts. Fortunately, as we discussed earlier, new empirical research studies are beginning to emerge in different time-same place contexts (i.e., asynchronous, co-located), which are beginning to show evidence of the positive effects of team cognition on performance (Faraj et al. 2000, Lewis 2000), and helping identify antecedents of team cognition (Levesque et al. 2001, Rentsch et al. 2001).

On the other hand, there is almost no empirical research in different place contexts (i.e., geographically dispersed). Our research studies try to fill this gap by focusing specifically on different place and/or different time contexts. For example, a lab study with co-located dyadic teams engaged in a flight simulation task found that shared mental models of the task and team have a positive effect on team process, which in turn had an effect on team performance (Mathieu et al. 2000). In contrast, our study with decision teams found a positive effect of the shared mental model of the task, but found no effect of the shared mental model of the team. Our explanation for this is that the shared mental model of the team is perhaps important in high-paced, real-time tasks like a flight simulation because this shared model helps team members explain and anticipate each other's actions, which helps manage member dependencies, which are key to performance in these contexts. Conversely, management decision teams are asynchronous, so having shared knowledge of the joint task is perhaps more important for managing the respective task dependencies than having a well-developed shared mental model of the team. Similarly, our research with software teams (i.e., asynchronous, dispersed) also found that dependencies and coordination needs are very different for co-located than for dispersed teams, but that having organized shared knowledge about key concepts, processes and products (i.e., shared mental models), and knowing who knows what in the team (i.e., transactive memory) helps teams coordinate (Espinosa et al. 2000) thus reducing some of the coordination problems associated with geographic distance.

As discussed in the prior section, research in team cognition can benefit by taking into account the task context, particularly synchronicity and geographic dispersion. Research findings are likely to differ depending on time and distance separation because the nature of the dependencies present will change. In fact, interventions that change the synchronicity or dispersion of a task can have substantial effects on coordination. For example, two separate studies in the same organization where we conducted our study with software teams found that large-scale software projects have more complex dependencies and coordination challenges when development is done across sites than when it is done at a single site (Herbsleb et al. 2001, Herbsleb et al. 2000). But a related study also found that large-scale software development is generally distributed across sites in one of four ways (to purposely reduce dependencies across sites): (a) develop all related software functions (e.g., call handling functions, billing functions) in a single site; (b) develop software for similar parts of the software product (e.g., base station controllers) in a single site; (c) keep similar software development phases (e.g., testing) in a single site; (d) develop a core product in a single site and customize it for specific client needs at a site near the client (Herbsleb & Grinter 1999).

Discussion

Our arguments and examples in this chapter bring attention to the fact that it is important to consider explicit coordination mechanisms and task dependencies when we study team cognition. Results from empirical studies in team cognition will be more useful if they limit the scope and validity of the study to particular contexts involving the use of a given set of explicit coordination mechanisms. On the other hand, if the use of explicit mechanisms vary across teams, it is important to account for these mechanisms, thus the importance of using an integrated research framework that incorporates both, implicit and explicit coordination mechanisms. It also important to understand how task, team and context variables affect work arrangements and task dependencies (i.e., open the task's black box) when we study team coordination. Empirical findings from research studies about the effect of a particular team cognition construct on team coordination and performance will be more useful for research and practice if the specifics of the task dependencies are articulated and considered, than empirical findings about more generalized effects of these constructs. Also, while coordination may not be a sufficient condition for effective team performance, it is certainly necessary for tasks with substantial dependencies. At the same time, teams that are highly coordinated in certain aspects of the task may not necessarily exhibit high performance levels unless other antecedents of performance are also present.

There is an increased interest these days in the team cognition research literature on how different types of shared cognition influence team coordination and performance (Cannon-Bowers et al. 2001, Cannon-Bowers et al. 1993, Klimoski et al. 1994). Many team cognition constructs have been described and studied in this literature (Mohammed & Dumville 2001) and, while these constructs are conceptually distinct, they share some similarities in that most of them involve some form of compatible, complementary or similar knowledge teammates have (Cannon-Bowers et al. 2001), which helps them develop accurate explanations and expectations about task states and member actions, thus helping the team manage task dependencies and coordinate implicitly. These concepts and theories are very intuitive and useful in helping us explain team coordination and performance, but as acknowledged in this literature, most of this work is conceptual and theoretical. Through our research studies, we have identified a few specific areas and issues that need further development to better understand team cognition:

1. There is very little agreement or consistency in the literature about how to measure shared cognition (Cooke et al. 2000, Lewis 2000, Mohammed et al. 2001). Further research and more consistency of methods are needed in this area to be able to compare results across studies.
2. There is very little empirical work supporting these theories and concepts. More empirical work is necessary in this area (Mathieu et al. 2000, Mohammed et al. 2001). Part of the reason for this has to do with the difficulty of measuring shared cognition.
3. There is not much discussion in the literature about how the effects of team cognition may vary depending on the type of task involved. However, the nature of the task really makes a difference when it comes to understanding team performance (McGrath 1991). Furthermore whether teams work synchronously or asynchronously, and whether team members are co-located or geographically distributed also makes a difference (Bullen et al. 1993).
4. How team cognition affects coordination and performance will depend on the extent to which teams use certain explicit coordination mechanisms like collaboration tools, electronic communication, shared databases, and division of labor. Further research is needed in which the effects of explicit and implicit coordination mechanisms are jointly explored.

5. Complex tasks have multiple dependencies. The management of the specific dependencies involved in a task will require different types of coordination. Further research is needed in which such dependencies are investigated in order to better understand: (a) the different effects that specific explicit and implicit coordination mechanisms have on the effective management of specific task dependencies; and (b) how different types of coordination affect different aspects of team performance.

The first point about measures has been partly addressed in other studies (Cooke et al. 2000, Espinosa 2001, Espinosa & Carley 2001a). In this chapter we have addressed the remaining four issues by focusing on two overarching themes of concern for team cognition research. The first one is the need for a research framework that incorporates explicit mechanisms (from the classical organizational research literature) and implicit coordination mechanisms (from more recent research in team cognition). The second overarching theme has to do with the need to open the task's black box to better understand the nature of the multiple dependencies and coordination types involved in a task (i.e., one size does not fit all). Complex tasks often have many types of dependencies, some more critical than others and some coordination mechanisms may be more effective than others for particular types of dependencies. Conversely, one particular mechanism (e.g. shared mental model of the task) may not be equally effective in managing different dependencies. Consequently, an effective strategy for coordination success (i.e., high state of coordination) involves finding a mix of coordination mechanisms well suited for the task. Furthermore, this mix may need to change as the task progresses over time (e.g., communication may be more effective at the beginning of the task while division of labor or shared mental models may be more effective later on). This mix will also need to change with time and distance separation (i.e., co-located vs. distributed).

References

Barley, S. (1990). "The Alignment of Technology and Structure Through Roles and Networks". *Administrative Science Quarterly*, 35(March): 61-103.

Brooks, F. (1995). *The Mythical Man-Month: Essays on Software Engineering* (Anniversary ed.): A. Wesley.

Bullen, C. & Bennett, J. (1993). "Groupware in Practice: An Interpretation of Work Experiences". In Baecker, R. (Ed.), *Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration*. CA: Morgan Kaufman Publishers.

Cannon-Bowers, J. E., Salas, E., & Converse, S. (1993). "Shared Mental Models in Expert Team Decision-Making". In Castellan, J. (Ed.), *Individual and Group Decision-Making: Current Issues*: 221: LEA Publishers.

Cannon-Bowers, J. E. & Salas, E. (2001). "Reflections on Shared Cognition". *Journal of Organizational Behavior*, 22(March): 195-202.

Cooke, N. J., Salas, E., Cannon-Bowers, J. A., & Stout, R. J. (2000). "Measuring Team Knowledge". *Human Factors*, 42(1): 151-173.

Crowston, K. & Kammerer, E. E. (1998). "Coordination and Collective Mind in Software Requirements Development". *IBM Systems Journal*, 37(2): 227-245.

Curtis, B., Krasner, H., & Iscoe, N. (1988). "A Field Study of the Software Design Process for Large Systems". *Communications of the ACM*, 31(11): 1268-1286.

DeSanctis, G. & Poole, M. S. (1994). "Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory". *Organization Science*, 5(2): 121-145.

Endsley, M. (1995). "Toward a Theory of Situation Awareness in Dynamic Systems". *Human Factors*, 37(1).

Espinosa, J. A., Herbsleb, J. D., Mockus, A., Kraut, R. E., & Slaughter, S. A. (2000). "Coordination in Large-Scale, Distributed Software Development: A Field Study". *working paper*.

Espinosa, J. A. (2001). "Shared Mental Models: Accuracy and Visual Representation". Paper presented at the Americas Conference in Information Systems, Boston MA.

Espinosa, J. A. & Carley, K. (2001a). "Measuring Team Mental Models". Paper presented at the Academy of Management Conference, Washington D.C.

Espinosa, J. A., Carley, K., Kraut, R. E., Lerch, F. J., & Fussell, S. (2001b). "The Effect of Shared Mental Models and Knowledge Distribution on Asynchronous Team Coordination and Performance: Empirical Evidence from Management Teams". *working paper*.

Faraj, S. & Sproull, L. (2000). "Coordinating Expertise in Software Development Teams". *Management Science*, 46(12): 1554-1568.

Grinter, R. E., Herbsleb, J. D., & Perry, D. E. (1999). "The Geography of Coordination: Dealing with Distance in R&D Work". Paper presented at the Group '99, Phoenix, Arizona.

Hackman, R. (1987). "The Design of Work Teams". In Lorsch, J. (Ed.), *Handbook of Organizational Behavior*: Prentice-Hall.

Helmreich, R. L. (1997). "Managing Human Error in Aviation", Scientific American, Vol. 277: 40.

Herbsleb, J., Mockus, A., Finholt, T., & Grinter, R. E. (2001). "An Empirical Study of Global Software Development: Distance and Speed". Paper presented at the 23rd. International Conference on Software Engineering (ICSE), Toronto, Canada.

Herbsleb, J. D. & Grinter, R. E. (1999). "Architectures, Coordination, and Distance: Conway's Law and Beyond". *IEEE Software*(September/October): 63-70.

Herbsleb, J. D., Mockus, A., Finholt, T., & Grinter, R. E. (2000). "Distance, Dependencies and Delay in a Global Collaboration". Paper presented at the CSCW 2000, Philadelphia, Pennsylvania.

Kiesler, S., Wholey, D., & Carley, K. (1994). "Coordination as Linkage: The Case of Software Development Teams". In (Ed.), D. H. (Ed.), *Organizational Linkages: Understanding the Productivity Paradox*: National Academy Press.

Klimoski, R. J. & Mohammed, S. (1994). "Team Mental Model: Construct or Methaphor". *Journal of Management*, 20.

Kraemer, K. & Pinsonneault, A. (1990). "Technology and Groups: Assessment of the Empirical Research". In Galegher, J. & Kraut, R. E. & Egido, C. (Eds.), *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*: LEA Publishers.

Kraiger, K. & Wenzel, L. (1997). "Conceptual Development and Empirical Evaluation of Measures of Shared Mental Models as Indicators of Team Effectiveness". In M. Brannick & E. Salas & Prince, C. (Eds.), *Team Performance Assessment and Measurement*: LEA Publishers.

Kraut, R. E. & Streeter, L. A. (1995). "Coordination in Software Development". *Communications of the ACM*, 38(3): 69-81.

Levesque, L. L., Wilson, J. M., & Wholey, D. R. (2001). "Cognitive Divergence and Shared Mental Models in Software Development Project Teams". *Journal of Organizational Behavior*, 22(March): 135-144.

Lewis, K. (2000). "Transactive Memory and Performance of Management Consulting Teams: Examining Construct and Predictive Validity of a New Scale". Paper presented at the Academy of Management Conference, Toronto.

Liang, D., Moreland, R., & Argote, L. (1995). "Group Versus Individual Training and Group Performance: The Mediating Role of Transactive Memory". *Personality and Social Psychology Bulletin*, 21.

Malone, T. & Crowston, K. (1990). "What is Coordination Theory and How Can it Help Design Cooperative Work Systems". Paper presented at the Computer Supported Collaborative Work.

Malone, T. & Crowston, K. (1994). "The Interdisciplinary Study of Coordination". *ACM Computing Surveys*(March).

March, J. & Simon, H. (1958). *Organizations*: John Wiley and Sons.

Mathieu, J., Goodwin, G. F., Heffner, T. S., Salas, E., & Cannon-Bowers, J. A. (2000). "The Influence of Shared Mental Models on Team Process and Performance". *Journal of Applied Psychology*, 85(2): 273-283.

McGrath, J. (1991). "Time, Interaction and Performance (TIP)". *Small Group Research*, 22.

McGrath, J. & Hollingshead, A. (1994). *Groups Interacting With Technology*: Sage Publications.

Mohammed, S. & Dumville, B. C. (2001). "Team Mental Models in a Team Knowledge Framework: Expanding Theory and Measurement Across Disciplinary Boundaries". *Journal of Organizational Behavior*, 22(March): 89-106.

Orlikowski, W. (1992). "Learning from Notes: Organizational Issues in Groupware Implementation". Paper presented at the CSCW.

Orlikowski, W. (1996). "Improvising Organizational Transformation Over Time: A Situated Change Perspective". *Information Systems Research*(March).

Rentsch, J. R. & Klimoski, R. J. (2001). "Why do Great Minds Think Alike?: Antecedents of Team Member Schema Agreement". *Journal of Organizational Behavior*, 22(March): 107-120.

Rouse, W. B. & Morris, N. M. (1986). "On Looking Into the Black Box: Prospects and Limits in the Search for Mental Models". *Psychological Bulletin*, 100: 349-363.

Sproull, L. & Kiesler, S. (1991). *Connections: New Ways of Working in the Networked Organization*: MIT Press.

Stasser, G. & Stewart, D. (1992). "Discovery of Hidden Profiles by Decision-Making Groups: Solving a Problem Versus Making a Judgement". *Journal of Personality and Social Psychology*, 63(3): 426-434.

Stout, J. A., Cannon-Bowers, J., & Salas, E. (1999). "Planning, Shared Mental Models, and Coordinated Performance: An Empirical Link is Established". *Human Factors*, 41: 61-71.

Thompson, J. (1967). *Organizations in Action*: McGraw-Hill.

VanDeVen, A. H., Delbecq, L. A., & Koenig, R. J. (1976). "Determinants of Coordination Modes Within Organizations". *American Sociological Review*, 41(April): 322-338.

Wegner, D. (1986). "Transactive Memory: A Contemporary Analysis of the Group Mind". In B. Mullen & Goethals, G. (Eds.), *Theories of Group Behavior*: Springer-Verlag.

Weick, K. (1990). "The Vulnerable System: An Analysis of the Tenerife Air Disaster". *Journal of Management*, 16(3): 571-593.

Weick, K. (1993). "The Collapse of Sensemaking in Organizations: The Mann Gulch Disaster". *Administrative Science Quarterly*, 38(4): 628-652.

Weick, K. & Roberts, K. (1993). "Collective Mind in Organizations: Heedful Interrelating on Flight Decks". *Administrative Science Quarterly*, 38(3): 357-381.

Wellens, R. (1993). "Group Situation Awareness and Distributed Decision-Making: From Military to Civilian Applications". In Castellan, J. (Ed.), *Individual and Group Decision-Making: Current Issues*: 267: LEA Publishers.

Wholey, D., Kiesler, S., & Carley, K. (1996). "Learning Teamwork: Emergence of Organization in Novice Software Development Teams". *Working paper, Carnegie Mellon University*.

Wittenbaum, G. M. & Stasser, G. (1996). "Management of Information in Small Groups". In Nye, J. L. & Brower, A. M. (Eds.), *What's Social about Social Cognition?*: p.3: Sage Publications.

Appendix A

Study with Decision Teams

The data for this research comes from Carnegie Mellon's Management Game, a graduate-level management course (1998 session) in which student teams run simulated companies and compete with each other for approximately fourteen weeks. Game teams range in size from 4 to 6 members, but 75% or more of them have 5 members. Each team (i.e., firm) reports to a board of directors composed mostly of business professionals from the local business community. Team members assume management roles (i.e. president, v.p. marketing, v.p. finance, etc.). Firms compete against each other by formulating strategies based on multidisciplinary decisions involving production, distribution, finance, marketing and strategy. There is also a simulated stock market in which company shares are traded among students. The Game environment represents competitive business environments in which management teams make routine decisions and handle ad-hoc crises, which are introduced by instructors to keep teams under pressure and in constant need to share information and make decisions. Game students compete with each other for their grades, which are largely based (70%) on firm financial performance and board evaluations. Students meet in the classroom only once per week, mostly to exchange information with course instructors, but they do most of their work outside of the classroom. Teams make marketing, production and finance decisions, which are entered into a simulation that models firm competition and yields financial performance results for all firms. Each team meets with its board of directors three times during the course to report on current performance and obtain advice and consent on actions and strategies. These meetings are long and involved, often lasting several hours. Data is systematically collected for research purposes during the course. The data used for this study consists primarily of: (1) Three voluntary student surveys, conducted at the beginning, middle and end of the course. Approximately 70% of the students completed the surveys. We only used data for teams with 3 or more responses, representing approximately 74% of the teams; (2) Firm financial performance data (i.e., profits, return on investment, and stock price) recorded for each of the 10 simulated quarters in the course; and (3) Three team evaluations by board members completed immediately after each board meeting, roughly coinciding in time with student surveys.

Appendix B

Study with Software Teams

This study investigated the effect of team cognition on coordination in large-scale, geographically distributed software development at a large telecommunications company, with a specific focus on shared mental models. The study was conducted using a multi-method approach, which included three separate studies: a qualitative study, a survey study, and a quantitative study. The qualitative study is based on face-to-face, semi-structured interviews of 36 software developers, technical managers and project managers; observations of 9 cross-site coordination meetings (using either voice conferencing or video conferencing), and analysis of approximately 300 messages of group electronic mail correspondence by coordination groups collected over a period of one year. The interviews were conducted at two European sites that produce software for the wireless telecommunications industry. The survey and the quantitative studies were based on software development projects for telephony switching equipment. The survey study was conducted at two sites—England and the U.S.—who develop over 90% of the software code for one of the main sub-systems of these switches. The web survey instrument was dynamically generated for each respondent with questions about specific projects in which they worked, and about each team member that collaborated with them on these projects. The survey was designed to measure process variables, particularly those related to team communication, task programming and shared mental models, and to obtain measures of technical, temporal and software process coordination outcomes. The quantitative study was conducted with data from the software configuration management system for the same switch sub-system, which tracks production data from every modification made to the software. This data was used to evaluate the effect of prior work similarity (i.e., how many projects, modules and files each pair of developers had in common) on project implementation time interval.