

A Collaborative Project Management Architecture

Fang Chen University of Arizona FChen@CMI.Arizona.EDU	Nicholas C. Romano, Jr. Oklahoma State University Nicholas-Romano@MSTM.OKState.EDU
Jay F. Nunamaker, Jr University of Arizona Jnunamaker@cmi.arizona.edu	Robert O. Briggs University of Arizona RBriggs@cmi.arizona.edu

Abstract

The Project Management (PM) paradigm is rapidly shifting due to business globalization and information technology (IT) advances that support distributed and virtual project teams. Traditional PM focuses on a single project at a single location [16] and is more concerned with project inputs and outputs than with project process [51]. Management in the past implied projects were conducted with a top down view [13]. The PM paradigm has begun to change due to the increasing number of distributed projects involving project collaborators from different locations, organizations, and cultures [27]. Current and future PM will be more concerned with tracking project work processes and efficient and effective sharing of information and knowledge, among project contributors. High-levels of collaboration will become essential for distributed project success. Task interdependence and member distribution across time, space, and technology will make high degrees of collaboration necessary to accomplish project work. Adequate and timely sharing of information, and knowledge in all directions, proactive change management, and process monitoring are some of the important factors required for successful project collaboration [36]. In this article, we review problems associated with traditional PM scenarios, explain how collaborative PM can provide solutions, present a comparison of current commercial collaborative PM tools, and propose a collaborative PM architecture to address the challenges facing distributed projects teams.

Keywords:

Collaborative Project Management Architecture, Explicit Communication, Explicit Project Knowledge, Tacit Knowledge, Collaborative Middleware, Collaborative Presence, Process Management.

1. Introduction

The project management (PM) paradigm has been shifting in recent years toward a more collaborative model [13, 16, 27, 36, 44]. In the past PM focused on ‘*management*’, which implied a top down view of how projects are conducted [13]. A few individuals high in the organizational hierarchy had the total picture of a project, planned the project and assigned tasks to others for completion. Individuals were not supposed to, nor allowed to make decisions, and might not be clear as to what possible effects their individual work might have on the project as a whole. In this situation, the only information an individual needs to know is that related to their individual tasks. However, this style of PM only works in repeat product and process environments [19]. The assumption that projects are conducted

within repeat product and process environments is no longer valid for most projects today, due to rapid technology advancement, business globalization, high personnel turn over, and distributed team membership [16, 19, 27, 44].

Over the past decade, the project landscape has undergone a major change due to international mergers, shortened time to market, and changing labor costs, and increasing involvement of professionals distributed in geographical locations [16].

Distributed projects are also called “*virtual*” projects. Distributed (virtual) projects are different from local projects in several aspects: “*contributors lack face-to-face interactions and have different cultural backgrounds, advanced information technology and infrastructures mediate remote cooperation, and time zone differences reduce real-time communications to only a few hours per day*” [18]. For traditional PM the key issue is scheduling [16], for multiple traditional projects, in addition to scheduling, there is a need to share resources to achieve global optimization. “*A critical difference between distributed projects and the prior programs or traditional projects of various types is related to the focus on the coordination mechanisms*” [16]. Coordination and collaboration are important components for local and virtual projects: coordination is within location for traditional projects, and across locations for distributed projects [16].

One challenge for distributed projects is inter-member communication. A high degree of informal and “*ad hoc*” communication is important for distributed project success [35]. However, distance affects the frequency of communication, especially the informal communication and tacit knowledge sharing among project members [5]. In addition to communication, distributed projects impose greater challenge for project coordination due to the interdependence among project tasks

Traditional PM focuses on management [13], scheduling and project outcomes [51]. Distributed PM emphasizes collaboration [24] and project process [36]. Distributed projects thus impose at least three major challenges for PM: collaboration (supporting high levels of interaction, communication, and coordination); knowledge management (capturing tacit knowledge, turning tacit knowledge into explicit knowledge, and sharing explicit knowledge); and work process (analyzing task interdependence, documenting decision rationale and the work process itself). One type of software or another has addressed each of these

challenges separately. For example, Group Support Systems (GSS) was designed to support group meetings and communication, however, GSS does not provide systematic ways of managing knowledge and work process. As an alternative, workflow management systems can be used to support and manage work processes. However, workflow management systems can only support repetitive and predefined work processes [36]. Maurer concluded that GSS does not provide enough structure and workflow techniques are not flexible to support virtual projects [36]. This leads us to believe that the areas supported by each must be included in the architecture and the software to support collaborative distributed projects.

Telephone calls, and email exchanges can provide flexible communication for project contributors; however, the content of this ad hoc communication has rarely been tracked for later use [54]. Knowledge management (KM) may be conducted via paper and electronic formats, but the storage structure is not well defined and locating a specific document or text fragment is difficult at best. Moreover, documents are usually store in a disjointed fashion and are more often than not only in a paper format [54]. Such paper-based storage makes coordination a challenge that may fail under time constraints [7, 49].

We assert that there is a need for Collaborative Project Management Architectures (CPMAs) within which to build systems that address the aforementioned challenges. Support for our assertion is found in the rapidly increasing collaborative PM software market. A report published in May 2000 estimated that distributed PM software market revenues were expected to grow 36% over the next 12 months and that by 2003 revenues may increase from \$700M to nearly \$1.5B [1]. Clearly there is an expected need for CPMAs and CPM systems.

The remainder of this article is structured as follows: section 2 reviews problems associated with traditional PM scenarios. Section 3 explains how collaborative project management can provide solutions. Section 4 presents our collaborative PM architecture and presents a comparison of current commercially available CPM software. Section 5 presents a summary discussion and future research directions. The contribution of the paper is the outline of a CPMA within which a CPM system can be built that will provide the levels of collaborative and process support required for distributed project success.

2. Traditional Project Management Scenarios

When individuals or organizations carry out PM there are many potential mistakes or pitfalls to which they may fall prey. Instead of listing them all, we focus on several common overarching themes identified in the literature as follows: Overemphasizing the project reporting aspect of PM [12], ineffective and inefficient communication [12, 24, 47], managing project inputs and outputs but not process [51], reactive PM [17, 37, 47], and the lack of a project repository [54]. Together

all of these themes account for the reason why many distributed project either fail or are significantly less efficient and effective than they could be with increased collaborative and process support.

2.1. Overemphasis of PM as a Project Reporting Mechanism

Traditional PM is often employs a simple passive reporting mechanism instead of a dynamic teamwork coordinating effort [12, 51]. Clarke [12] illustrated this point: *“in many organizations, the project management methodology is regarded as a corporate reporting tool rather than a useful system that the various parts of the organization can use to help themselves.”* In this situation, information flow is minimal and inadequate among project contributors. Low information sharing may result in ineffective communication [12, 24, 47]. This is a common problem in many face-to-face projects that is exacerbated in distributed environments.

2.2 Ineffective and Inefficient Communication

In traditional PM communication may be ineffective in many senses: misunderstandings due to implicit communication; members having a poor grasp of the problem; failure to attain a shared vision; hidden agendas; and different interpretations by different people. Communication may be inefficient in a number of way as well: untimely communication, failure to notify everyone that needs to know, top down communication only with no concomitant bottom up communication, and failure to store information for future use in formats that are retrievable and searchable. Poor communication skills and capabilities are often cited as the primary reason for project failure [12, 24, 47].

2.3. Managing Project Inputs and Outputs but not Process

Another serious problem facing PM is that people manage deliverables and resources, but not project work nor project process [51]. Project managers create PERT and Gantt charts to plan the project timeline, they manage time, money, equipment, human resources and the product; however, they don't often manage work process [17, 51]. As a result, there is little written about how to manage the work of a project [51]. One reason software projects fail is the lack of a real-time progress measurement systems to identify potential risks early, before they become serious threats to success [17]. If people only manage project inputs and outputs, the process remains a black box and project members don't know something has gone wrong until it may be too late to correct the problem without causing large amounts of rework. This tends to make PM a reactive process, rather than a proactive one [37, 47].

2.4. Reactive Management

“Reactive management” refers to a passive PM strategy in which project managers conduct inadequate planning hoping that everything will turn out all right in the end. Common practices of reactive management include: failure to adequately plan out the entire project and generate sufficient alternatives at the beginning of the

project life cycle [47], insufficient risk analysis and management, abandonment of planning under pressure, inadequate analysis and design, and procrastination of planning tasks in the hopes of catching up later in the life cycle [37]. Reactive project managers respond to what has happened and seldom plan for the future, and they may not review their own or others' past experiences to gain insight from lessons learned over time. In reactive management, people spend a significant amount of project time on reworking deliverables and correcting errors [47]. Another common problem in reactive situations is that much of the rework must be done manually, including searching for work that is affected by changes in other parts of the project. Reactive PM is often accompanied by lack of systematic method for storing project information, thus compounding the problems of poor planning and the need for rework.

2.5. Lack of an Electronic Project Repository

Lack of an electronic repository is an organization-wide issue as well as a project-specific problem. Historically, companies developed paper-based, function-centered Management Information Systems (MIS) [7]. One study reported that 90% of North American data is only available in hard-copy form [7, 48]. A separate study indicated that only about 5% of a firm's documents are in digital format and able to be electronically processed [7, 29]. A paper-based repository has several drawbacks including: retrieval delays, lost documents, and storage problems. Additionally paper storage is error-prone due to data extraction, interpretation, and repackaging [7, 30, 33, 49]. Paper-based systems are hard to coordinate and may fail under time constraints [7, 49].

When organizations begin to utilize IT to manage information, they may end up creating information islands [7, 20]. Information management may be automated in one functional area but not in other areas, or all areas may be automated separately such that there is no effective connection between the information among the different areas. A study indicated that about 70% of all business data manually entered into one computer are eventually manually entered into another computer and that 25% of the total cost of business transactions is due to data entry and reentry [7, 43], again illustrating the creation of islands of information that is not shared for the good of a project team or the organization as a whole [7, 20].

Lack of an electronic project repository also results in insufficient project documentation. Project members are usually more concerned with completing current project tasks than with capturing and archiving information that may be useful at a later time [54]. Much project related information is not captured at all such as project processes, contexts, rationales, or artifacts. Even if they are captured, they may not be stored, organized and indexed in a way that enables project members to easily access, search, and retrieve the information [54]. *“Formal project documentation such as project reports, meeting minutes, and*

correspondence usually exists as disjointed documents stored in file folders. Informal items like electronic mail messages and personal notes are often not retained at all. Project contexts, underlying idea progressions, and rationales behind key decisions are effectively lost as an organizational resource when project team members leave for new assignments and human memories fade” [54]. Lack of a project repository may lead to rework because project members may not be aware that others have already complete the same or a very similar task. It may also result in losing the opportunity to reuse some project artifacts and processes in the future. The long-term effect of lack of a project repository is a loss of organization memory and learning [25, 54].

3. Collaborative Project Management as a Solution

We assert that each of the problems discussed in section 2 can be addressed by using collaborative PM tools and processes. A collaborative PM tool focuses on explicit representation of project information and timely sharing of the information. The overarching goal is to get the right information to the right people at the right time. We discuss how a collaborative PM environment can overcome the limitations that plague traditional PM.

3.1 Emphasizing PM as A Project Analysis Mechanism

When people treat PM as a project reporting tool, they emphasize the outputs of the PM rather than the analysis process which produces those outputs. For example, people may make a PERT chart or Gantt chart to schedule the project. If people have gathered together and made significant effort to make a reasonable chart, they may have gone through multiple decision making loops to produce the chart: such as breaking down the project into manageable tasks, estimating processing time for each task, organizing task order, identifying task interdependencies, estimating possible risks related to each task, and selecting alternatives to mitigate the risks. When people emphasize PM as a project reporting tool, the product of the above process is a PERT chart or Gantt chart. A great deal of important additional project-related information is usually not formally captured, and will effectively be lost when memory fades.

On the other hand, when people treat PM as a project analysis tool rather than merely a reporting tool, the product will be the chart, task information, decision rationale, and other related artifacts. A collaborative PM tool should facilitate members in conducting group processes such as: generating ideas, organizing ideas, and selecting alternatives. If results are stored in a permanent repository they can be used for future project analysis. Later on when changes are made in the schedule, people can retrieve related information to estimate the possible effects or consequences the change may cause, notify those whose work will be affected, and mitigate negative change consequences.

Collaborative PM tools can provide for high levels of information flow between project team members and thus result in more efficient and effective communication. The PM process can benefit from process gains such as more information, synergy, more complete alternative analysis, and idea triggering, that have been well documented in the GSS literature [40.] Effective communication and timely information sharing are essential for successful PM [11, 26, 28, 31, 50, 53.]

3.2 Effective and Efficient Communication

Explicit representation of project information is essential to effective and efficient communication, especially in distributed situations. *“A tool for project coordination support has to be based on explicit representations (or ontologies) of the processes, products, resources, organizational structures, interactions, and negotiation & co-operation strategies.... Explicit representations allow the system to have a ‘kind of understanding’ about what is going on in the project”* [36.] Documenting project information at a great level of detail will enable team members to gain a comprehensive project understanding. For example, co-located project members may work independently on their own tasks, write the report formally, and report the process and rationale to the group orally. Distributed project members have to document many more details including process (e.g. the steps of conducting tasks, the tools used) the rationale (e.g. the reasons of choosing a certain tool or methodology), and context (e.g. task participants, the assignment date, the due date, the actual finished date)

Effective communication also implies clear specification and unanimous agreement of important project information such as key concepts, ideas, project process, and member responsibilities. All these have to be documented and saved for project members’ reference. To provide this fine level of detail, a collaborative PM tool must not only support but sometimes even force members to document the context and process information of a task. In addition to support for explicit representation of project information, a collaborative PM tool needs to support automatic notification of task status changes, and allow members to discuss and comment on one another’s work. Explicit representation is not the same as effective communication, however, it is an important first step toward effective communication.

Such explicit documentation during the course of the process also leads to more efficient communication as the information is systematically stored in a manner that facilitates easy and fast retrieval. The structured nature of such detailed information can also speed up the process of searching for specific information or trends and patterns across time, tasks, and even multiple projects.

3.3 Managing Project Process as well as Inputs and Outputs

It is important to manage the process as well as the inputs and outputs. Managing the project process is the most dynamic part of PM, and is also an area in which little research has been done [51]. One way to think about the process is through a project lifecycle. The project lifecycle can be summarized into four major steps: understanding the project (problem definition); planning the project; executing, tracking, and controlling the project; and closing the project [32]. When people manage inputs and outputs, but not the process, they overemphasize steps 1, 2 and 4 at the expense of step 3. Process should be considered throughout the project lifecycle. Process tracking enables efficient and effective change management. Except for simple or recurring projects, there are usually some changes in project inputs, outputs, requirements, technology used, and availability of resources (e.g. time, money, and personnel). Without close monitoring of the process, the current project status may not be clear to the project managers, let alone the team members in the trenches carrying out the day to day operational tasks. When status is unclear, it will be difficult if not impossible to estimate risks caused by changes or to identify alternatives to mitigate those risks in an efficient and timely manner. Project processes are by their very nature dynamic and therefore may change significantly from the original project plans and expectations as the project progresses. Ongoing process will almost always lead to some changes in project inputs and outputs and these changes, in turn, will lead to further changes in the project process.

Insufficient project tracking may result in the process being viewed as a black box, which allows many potential problems to arise: such as injection of errors, off-track project efforts, and lack of control. On the other hand, sufficient project tracking provides visibility of the project process and therefore increases the probability of identifying project risks early on, learning successful behaviors, mitigating risks accordingly, and documenting lessons learned for future use [37]. A collaborative PM tool would allow project members to update, view one another’s work progress, collect project measures (e.g. resource spend on the task), and access the current work of others in a timely manner [42, 50, 53]. These project tracking mechanisms help project members to conduct process management in a more efficient and effective way. If a task is pending due, the task owner may get a warning message. If a task is available before the due date, members waiting for task completion can be notified and start the next task earlier than expected, thus injecting slack into the schedule, which may be needed later if unforeseen problems arise downstream. If a task is past due, those whose work will be affected can be notified. Project measures can help managers and members to assess the wellbeing of the process. If baselines are available for a specific project measure, project members can compare current performance against the baseline and diagnose the wellbeing of their process. The overall process management goal is to bring a project under control and on-target. Adequate

process management is essential for proactive management.

3.4 Proactive Project Management

Proactive project management means future-oriented planning, risk management, and change management [37]. Proactive management requires project team members to conduct clear and detailed planning at the beginning of the project cycle, identify potential risks, and make plans to mitigate those risks. People who conduct proactive management analyze task interdependencies carefully, monitor project process closely, make necessary changes accordingly, and limit the effect of negative changes as much as possible. They also make their decisions based on accurate “hard” data rather than wishful thinking. Such proactive management is likely to enable them to better predict what may happen and be prepared for it. They also reflect upon past experience to benefit from organizational memory, and apply successful lessons learned to the present project. Proactive management leads to learning. The components of a collaborative PM tool need to facilitate project analysis, effective communication, and process monitoring to enable effective proactive management. Proactive management of the PM process requires an organizational “*project memory*” from which members can learn during the present project and refer back to on future projects. One way to implement an effective organizational project memory is with an electronic project repository [47].

3.5 Employing an Electronic Project Repository

The paper-based repository should be replaced with an electronic project repository. With the advancement of information technology, documents in digital format are easier to store, access, retrieve, edit, and route. The goal of an electronic project repository is to efficiently and effectively manage and share project information. Effective information management can improve project performance by saving money, reducing data entry and reentry costs, eliminating duplication and information loss, reducing product development time, fostering improvement in process quality, standardizing work processes, improving management’s ability to efficiently retrieve accurate information, and increasing management control [7]. An electronic project repository can be connected via middleware with other information systems in the organization and provide a smooth information flow.

4. Collaborative Project Management Architecture

From the above discussion, we can see that there are many functions a collaborative PM tool has to support to provide value-added for the project team. We are not concerned how to implement each of these functions by using a specific technology; rather we think it is more valuable to propose a top-level collaborative PM architecture, within which CPM systems could be built and integrated with other organizational systems that will need to be employed during project execution. The architecture is technology independent and offers a comprehensive view of what a collaborative PM

environment might look like. Our architecture provides guidance for collaborative PM tool development by laying out the major components and a proposed method of integration.

We propose a top-level collaborative PM architecture based on the previous research. We have illustrated that collaborative PM environments can mitigate many of the problems associated with traditional PM and distributed projects; however there is currently no overarching CPM architecture to adequately support the distributed PM process. While some researchers have developed PM architectures, they do not offer the kind of collaborative support necessary for distributed projects to be successful. In this section we first discuss two previous architectures that influences our thinking, and then we present our CPMA. This architecture serves as an overview of collaborative PM: inputs and outputs of the system, factors that need to be considered by the system, services provided by the system, and how services coordinate and integrate with one another.

4.1. Two Previous Project Management Architectures

A number of PM architectures have been proposed. Figures 1 and 2 present two that influenced the development of our collaborative PM Architecture.

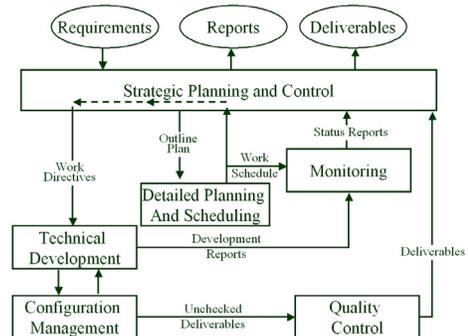


Figure 1. Dixon’s Integrated Model for PM [15]

Figure 1 is a software development PM tool architecture developed by Dixon [15]. The system supports three major management areas: PM, resource management, and cost management. PM involves planning, estimating, and scheduling the activities within the resource constraints to meet product performance criteria. Resource management involves resource identification and allocation. Cost management involves the analysis of information about planned and actual consumption of resources within the project and is concerned with project monitoring and control. The system inputs are the requirements. The Detailed Planning And Scheduling module performs both project and resource management. The Technical Development and Configuration Management modules perform PM functions. Quality Control and Monitoring modules provide monitoring and control services. System outputs include reports and deliverables. Dixon’s model does not include project repository, and

has no collaborative aspects. It seems that the management process is sequential in nature and the influence of one module on the next is one-way. In real PM situations, different management considerations influence one another in a parallel, cyclic manner, and there is seldom a sequential or one-way influence. This model may be applicable to well-defined and repeat environments; however, it may underestimate the complexity of distributed projects and the collaboration required to make them successful. Figure 2 presents a generic architecture of a project coordination support system discussed by Maurer [36].

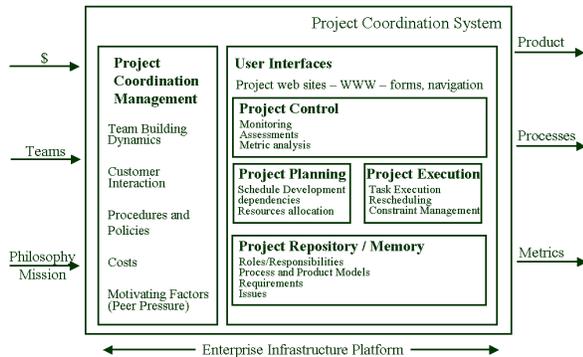


Figure 2. Maurer's Project Coordination Architecture

System inputs include budget, resources, and mission. System outputs include product, processes, and metrics. Metrics are used to evaluate project performance. The Project Coordination Management module handles the softer side of the PM, which mainly deals with personal interactions. There are four major components in the project coordination system:

- The project repository serves as a project memory: all information about the project is stored here.
- The project planning component allows users analyze dependencies between information items and plan the project in terms of time and resources.
- The project execution component supports workflow management by using the project plan. It allows re-planning and re-scheduling.
- The project control component supports monitoring of the project, allow users to assess the current state and collect metrics.

A web-enabled user interface is provided for all components so that users can access project web sites by using web browser.

This model does allude to collaboration, however it focuses only on the coordination level, and does not address the concerted level, wherein project team members work together to accomplish a shared goal [39]. This architecture goes further than the one presented by Dixon, but it still lacks integration of the components, high level collaborative support, and process support.

4.2. Collaborative Project Management Architecture

Both Dixon's and Maurer's models give an overview of the generic PM process and architecture, and they

clearly specify the inputs and outputs of the system. Their specifications of input and output encourage us to consider additional inputs to a PM system, and outputs produced by the system. Dixon's model illustrates how system functions and services influence one another. However his model does not include some important project contexts. On the other hand, Maurer's model is more comprehensive in that it includes both system functions and the supporting management context in which the functions work. Maurer's model lays out the system functions and services as modules, but does not specify how these modules are related with one another. These two models give us insights into what a top-level collaborative PM architecture should have: 1) the overview of the system, a clear specification of inputs and outputs; 2) the basic system functions and services; 3) the context in which system functions and services work; 4) specification about how different system functions and services collaborate and influence with one another. Neither architecture offers the level of collaborative or process support required for distributed projects. While Maurer provides a web interface, this is insufficient to support the complex processes and communication patterns that often emerge in distributed project environments.

Figure 3 illustrates our proposed Collaborative PM Architecture as consisting of four core components: Project Presence; Collaborative Support Levels, Project Knowledge Management, and the Project Cycle. Additionally collaborative middleware provides for communication among the core components and the tools within them. The system architecture is described at the conceptual level to provide an overall picture of requirements for a collaborative PM system.

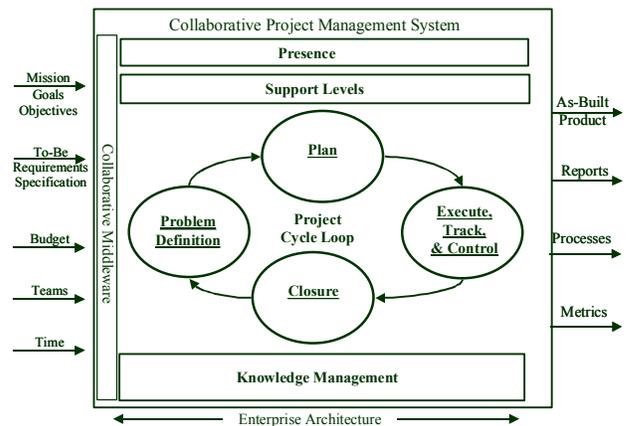


Figure 3. Collaborative Project Management Architecture (CPMA) (Adapted from [44])

System inputs include goals, objectives, to-be requirements specification, budget, teams, and time. System outputs include as-built product, reports, processes, and metrics. Our architecture considers more factors for input and output than the previous two models. As we discussed earlier in the paper, some common PM mistakes include ineffective communication, reactive management, and treating PM

as reporting mechanism. These problems have been addressed specifically by our architecture. By considering more inputs and outputs, project members have more metrics to clearly specify what resources are available, what requirements have to be considered, and what product criteria must be met. The analysis of these inputs and outputs can help plan the entire project at a detailed level up front in the project life cycle, more explicit communication of project information to all project contributors and which project measures and metrics need to be collected and tracked. The central feature is support for much more detailed and explicit metrics at a finer level of task granularity thus increasing the progress visibility and tracking accuracy. In the following sections we discuss each of the core components of our CPMA and the middleware that enables them to exchange information efficiently and effectively.

4.2.1. Project Presence

Presence can be defined as the sense of being within an environment and it refers to presence in natural settings [46]. Typically in projects that involve geographically distributed participants it is not possible for them to be physically co-present or co-located. One advantage of co-located PM is that it is easy for project members to develop a high level of project awareness, context, and progress by informal chat, email, phone, and face-to-face meetings. Misunderstandings of project information are easy to detect and correct. For example, people may use same term for different things, or they may use different terms for the same thing. If people are present at the same site they will consciously or subconsciously converge their understanding of term implication for different context. This is harder to achieve for distributed project members due to reduced frequency of communication and other difficulties. A collaborative PM system must facilitate distributed project members to develop high levels of project awareness through explicit knowledge representation. Phone calls and email are still needed, however, knowledge needs to be captured and stored permanently for easy retrieval. The following three components may help distributed project members gain a better shared understanding of project context.

- 1) **Project dictionary.** Where key terms, concepts, jargons, and methodology are defined and clarified.
- 2) **Business Rules And Policies.** Project members can explicitly specify project related rules and policies for all sites. For example, meetings notes should be taken for all meetings and should be available for all sites to review. Documentation should have author, date, and observe certain format rules. These rules and policies allow project members follow certain standards for project activities and document these activities for later retrieval.
- 3) **Project Context Information.** Project members have to be familiar with project context to be productive in the long run. Project background, boundary, objectives, and available resources (e.g. time, budget, equipment, and personnel) need to be documented and shared with all project members.

These three components may increase members' project awareness of context and increase the feeling shared project presence across the different locations. This module may be especially important when the project is conducted in organizations with very different organizational cultures.

Project presence addresses contextual awareness, which is more static than project activities or progress. Project activities involve different levels of coordination among project members and a collaborative PM system needs to support all coordination levels.

4.2.2. Collaborative Support Levels

We define *Collaboration* as making joint cognitive effort toward achieving an agreed upon goal. Collaboration can be represented as a hierarchy (Desanctis and Gallupe, 1987; Briggs & Nunamaker, 1994). As people collaborate, there are at least three modes in which they can work **collected work, coordinated work, and concerted work.**

4.2.2.1 Collected Work

At this level, each team member makes an individual effort. No coordination among members is required for the individuals to be productive. Group productivity is simply the aggregate of individual efforts. This mode of work is analogous to a team of sprinters, each of whom makes the best individual effort possible. The team productivity is simply the aggregate of individual productivity.

Processes are individual-centric start-to-finish and usually not integrated until the fruits of each individual's labors are collected. Therefore process structure and task structure can be low or nonexistent. The need for interactive communication cues is typically also quite low. Typical computer applications to support collected work are word processing, spreadsheets and graphics applications.

Typical PM scenarios at this level have been presented earlier in the article: project is a single-location project, tasks in the project are not coupled, or very loosely coupled. Every individual in the project needs to know his/her own job, and it is the project manager the responsibility to aggregate the final results. The coordination and communication among project members is very low. A simple project in a repeat, fixed environment may conduct at this level and can be managed by walking around physically. PM tool at this level should support: scheduling, resource allocation, task allocation, milestone process tracking, project-related information storage.

4.2.2.2 Coordinated Collaborative Work

At this level, team members still make individual efforts, but the success of some members may depend on the timely receipt of the deliverables produced by other members. Therefore, the success of the team depends on their ability to coordinate their efforts. This mode of work is like a team of relay runners, each of

whom makes their best individual efforts, but each of whom must execute a carefully coordinated hand-off of deliverables to the next runner.

This level of collaboration involves managing interdependencies between activities [34]. Coordinated collaborative processes tend to be ordered and characterized by hand-offs and progressive integration, thus task and process structure must be higher than for collected collaboration. The need for interactive communication also rises so that team members can agree on and monitor progress toward their coordinated hand-offs. Typical computer applications to support coordinated work include e-mail, team calendaring, and workflow automation. This level differs from the collective level in that it is more structured in terms of process and specific milestones and handoffs. It is worth noting that formalized coordinated work processes constitute an important constituent of an organization's Intellectual Capital.

PM at this level requires coordination among project individuals. PM tool should support, in addition to all collected collaboration functions: group calendaring, task interdependence analysis, timely change notification for appropriate users, easy access to project-related information, and routine process tracking.

4.2.2.3 Concerted Collaborative Work

At this level, all members of a team must contribute in concert to the group effort, and performance of any one member influences the ability of all other members to perform. A rowing team is a useful metaphor for this level of work. All rowers must synchronize their efforts and contribute simultaneously to succeed in the race. If one member falls out of synchronization, none can perform adequately. A sailboat racing team is another apt metaphor. Each member of the crew must perform a different task, but all must perform in concert or the boat cannot win the race. An aggregation of uncoordinated, individual efforts would yield nothing.

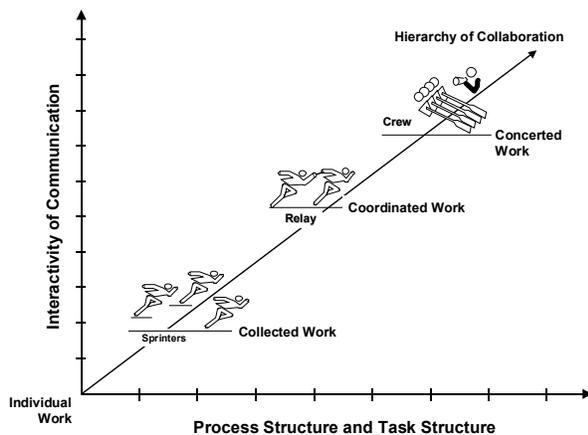


Figure 4. The Hierarchy of Collaboration

Task and process structure must be far higher for concerted work than for coordinated work, because almost any behavior by one team member immediately affects the productivity of others, and the need for interactive communication (verbal or non-verbal) may be nearly continuous. Tools to support concerted collaboration must support and accommodate attention dynamics, and make it easier for people to maintain task focus. There are not individual results to be aggregated or handed-off. Rather, is a result produced by the joint efforts of all team members in concert. Group Support Systems (GSS) are a commonly used example of technology to support this level of work [39].

PM at the concerted level requires tight coordination among project individuals. PM tool should support all functions mentioned at the collected and coordinated level. And it will provide some more advanced functions: explicit documentation of process, document version control, multiple users co-authoring a document, linking relevant information together. At this level, a user can search, retrieve, update, and upload documents according to the predefined user role. The balance of information overflow and underflow is achieved by enforcing information access policy based on user roles. At the coordinated level, the document management may be conducted by the project manager, and project manager make documents accessible to other members. At the concerted level, every project member has the responsibility to manage the documents and privileges to view relevant documents. We will document process and task interdependence through GSS templates and group memory in our implementation. A Collaborative PM system must be designed to support all three collaboration levels. This section only specifies how to support collaboration at a conceptual level; the Project Cycle section will specify what activities or group processes require collaborative support.

4.2.3. Project Cycle

We have discussed that project cycle has four major steps, each step has its own group activities and deliverables. If the project manager and members can identify the possible group activities and deliverables for each step in a problem domain and specify the level of collaboration needed they may be able to standardize the project process and use technology to facilitate the execution of these repeatable processes. We identify some general activities that need to be accomplished at each step, different project may have variations for these steps.

Step 1 is to gain a clear understanding of the project. Sense-making and decision-making activities are typical at this stage such as identifying the project scope, objectives, key stakeholders, and the gap between the current situation and ideal situation (the gap between "As Is" and "To Be"); estimating resource needs for the project (e.g. money, time, and personnel); analyzing solution alternatives and evaluating the solution alternatives; and conducting risk analysis. The thorough understanding of the project helps

communicate the team goal to all key stakeholders and achieve the team goal congruence, which is very important for project success [10].

Step 2 is to make a plan to achieve the project goals. Typical activities are analysis and decision making activities, such as breaking down the project into manageable tasks and subtasks, analyzing the interdependency of tasks, forming the project team, assigning resources and tasks to team members, defining milestones for the project, making project schedule, defining progress measurements, planning risk management and change management, forming the communication plan (how project activities are to be communicated among key stakeholders), and setting up Project Notebook which consisting of all project related documents.

Step 3 is to execute the project plan, collect project progress information, execute risk and change management, update and maintain the Project Notebook. As we have discussed, this part is the most dynamic part in PM, and least research has been done for this part. A collaborative PM tool would greatly enhance the project tracking ability.

Step 4 is to identify the sign-off criteria; reflect upon the project process – what went right, and what went wrong; and compare the initial project planning with the actual project process and identify possible improvement if the identical project will be conducted in the future.

Steps 1, 2, and 4 involve project understanding and planning and all project members should be involved. These steps require coordinated and concerted level of collaboration support. GSS technologies have tools to facilitate group process such as divergent thinking, convergent thinking, group discussion, group negotiation, and group writing. A collaborative PM platform would have specific tools to facilitate a single piece of group processes and provide the flexibility to assemble the tools in such a way that meet the process sequence.

Step 3 requires all three levels of collaborative support. For example, email can be used to check a task status from individuals, the bulletin board can be utilized to post the task status for the sub-groups, and GSS tools can be used to track the entire project progress and discuss issues that affect the entire group

Unlike other modules, Project Cycle does not specify system functions rather it specifies contents that need collaborative support. How to map different PM issues and contents into different support levels is beyond the scope of this short paper.

4.2.4. Collaborative Knowledge Management

In this section we will discuss definition of knowledge management (KM), the difference between PM and KM, and the importance of KM to PM. An electronic project repository may focuses on manage data,

information, and knowledge related to a single project. An KM module tends to focuses on manage data, information, and knowledge at the organizational level. Project managers and members can establish baselines for project execution, compare and contrast the multiple projects across time and derive useful patterns of PM.

Davenport and Prusak (1998) [14] defined knowledge as “*a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluation and incorporating new experiences and information.*” There are at least two types of knowledge: tacit knowledge (to know how) and explicit knowledge (to know about facts and theories) [8, 28, 38, 41]. Knowledge Management (KM) can be defined as “*the process of acquiring, creating, sharing, and using knowledge*” [28]. Many researchers have developed knowledge hierarchies to explain how data, information, knowledge and wisdom are related [2, 4, 3, 6, 9, 39, 45, 52], but few have discussed this in terms of PM [28].

Katzy et al. [28] presented two differences between PM and KM. First, PM is a finite effort for a specific period of time, while KM is ongoing and knowledge should be maintained as long as it is useful. Second, PM is goal oriented, on the other hand, KM is not necessarily an end in and of itself. Knowledge is created and modified as project activities occur, and the context of knowledge creation and application are important. Projects make KM necessary across time and contexts [28]. Different activities help knowledge Sharing and conversion. Communication allows people to exchange tacit knowledge, externalization is to turn tacit knowledge into explicit knowledge, internalization is to turn explicit knowledge into tacit knowledge, and combination is to integrate implicit knowledge with explicit knowledge [28, 38]. A KM tool can help all these knowledge generation and conversion activities.

KM specifies rules and provides functions for information gathering, access, update, retrieval, organization, and archival. It also provides functions for information integration from different sources. The actual data and information are stored in a document repository in a variety of document formats. People can use the document repository to manage multiple projects over time. The document repository provides an online access for project team members. They can upload document of a variety of formats into the repository, and they can search, view, edit, and reload a document depending on the roles they are granted. KM provides functions to transfer data from one source to another, for example, importing meeting contents from a GSS tool into the document repository or archiving important email exchanges as text files. By collecting data and information from multiple projects KM allows project managers to compare or aggregate information across projects to derive patterns and thus create knowledge.

4.2.5. Middleware for Collaborative Project Management

In the previous section, we noted that under certain circumstances, GSS might substantially improve the performance of project team members engaged in concerted collaboration. Complex projects that require concerted collaboration are likely to require that team members exchange and use many diverse kinds of information, but current GSS environments tend to be monolithic, closed systems that offer users little ability to access data from outside applications. Gregory and Briggs [21] proposed a middleware architecture for collaboration that may be usefully extended to the larger task of PM. Middleware is software that mediates transactions between client applications and data repositories, or that mediates transactions among various client applications. The Gregory-Briggs [21] middleware architecture is based on a Universal Data Model (UDM). The UDM employs a relational model, which allows metadata to be established on the server at database design time. Matching metadata is built into client queries. No a priori metadata exists on the server with the UDM. The client establishes its own metadata on the fly, and submits metadata with its data at run time. Digital objects of any type may be contributed to a server along with any number of relationships of any type with any other elements stored in the repository or contributed simultaneously. Clients that create similar metadata may share data; clients that create dissimilar metadata may not. Data from a UDM server can be transformed to relational data on the fly if the need arises. A GSS built on a UDM might be able to accommodate the diversity of data structures, digital documents, and data from third party application, that are likely to be required for successful PM, such as in distributed projects, where it is unlikely that data structures and document types of all possible projects and tools can be known in advance.

The very collaborative nature of PM, makes the five-layer middleware integration called for in the Gregory-Briggs [21] architecture advantageous as the foundation for a collaborative PM system: their approach calls for optimizing collaboration by combining into a single server Object Oriented Middleware (OOM), Data Access Middleware (DAM), Transaction Processing Middleware (TPM), Message-oriented Middleware (MOM) and Remote Procedure Call (RPC). (A detailed description of the middleware can be found in [23], [22], and [21]). Their [21] middleware approach works as follows: collaborative interfaces serve as filtered views of the data set. Each collaborative interface communicates with the server using a publish-and-subscribe push capability (a feature of MOM). Under this scheme, each interface registers with the server, and subscribing to certain data types within certain data sets (Ex. From the Smith_Inc. data set give me all data-elements of type "Action_item", and give me all data elements of type "Deliverable" that are child-of those business processes). The server then checks all incoming data elements to see if they match the subscriptions of any client applications. The server forwards any elements that match subscription criteria

to waiting client applications. Data access middleware exchanges data with clients and third-party applications, assuring that transactions commit correctly to the Universal data model and that the UDM persists correctly to permanent data stores like an enterprise database. It provides conversion services among applications with different data models. The RPC mechanism allows task-critical business logic to be stored and executed by the server, helping keep clients thin, centralizing quality control for critical code, and enabling the reuse of code for critical capabilities

4.3 Present CPM Tools

Many companies claim they have collaborative PM tools, and these tools support different levels of collaboration. Table 1 provides a brief view for some tools: company name, PM tool name, project types the tool supports, and the collaboration level the tool supports.

Company Name	PM Tool Name	Category of project	Collaboration level
Rational.com	Rational SoDA	Software project	Collected
OnProject.com	OnProject	Business project	Coordinated
Citadon.com	ProjectNet ProjectNet Process	Project of Engineering, Building, and Real Estate	Collected
Surdex.com	CPMS (Collaborative Project Management System)	photography and mapping services	Collected
Viecon.com	ProjectBank ProjectWise	Engineering	Collected
Microsoft.com	Microsoft Project 2000	Any kinds of project	Coordinated

Table 1. Collaborative PM Software Levels

From table 1, we can see that most of the collaborative PM tools support the lower levels of collaboration. Information sharing and scheduling are the focuses for these tools. Task interdependence analysis, process documentation, timely change notifications, and document management at the user level are left out in these systems.

4.4 CPM Architecture Summary

Our CPM architecture is described at the conceptual level. We plan to evaluate it by using five major PM themes that have been discussed at the beginning of this paper. Our CPM architecture does not treat PM as merely a reporting mechanism, instead it treats PM as the tool to analyze task interdependence, track task progress, and to share and organize information. CPM facilitates effective and efficient communication by explicitly documenting project related concepts, rules, procedures, and processes. Telephone calls or email exchanges can be augmented to increase communication effectiveness, however, the contents of phone calls and email exchanges should be documented and saved in the CPM document repository.

Traditional PM focuses on managing project input, output, but not the process. Our CPM implies that process will be explicitly captured and stored permanently. The group meeting processes can be captured by GSS tools and exported to document repository. Project members are responsible to capture their work deliverables and the underlying processes, as well as update others about task status and keep abreast of the status of tasks performed by others on the project team. Not only are the project team members distributed geographically, but so too must the decision-making power, the responsibility of project and process quality, and the flow of information be distributed among all members. The project information is stored and organized in the project repository to help members easily access the information, get new project members familiar with the project quickly, and facilitate project learning and memory over time. Our CPM allows active and proactive PM by utilizing task interdependence analysis, change notification, and all levels of collaboration

5. Discussion

Traditional PM practices are inadequate to address the communication, collaboration, and information-sharing challenges that arise from distributed or virtual projects. The new PM paradigm requires a quite different PM methodology from traditional approaches and a much more robust IS to support PM. We have proposed a Collaborative PM Architecture (CPMA) that builds on two previous PM architectures and integrates the important new components of project presence, collaborative project team support, and collaborative middleware. Our proposed architecture provides a foundation for the components needed to build a collaborative PM system: inputs and outputs, system functionality, and the relationships among the functions.

Our PM architecture integrates five modules: Project Presence, Collaborative Support Levels, Knowledge Management, the Project Cycle, and Collaborative Middleware. Our future research will involve developing and field-testing a prototype CPM System built on top of our CPM architecture. We will perform action learning and research to provide PM interventions and real-time prototype enhancement to improve and extend the level of collaborative PM support that can be provided for distributed projects. We believe that that the resulting system will provide CPM support beyond what current tools and environments are capable of supporting and will enable levels of efficiency and effectiveness in CPM that will make distributed projects more successful in both the short run and over time.

References

1. *Distributed Project Management: A Marketplace and Software Vendor Analysis*. 2000 Online at: http://www.collaborate.com/announcements/announce_3.html.
2. Ackoff, R., *On Learning and Systems That Facilitate It*. Center for Quality of Management Journal, 1996. 5(2 (Fall)): p. 27-35.
3. Ackoff, R.L., *From Data to Wisdom*. Journal of Applied Systems Analysis, 1989. 16: p. 3-9.
4. Ackoff, R.L., *Management misinformation systems*. Management Science, 1967. 14(4): p. B-147-B-156.
5. Allen, T., *Managing the Flow of Technology*. 1977, Cambridge MA: MIT Press.
6. Alter, S., *Information Systems: A Management Perspective*. 3rd ed. 1999 (September), Auckland, NZ: Addison-Wesley Pub Co.
7. Back, W.E., *Information Management Strategies for Project Management*. Project Management Journal, 2001. 32(1): p. 10-20.
8. Bair, J.H. and E. O'Connor, *The state of the product in knowledge management*. Journal of Knowledge Management, 1998. 2(2): p. 20-27.
9. Beckman, T.J. *A Methodology for Knowledge Management* Proceedings of the International Association of Science and Technology for Development (IASTED) AI and Soft Computing Conference. 1997. Banff, Canada.
10. Briggs, R.O., *The Focus Theory Of Group Productivity And Its Application To Development And Testing Of Electronic Group Support Systems*, in *Department of Management Information Systems*. 1994, University of Arizona: Tucson.
11. Callahan, J. and S. Ramakrishnan. *Software Project Management and Measurement on the World-Wide-Web (WWW)* Proceedings of the 5th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'96). 1996. Stanford, CA: IEEE Computer Society.
12. Clarke, A., *A practical use of key success factors to improve the effectiveness of project management*. International Journal of Project Management, 1999. 17(3): p. 139-145.
13. Cleetus, K.J., G.C. Cascaval, and K. Matsuzaki. *PACT - A Software Package to Manage Projects and Coordinate People* Proceedings of the Proceedings of the 5th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'96). 1996. Stanford University, California, USA.
14. Davenport, T.H. and L. Prusak, *Working Knowledge: How Organizations Manage What They Know*. 1998, Boston, MA: Harvard Business School Press.
15. Dixon, D. *Integrated support for project management* Proceedings of the Proceedings of 10th International Conference on Software Engineering. 1988. Singapore. p. 49 - 58.
16. Evaristo, R. and P.C. van Fenema, *A typology of project management: emergence and evolution of new forms*. International Journal of Project Management, 1999. 17(5): p. 275-281.
17. Ewusi-Mensah, K., *Critical issues in abandoned information systems development projects*. Communications of the ACM, 1997. 40(9): p. 74 - 80.
18. Fenema, P.C.V. and K. JKumar, eds. *Coupling, interdependence and control in global projects*. Projects as business constituents and guiding motives, ed. R.A.Lundin and F. Hartman. 2000, Kluwer Academic Publishers: Boston.
19. Graham, R.J. and R.L. Englundm, *Creating an Environment for Successful Projects*. Jossey-Bass business & management series. 1997, San Francisco, CA: Jossey-Bass Inc.

20. Gray, J.L., Hazeltine, N., and Lynch, J., *Bridging islands of information with cooperation software*. AT&T Technology, 1992. 7(1): p. 8-33.
21. Gregory, T. and R.O. Briggs, *An Approach to Middleware for Repeatable Collaborative Processes*, in *Thirty-Fifth Annual Hawaii International Conference on Systems Sciences*, J. Ralph H. Sprague, Editor. 2002, IEEE Computer Society Press: Kona, Hi.
22. Hailstone, R., *Western European Middleware, Businessware, & Application server Market Overview, 2000-2006. Bulletin #WES33H*. 2001, IDC.
23. Heinman, R.V. and D. Byron, *IDC's Software Market Taxonomy. Report # 23998*. 2001, IDC.
24. Helbrough, B., *Computer assisted collaboration - the fourth dimension of project management?* International journal of project management : the journal of the International Project Management Association, 1995. 13(5): p. 329-333.
25. Jarvenpaa, S.L. and B. Ives, *The global network organization of the future: information management opportunities and challenges*. Journal of Management Information Systems, 1994. 10(4): p. 25-57.
26. Jarvenpaa, S.L., Knoll, K., & Leidner, D.E., *Is anybody out there? Trust in global virtual teams*. Journal of Management Information Systems, 1998. 14(4): p. 29-64.
27. Jonsson, N., D. Novosel, J. Lillieskold, and M. Eriksson, *Successful Management of Complex, Multinational R&D Projects*. in J. Ralph H. Sprague, (Ed.) *Proceedings of the Thirty-Fourth Annual Hawaii International Conference on Systems Sciences*. 2001. Maui, Hawaii: IEEE Computer Society Press.
28. Katzy, B., R. Evaristo, and I. Zigurs, *Knowledge Management in Virtual Projects: A Research Agenda*. in R.H. Sprague Jr., (Ed.) *Proceedings of the Thirty-Third Hawaii International Conference on System Sciences*. 2000. Maui, HI USA: IEEE Computer Society.
29. Keen, P.W., *Shaping the future*. 1991, Boston: Harvard Business School Press.
30. Kingman, L.C., R.E. Lambert, and R.P. Steen, *Operational image systems: A new Opportunity*. IBM Systems Journal, 1990. 29(3): p. 304-312.
31. Lam, H.E. and P. Maheshwari, *Task and Team Management in the Distributed Software Project Management Tool*. Proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC.01). 2001: IEEE Computer Society.
32. Lewis, J.P., *Fundamentals of project management : developing core competencies to help outperform the competition*. 2nd ed ed. 2001: New York: AMACOM.
33. Luiten, G.T. and F.P. Tolman, *Automating communication in civil engineering*. 1995, CIFE working paper 038. Stanford, CA: Stanford University.
34. Malone, T. and K. Crowston, *The interdisciplinary study of coordination*. ACM Computing Surveys, 1994. 26(1 (January)): p. 87-119.
35. Marttiin, P., J.A. Lehto, and G. Nyman, *Understanding and evaluating collaborative work in multi-site software projects - a framework proposal and preliminary results*. in R.H.J.S.a.J.F.J. Nunamaker, (Ed.) *Proceedings of the Proceedings of Thirty-Fifth Annual Hawaii International Conference on Systems Sciences*, . 2002. Wikoaloa Village Kona, Hi: Hi, 2002 CDROM, IEEE Computer Society Press.
36. Maurer, F. *Working Group report on Computer Support in Project Coordination*. Proceedings of the the Project Coordination Workshop of the IEEE Fifth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE). 1996. Standford University, CA, USA.
37. McConnell, S., *Rapid Development*. 1996, Redmond: Microsoft Press.
38. Nonaka, I., H. Takeuchi, and H. Takeuchi, *The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation*. 1995, New York, NY: Oxford University Press.
39. Nunamaker, J.F., Jr., N.C. Romano, Jr., and R.O. Briggs, *Increasing Intellectual Bandwidth: Generating Value from Intellectual Capital with Information Technology*. Group Decision and Negotiation, 2001-2002. 11(2 (March)): p. 69-86.
40. Nunamaker, J.F., Jr., Dennis, A. R., Valacich, J. S., Vogel, D. R., and George, J. F., *Electronic meeting systems to support group work: theory and practice at Arizona*. Communications of the ACM, 1991. 34(7 (July)): p. 40-61.
41. Polanyi, M., *The tacit dimension*. 1967, New York, NY USA: Doubleday.
42. Port, D. and G. Kaiser, *Collaborative Technologies for Evolving Software Systems*. IEEE Internet Computing, 1998. 2(6 (November)): p. 79-83.
43. Ramamurthy, K. and G. Premkumar, *Determinants and outcomes of electronic data interchange diffusion*. IEEE Transactions on Engineering Management, 1995. 42(4): p. 332-351.
44. Romano, N.C., Jr., F. Chen, and J.F. Nunamaker, Jr. *Collaborative Project Management Software*. in R.H.J. Sprague and J.F.J. Nunamaker, (Eds.) *Proceedings of the Proceedings of Thirty-Fifth Annual Hawaii International Conference on Systems Sciences*. 2002. Wikoaloa Village Kona, Hi.: 2002 CDROM, IEEE Computer Society Press.
45. Sena, J.A. and A.B. Shani, *Chapter 8 - Intellectual Capital and Knowledge Creation: Towards an alternative Framework*, in *Knowledge management handbook*, J. Liebowitz, (Ed.). 1999, CRC Press: Boca Raton, Fla. p. 8-1 - 8-16.
46. Steuer, J., *Defining virtual reality: Dimensions determining telepresence*, in *Communication in the age of virtual reality*, F. Biocca, and Levy, M. R. (ed.), (Ed.). 1995, Lawrence Erlbaum: Hillsdale, NJ. p. 33-56.
47. Surubhotla, S., *Project Management*. 2001, <http://www.rushmoreu.com>.
48. Taylor, B.W., *Cataloging: The first step to data management*. Society of Petroleum Engineers (SPE) 24440, 1992: p. 193-197.
49. Teicholz, P. and M. Fisher, *Strategy for computer integrated construction technology*. Journal of Construction Engineering and Management, ASCE, 1994. 120(1): p. 117-131.
50. Tollmar, K., O. Sandor, and S.A. . *Supporting social awareness @work - design and experience*. Proceedings of the Computer-Supported Cooperative Work (CSCW) 1996. 1996: ACM Press. p. 298-307.
51. Turner, J.R., *Do you manage work, deliverables or resources?* International Journal of Project Management, 2000. 18(2): p. 83-84.
52. van der Spek, R., and Spijkervet, A., *Knowledge Management: Dealing Intelligently with Knowledge*, in *Knowledge Management and Its Integrative Elements*, a.W. Liebowitz J., L. C., (Ed.). 1997 (August), CRC Press.
53. Weisband, S., *Chapter 13: Maintaining Awareness in Distributed Team Collaboration: Implications for Leadership and Performance*, in *Distributed Work*, P.H.a.S. Kiesler, (Ed.). 2002: MIT Press, Boston, MA, USA.
54. Weiser, M. and J. Morrison, *Project Memory: Information management for project teams*. Journal of Management Information Systems, 1998. 14(4): p. 149-166.