1

Designing Groupware Applications: A Work-Centered Design Approach

KATE EHRLICH
Lotus Development Corp.

ABSTRACT

Group-ware is about group-work. It is about developing technologies that support the way people communicate and collaborate to accomplish work goals in the context of personal, managerial and organizational imperatives. In contrast to single user applications which support peoples' tasks, groupware supports peoples' work. Tasks are often explicit, observable and concrete. Work is often tacit, invisible and amorphous. The challenge in developing a groupware application lies in understanding, explicating and then supporting the invisible work. The chapter provides some insight into the process of developing groupware applications by first describing some new methodologies for generating requirements. It then outlines several themes — communication, awareness, anonymity — which have emerged as common across many groupware applications. The chapter also addresses the technical and social issues that emerge when deploying a groupware application in an organization. Application deployment is one of the most challenging aspects of developing a groupware application. The chapter concludes with a case study of an application designed to support coordination and communication in distributed teams. The case study brings together the topics of Methodology, Design and Deployment in a concrete setting.

1.1 INTRODUCTION

Group-*ware* is about group *work*. Group work is the work practice that evolves to get ordinary, daily work done. Group work includes the informal ad hoc communication that happens between people in adjoining offices or people in different countries and time zones. Group work

Computer Supported Cooperative Work, Edited by Beaudouin-Lafon

happens in a context of personal, managerial and organizational imperatives that encourage people to share their work with others and reward them when they do.

The design of single-user applications translates users' tasks and needs into a functional description which directs the overall design and development of the application. When delivered, most single user applications can be used "right out of the box" — aside from time spent learning the application. Groupware applications, designed to support group work, require a different methodology to understand the tacit, invisible aspects of work practices. Translation into cogent, explicit requirements is not straightforward but requires extensive ongoing collaboration between researchers and application developers to translate descriptions of group work into application features. The functional requirements govern the technical development of the application. But adoption of the application is just as likely to be determined by organizational and managerial preparedness as by the design and technical implementation of the application itself.

This chapter draws on recent research and practical examples to examine groupware applications from three perspectives:

- *methodologies* for providing new product ideas or for extracting requirements from work practices
- common themes that emerge in the *design* of many groupware applications
- technical and social challenges in *deploying* an application.

The chapter concludes with a case study of the design, development and deployment of an application to support coordination in distributed teams.

1.1.1 Definitions and examples of groupware applications

Groupware, the applied side of CSCW (Computer Supported Cooperative Work), has been described by Cameron et al. [Cam95] in the Forrester report as:

> " Technology that communicates and organizes unpredictable information, allowing dynamic groups to interact across time and space."

And by Bob Johansen of the Institute for the Future as:

> "a generic term for specialized computer aids that are designed for the use of collaborative work groups. Typically, these groups are small, project-oriented teams that have important tasks and tight deadlines. Groupware can involve software, hardware, services and/or group process support." [Joh88, page 1]

Groupware applications provide computer support for group work. At a general level, group work includes *written and spoken communication*, *meetings*, *shared information*, and *coordinated work*. Some group work occurs when people interact with each other at the same time (synchronously). Face-to-face meetings are an example of people working together at the same time and often in the same place. People can also work together at different times (asynchronously). When people leave messages in electronic mail, the communication occurs over a period of time.

1.1.1.1 Communication

Perhaps the most common type of group work is communication between individuals or groups. Groupware applications to support *synchronous* communication includes videoconferencing, shared screens/applications, MediaSpaces, Chat (see Chapters 3, 4 and 5 in this book [Mac99, Ish99, Pra99]). These applications let people communicate with each other even though one person(s) is located at a different place than the other person(s), by using technology to link separate screens. For instance, when people communicate using videoconferencing a camera pointed at one person's face can relay that image and any sounds to the screen of someone sitting at another computer. That other computer could be located down the hall, or in another city or country. In this way people who cannot be physically in the same place at the same time can still communicate with each other.

Applications to support *asynchronous* communication include electronic mail, perhaps the most widely used groupware application. Electronic mail lets people leave messages for one or more other person at any time to be read by that person at any time and in any place.

Other things can emerge from communication such as the development of virtual communities whose continued communication/participation is then further supported by technology.

1.1.1.2 Meetings

One of the most common work activities in most organizations are face-to-face meetings. While it might seem that this is one place where there are no barriers of time or place, there is still opportunity for applications to support work. Groupware applications to support meetings include software that captures and organizes ideas for brainstorming, summarization and reporting. This software is most often used in specially equipped rooms with computers embedded in desks. People attending one of these meetings can enter ideas, comments, votes into the computer at particular times during the meeting. The software might simply display the written ideas, let someone, usually a meeting facilitator, group the ideas, or the software might tally votes. All of which supports the work of the meeting in a way that goes beyond what the people in the meeting could do on their own.

1.1.1.3 Information Sharing

When people work together, there is often a need not only to communicate with others but to share information. Information is commonly shared by leaving an electronic document in a database where it can be read by anyone with access to that database. This is different from electronic mail where a document is sent to a particular person or sometimes a group of people. Groupware applications include discussion databases, bulletin boards and electronic news groups where documents and their responses are often grouped together under a single heading or keyword making it easier to follow the thread of a discussion.

Applications for publishing documents have sometimes also been called groupware. However, this labeling is somewhat controversial. This chapter takes the position, that an application can be considered groupware only when more than one person has the opportunity to create documents or other responses. Applications that let one person publish documents to a wide audience of readers are not good examples of groupware.

1.1.1.4 Coordinating Work Processes

In many operational settings, people coordinate their work over time making sure that decisions made by one person are acted upon before being passed on to the next person. Commonly referred to as workflow systems, these applications often embody features such as privacy control, sequencing, notification, and routing (see also Chapter 2 in this book [Ell99]). Workflow systems may also include a decision support component as part of the overall process. Workflow systems are often thought of in the context of formal approval processes or large production settings. Systems designed to support coordination between people include project management, tracking systems, shared calendars. As defined here, workflow could also describe the kind of system used in medical settings where several people need to interact with the same data in handling administration of patient records. While not necessarily formally acknowledged as workflow or groupware, these kinds of applications meet the criteria of having multiple people interacting with the same application and hence the need on the part of the application developer to pay attention to issues of access, privacy and simultaneity of use.

Forrester [Bro97] gives this description of a groupware application built by a computer company to better manage projects:

> "We are the professional services division of a major computer company. Time is our enemy. The longer a project goes, the worse things are. We're building a project management system that puts the project plan on the Internet for feedback, updates, notation, and comments from all team members, including customers. Before, one person managed the project with relatively static plans, but with this program, project management becomes much more dynamic and fluid."

1.1.1.5 Groupware Solutions

In addition to applications designed to solve particular problems, there are also efforts such as digital libraries, electronic commerce, knowledge management and distance learning, all of which include some degree of groupware within a larger context. It is often the case that the scale of these efforts requires a stronger "solution"-based approach in which technology is embedded within technical and professional services such as training and management consulting. These "solutions" are often distributed through consulting and other service organizations who are best suited to adapt the solution to the customers' needs and to handle the larger scale effort.

1.2 REQUIREMENTS

This chapter assumes a phased approach in the development of a new groupware application. In broad terms, the first phase, called *requirements*, concerns the translation of the users' needs into a functional specification from which a detailed design can be made. The second phase, called *design*, focuses on what gets built and how. The third phase, called *deployment*, attends to the details of introducing and rolling out the application to the customer or user.

In gathering requirements for the development of a new groupware application, the focus should be on understanding the invisible work and work practices [Nar98a, Suc95] as well as the visible, and on understanding the physical and organizational context in which work is done. That work context is increasingly nontraditional — homes, airports, train stations, hotel rooms; anywhere where a laptop computer — or a fax or phone — can be carried and attached to a wide area network such as the Internet. If members of a group frequently work apart, both the technology and the culture of the group needs to accommodate a different style of interaction and coordination, as O'Hara-Devereaux and Johansen [OHa94] emphasize:

> "Global organizations cannot function without information technology. But the technology itself is not the answer to the myriad problems of working across geographical and cultural boundaries. The ultimate answers to these problems remain in the realm of human and organizational relations." [page 74]

It should be noted that the emphasis on groups, work, and work practices should not obscure the need for some level of task analysis within the application design. Workflow applications may have specific tasks embedded within them that are amenable to conventional task analysis methods. Similarly, applications originally designed for single users may be appropriated by the group. For example, spreadsheets, originally designed for individual use, were found to be part of an overall group collaboration (e.g. [Nar93]), especially in sharing expertise and generating alternative scenarios. Groupware applications also need to be deliberate about the design of the user interface so that each person interacting with the application understands how to *use* the application as well as understanding the meaning of the work that is being supported by the technology. Designing for ease of use is as much a part of the design of groupware as it is for single-user applications.

1.2.1 Work Practices

Before plunging into the methodological approaches toward collecting requirements, it is important to understand what is meant by work *practices* and how these might differ from what is thought of as work *processes* or routine work.

Most groups engage in some degree of routine work that appears predictable, is often thought to be tedious, may be time-consuming, and is often error-prone. Some organizations, such as insurance companies, make a business out of such routine work by, for instance, processing insurance claims. This would appear to be an area in which computer support could reduce the amount of uninteresting work that people do, reduce error rates, provide accurate, up-to-the-minute status information and save money. Because the work is routine, it should be relatively straightforward to write down the sequence of steps that make up these workflows.

However, it turns out that even in simple cases, the work is never simply "routine" (e.g. [Mul95]) even though the people doing it might describe it that way. When Suchman [Suc83] studied accounting clerks, they described their jobs in a way that corresponded to the formal procedures. However, when Suchman observed these clerks, it was clear that they relied on informal, locally determined practices to get their work done. These practices were not written down anywhere nor were they part of any explicit training that the clerks received. Rather these practices were learned on the job.

If the application fails to support local work practices, people will either stop using the application or develop workarounds so that they can continue to work in a way that has evolved to be effective and efficient. People may also resist adoption if the application is perceived as compromising core skills and competencies. The two examples below, from a rich literature of work practice studies (e.g. [Pyc96, Sta94, Hug92, Hea91]), illuminate what can happen when systems are designed and delivered based on externally generated processes rather than the actual work practices.

1.2.1.1 Printing

Bowers et al [Bow95] describe work done by a large printer with several offices around Britain. The organization used both traditional print technology, such as hot metal presses and offset lithography, as well as newer technologies, such as high-end photocopying and digital reprographics. From an outsider's point of view, printing might seem to be a well-known and somewhat routinized process that should be easily described in terms of the sequence of tasks and movement of materials. As such, the work of printers might be considered amenable to some level of automation through, for instance, a workflow application that would handle some of the administrative work by which jobs are categorized by type and assigned codes, customers, delivery dates and so forth. The application might further capture data about length of time and type of materials, as well as control some of the more routine operations so that the operator had more time to handle other parts of the job.

From the point of view of the people doing the work, however, the operations and their sequence are anything but routinized. In fact, close inspection of the work reveals that the print operators evolved numerous small but significant modifications to the normative sequence of operations to ensure a smooth and efficient flow of work. These practices included: prioritizing the work, anticipating the work, supporting each other's work, knowing the idiosyncrasies of the machines, identifying and allocating interruptible work. For instance, each print job, which often involved multiple processes and different people, was accompanied by a "docket" marking details of the job, such as materials required, cost code, and desired delivery date. This docket would get transferred with the print job from operator to operator. Operators were supposed to order these dockets in terms of delivery date and select the next print job with the earliest date. However, in practice the operators would ensure a smooth flow of work by sometimes juggling these jobs based on complexity of job, how long it would take and whether there were other time-consuming processes later on, as well as factoring in jobs remaining from the previous day. In fact, the digital reprographics technology used in the print industry meant that simply following date order would not utilize the equipment efficiently, requiring, for instance, frequent changes of paper type or size or long idle times following a short print job if the operator was busy with another part of the job.

The operators were very familiar with the competing demands on the equipment and other resources and had evolved practices which adapted the normative ordering to the situation enabling them to conduct their work smoothly and efficiently. Moreover, these variations on the explicit process were well understood by all the operators and the administrative staff and constituted, as it were, their shared and distributed cognition of the work.

This group had a contractual obligation to install and use a workflow application designed specifically for the print industry, although not necessarily for this particular group. The application was designed to automate many of the routine administrative tasks while also maintaining a record of time taken on a job, materials used etc. Such information was useful in preparing reports and maintaining stock control. By contrast with the efficient smooth flow of work that had evolved in practice, the imposition of the workflow application disrupted the smooth flow of work by requiring that all print jobs be handled only in a normative fashion. For instance, the application required that no job could be started until an order form had been submitted. While this is the correct procedure, in practice, the operators would often jump the gun and begin the work in order to utilize the equipment effectively. The method of recording jobs and time failed to take into account that an operator could be doing multiple jobs at the same time. Using the system generally demanded extra time by everyone.

Because the operators had a contractual obligation to use the application, ignoring it and returning to the familiar method of working was not an option. Instead, they responded to the system by either developing work arounds, or, in extreme cases, reorganizing the work itself to adapt the work to the system. In either case, the overall work was done less efficiently.

As Bowers et al [Bow95] expressed:

> "Workflow from within characterises the methods used on the shopfloor which emphasise the local and internal accomplishment of the ordering of work. Workers juggle their in-trays, jump the gun, glance across the shopfloor, listen to the sounds coming from machines, redistribute the work in the here and now so that what to do next can be resolved. ... In contrast, workflow from without seeks to order the work through methods other than those which the work itself provides."

When technology makes things worse, not better, there are various approaches to the redesign: a) features in the application should have more flexible mappings between processes and operators; b) redesign the application with greater emphasis on awareness and mutual monitoring; c) acknowledge real management practices and pressures to adopt technology and adapt some of the practices.

1.2.1.2 Trouble Ticketing System

A similar example of a well intentioned groupware application failing to embody the actual work practices comes from Sachs [Sac95]. She describes a system intended to improve the efficiency of assigning work to telephone company workers who are called in when there is a problem with a phone line. The system acts as a general dispatcher routing job tickets to the office nearest to the person to whom the work has been assigned. The job ticket gets recorded and is available for the worker to pick up. When one job is finished the worker picks up the next ticket in the stack. This dispatch function was one part of a larger system which also handled scheduling, work routing and record keeping.

While this method would seem to make sense and help increase the efficiency of getting information to the workers in a timely fashion, in practice it failed to acknowledge critically important information. When a linesman picked up a ticket he/she would spend some time talking with the coworker. During the conversation, the linesman would pick up incidental information such as useful phone numbers, prior history as well as a more detailed explanation, of the actual problem. These valuable "invisible" transactions were getting lost by the application. Where some level of diagnosis is involved, it helps if the person fixing the problem can converse with the person who detected it (see also [Ehr94, Ehr98]). What happened with the TTS system was that people reverted to their old habits and used the system after the fact to encode what happened rather than, as intended, to direct their work.

> "While TTS was designed to make job performance more efficient, it has created the opposite effect: discouraging the training of new hands, breaking up the community of practice by eliminating troubleshooting conversations, and extending the time spent on a single job by segmenting coherent troubleshooting efforts into unconnected ticket-based tasks." [page 41]

1.2.2 Methodologies

Having emphasized the importance of studying groups at work, the question arises, what is the best method to use to study work practices and group behavior. The work is largely tacit, invisible and unarticulated, distributed across time and place and hence hard to observe, and involves multiple people.

This section outlines three methodologies commonly used to understand work practices and group behavior: Ethnography, Participatory Design, Action Research. Even a deep understanding of work practices does not automatically result in requirements for applications to support those practices. There is an additional and explicit step required to translate the results of empirical research into ideas for new applications. This step is explored in a discussion of "Applied Ethnography" which describes how empirical results might be used to a) identify new product opportunities; b) evaluate existing technologies; c) provide input to design specifications.

It should be noted that groupware applications are frequently developed in direct consultation with the user. In cases where the application is built by a consulting group or an internal IS department, the close relationship with the customer often means that the customer's problem is known ahead of time. In these cases, the challenge for the application developers and designers is to elicit requirements from the customer that get at the root of what the problem *really* is, rather than what the customer says it is. Methods such as focus groups (e.g. [Hol93]), brainstorming and scenarios (e.g. [Car95]) may be employed, along with an iterative development process using rapid prototyping techniques to elicit these requirements.

When a particular customer has not been identified ahead of time, as is the case with research projects and "shrink-wrapped" commercial applications that are not designed for a particular customer, then methods derived from research may be more appropriate. These methods can, of course, also be used in consulting and other settings. Three methods are described here which have been used by researchers and practitioners of CSCW and groupware.

1.2.2.1 Ethnography

Perhaps the most common methodology used in CSCW and groupware derives from ethnography as it was developed in anthropology, building on the recognition that workplaces are types of specialized cultures (see especially [Blo93a, Jor96]).

> "As practiced by most ethnographers, developing an understanding of human behavior requires a period of field work where the ethnographer becomes immersed in the activities of the people studied. Typically, field work involves some combination of observation, informal interviewing, and participation in the ongoing events of the community. Through extensive contact with the people studied, ethnographers develop a descriptive understanding of the observed behaviors." [Blo93a, page 124]

By focusing on observation and the study of people at work in their normal work setting, ethnographic methods can uncover and articulate the tacit, invisible work practices.

> " The ethnographic method, through participant observation, pays attention to how actors construct their understandings with others through a set of shared practices." [Ban96, page 14]

Asking people directly about their work won't reveal what is going on because even those who spend time reflecting on their own work — and they are in a minority — are too engaged in the work to be able to step back and explain the minutiae of what they do. However, some researchers have developed video-based observational and analysis methods in part to elicit post-hoc reflections from the users (e.g. [Jor95]).

Yet, as we saw in the examples from the print industry and the phone company, developers must pay attention to the minutiae of work practices in order to design and build an application that will be accepted, adopted and adapted by users to their work.

> " The purpose of ethnography is to carry out the detailed observations of activities within their natural setting. The aim is to provide details of the routine practices through which work is accomplished, identifying the contingencies that can arise, how they are overcome and accommodated, how divisions of labor are actually achieved, how technology can hinder as well as support activities, and so on." [Bly97b, page 40]

1.2.2.2 Participatory Design

A complementary method is one in which the users and other stakeholders of the software are involved in the design from a very early stage and throughout the design and development process. Often referred to as Participatory Design ([Sch93, Mul93]), this approach emerged from work by labor unions and others in Scandinavia acting as advocates for workers and for workplace democracy (see [Gre91] for review of work). A Participatory Design approach privileges the users in design decisions.

> "The focus of participatory design (PD) is not only the improvement of the information system, but also the empowerment of workers so they can co-determine the development of the information system and of their workplace." [Cle93, page 29]

1.2.2.3 Action Research

There are a number of methods from social psychology and related social sciences which seek to understand groups and group behavior (see especially, [McG84]). Of these, Action Research is distinguished for its emphasis not only on groups — especially teams — but for its desire to apply the results of the research to interventions that are designed to improve team performance (e.g. [Arg78, Arg82]).

A premise of Action Research is that organizations learn — and hence improve — by reflecting and reexamining the premises under which they are operating:

> " The ultimate purpose of action science is to produce valid generalizations about how individuals and social systems, whether groups, intergroups, or organizations can (through their social agents) design and implement their intentions in everyday life. The generalization should lead the users to understand reality and to construct and take action within it." [Arg82, page 469]

Action Research resembles ethnographic methods only in so far as both rely on observation and qualitative rather than quantitative descriptions. They diverge, in how the empirical results are used. Ethnographers prefer to take a neutral position on imposing any value judgment on what they observe; action researchers have it as a goal to change, for the better, the team's behavior and performance.

Historically, practitioners of Action Research have eschewed technology, preferring direct personal interventions to achieve organizational change. However, there is no *a priori* reason why the understanding and insights from Action Research methods could not be applied to the design of technologies which reflect organizational practices. Indeed it is not uncommon for people engaged in adapting, advising on or building groupware applications to describe their work as Action Research. Action Research and Participatory Design differ in whether the application should support or challenge the current status quo. Participatory Design privileges current users and current practices and seeks, by and large, to design applications to support and maintain those practices. Action Research enters into a study of a team with a belief in the value of bringing in interventions in order to assess patterns of activity.

Some potential points of synergy between these methodologies can be found in Snyder [Sny98] who combines theories of organizational learning with ethnographic research to yield insights and potential new interventions to communities of practice. Orlikowski and Hofman [Orl97] discuss strategies for introducing technology into organizations.

1.2.3 Applied Ethnography

There has, unfortunately, often been a disconnect between those who study work practices and those who develop groupware systems. On the one hand, research ethnographers have generally shied away from translating their empirical results into specific design recommendations lest their descriptive findings be misconstrued as being too prescriptive. On the other hand, developers have not delved into the details of the findings to extricate the implications for their particular application.

The gap between empirical results and application is due in part to the difficulty in translating from the specificity of the work environment being studied to the general and often unknown constraints and requirements of the application environment. Plowman et al. [Plo95] argue that the lack of translation from ethnographic studies to application design arises in part because the people who do the workplace research by and large do not also develop the applications. This, they argue, means that someone has to translate the results from the empirical to the technical domain — a problem compounded by the inherently descriptive nature of ethnographic findings. Some (e.g. [Rog97]) have taken the translation task to heart and arrived at various techniques such as creating a set of guiding questions, use of video clips and photos, highlighting breakdowns in the current process to convey the results of ethnographic studies to the development team. Others (e.g. [Ben92]) acknowledge the philosophical differences between ethnographers and system developers. The ethnographers are able to influence the design by working closely with the development team and showing a willingness to be flexible.

When properly applied, insights and results of ethnographic studies can: a) identify new product opportunities, b) evaluate the use of existing technologies, and c) provide input to design specifications [Bly97b]. Examples of this “applied ethnography” can be found in the proceedings of conferences such as the biannual CSCW and European ECSCW conferences.

1.2.3.1 Identify New Product Opportunities

Ideas for new, innovative applications won’t come exclusively from ethnographic studies (see [Bro91] for extensive discussion on the source of ideas for innovative applications). However, ethnographic studies, because they focus so closely on the actual work being done, are well suited to generate insights into potential new software applications. When the result of an ethnographic study is used to identify new product opportunities, the group being studied are often different than the group targeted by the application. For instance:

1. In a study of a customer support organization, Ehrlich and Cash [Ehr94] observed that support analysts routinely shared references to previous cases and to printed or on-line material. Most of these references were shared as part of the normal dialog about a case. These and other observations of how people share recommendations led to the development of a collaborative filtering system for semi-automated personalized recommendations to on-line documents [Mal95]. Using the application, a person who finds a document that he/she believes will be of interest to a colleague can forward an e-mail link to the document along with a personal recommendation.
2. Nardi et al [Nar98b] report on the design of Apple Data Detectors, which are intelligent agents that analyze structured information and perform the appropriate operations. For instance, a user finding a meeting announcement could instruct the Detector to automatically add the announcement to a calendar. The development of this product emerged in part from observations made by Barreau and Nardi [Bar95] who, in a study of how people organize their desktops, found that users often complained of not being able to act on structured information found in common documents. The development of the product was also informed by a detailed ethnographic study of reference librarians [Nar96] who acted as agents on behalf of users looking for information. The results of that study translated into a design goal of having the software agent be unobtrusive and able to infer user needs.

1.2.3.2 Evaluation of Existing Technology

A variety of field methods can be used to evaluate how well existing applications are being incorporated into the work practice. If the application is not well suited to the setting, as we saw in the earlier example of the print shop, the failure will be readily apparent. Conversely, when technology has been successfully incorporated into the work practices, the application designer can consider extending the application or applying the application to other settings, but not without considering the consequences of transfer.

> In a recent ethnographic study of nurse reviewers who worked on disability and workers' compensation cases, Ehrlich and Cash [Ehr97] found that an administrative application with an embedded decision support component was well integrated into the nurse reviewers' work practice. They used the application to estimate the length of time that an injured worker should be away from work. The successful use by the nurse reviewers led to speculation that the application could be successfully deployed by physicians and physician assistants who were treating the patients. Although the physicians and nurse reviewers are linked in a type of extended enterprise, they nevertheless acted independently. The decisions made by the nurse reviewers were informed by their professional judgment and by their evolved work practices. Ehrlich and Cash argue that reallocating tasks to another part of an enterprise requires re-analysis of the overall context.

Lab rather than field methods can also be a useful way of evaluating applications, especially when the goal is to identify particular effects. For instance:

> Mark et al [Mar95] studied the effect of a hypermedia system, Dolphin, on the form, content and linkage amongst ideas created in a face-to-face meeting. Participants in the meeting were engaged in problem-solving exercises. Mark et al found that the people who used the application would group their ideas into networks rather than hierarchical structures and provide more

elaboration for their ideas. Those people not using the application generated less elaborated ideas. Thus, the technology had a qualitative effect on problem-solving behavior of the people in the meeting.

1.2.3.3 Input to Design Specifications

New applications often follow a process in which an initial concept — generated from marketing requirements or from the vision of a small group of people — is modified and elaborated into a richer functional specification. Ethnography and other field methods, *when used to study the intended user population,* can provide input to these design requirements. The examples below provide a diverse set of cases where ethnography had a direct influence on design directions.

1. In a study of air traffic controllers Bentley et al [Ben92] and Hughes et al [Hug92] found that seemingly routine work was coordinated through a sophisticated use of flight "strips". These pieces of paper carry static information about expected and current flights along with instructions to the aircraft being controlled by the center. However, the controllers had evolved a practice of manually organizing the strips on a visible flight progress board. The physical ordering of the strips provided implicit, tacit cues to help the controllers dynamically coordinate and allocate their work. Based on these observations the ethnographers could direct the design of an automated system for controllers away from an automatic assignment of strips and toward maintaining elements of the manual method. The study highlights one of the critical roles of ethnography which is to articulate and demonstrate to developers that "manual intervention and manipulation of information may be essential implicit methods of communication and cooperation".
2. Blomberg et al [Blo93b] offer brief descriptions of several studies done under the rubric of work-oriented design in which attention to the details of work is used to help guide the design of new applications. For instance, they studied the use of color and highlighting to distinguish the text annotations of different people on order forms as part of the coordination of activity across organizational boundaries. "By providing developers with visual representations of how the work of processing orders is supported by annotations, and by viewing videotapes of the people engaged in the work, we are exploring with developers and work practitioners how computationally active marks on paper might support this work."
3. In a different arena, Kukla et al [Kuk92] worked with Monsanto and Fisher Controls Inc. "to investigate and apply modern information technology" to Monsanto's integrated nylon facility in Pensacola, Florida. The goal was to optimize the use of raw materials and energy through the facility. An ethnographic approach, comprising interviews, observations and detailed information on one sector of the plant was used to "construct models of events, conversations and processes within that area of the plant. These models were to be used as a basis for developing software tools for use within the plant." At the beginning, work was characterized as routine and repetitious. But based on ethnography, a number of less visible aspects of work were uncovered. These included: the ability of people working in the plant to do ad-hoc juxtaposition of data screens (such as compare live process data to histograms or maintenance records); and the importance of manual, not automated, collection of data (e.g. by sensors of machines) by people to get the richness of the environment (e.g. noises, smells, comments by people working near and with machines). These and

other findings were translated into the design and development of specialized software for the process industry, linking realtime process data with desktop applications. The product, DEC@aGlance, was marketed in 1992.

4. Bly [Bly88] and Tang [Tan91] studied teams of people, working at distance, who need to work together to create drawings, designs and engage in general brainstorming. Ethnographic studies of people working together as well as people working apart led to many observations about such things as the use of gestures and marks to illustrate ideas, how control is passed from one person to another and how drawing and talking are combined. These observations led to the development of a prototype (e.g. [Min91]) for use in a research setting. That prototype subsequently influenced the development of products for synchronous shared collaboration from Sun Microsystems (e.g. ShowMe) and Xerox (e.g. LiveBoard).
5. Blythin et al [Bly97b] describe an ethnographic study at a bank of a service center which processed routine administrative details of accounts. Based on studying this group over time, the researchers uncovered limitations and problems imposed by the physical and organizational setting which impeded the effective and smooth flow of work. For instance, there was a physical separation between some supervisors and their teams which reduced the opportunity for informal awareness of the progress of work. Based on these and other findings, the researchers made recommendations for changes in management practices and processes, to provide better review and oversight and changes in (physical and functional) office assignments. These changes helped increase the supervisor's awareness of the group's work.
6. Katzenberg and Piela [Kat93b] used work language analysis combined with ethnography to study and verify "work language" in the form of names that different groups of people use to label computer systems, such as "compile, instantiate, create, build". The results of the ethnographic study were a set of guidelines for the continued development of a technology used in engineering and economic forecasting to analyze design alternatives.

Being able to use the results of ethnographic studies means that researchers and practitioners must be open to question their initial assumptions in the face of user data. For instance, in a field study of a distributed team, Bellotti and Bly [Bel96] observed that members of the team were rarely at their desks but instead could be found in the hallways or working in labs. Although Bellotti and Bly went to the site to gather requirements for a computer-based application to support distance collaboration, it was apparent that such an application would not be used if it was only available from the computer. Instead, the researchers were able to recommend alternative solutions based on mobile computing devices.

1.2.3.4 Working Together

These examples also draw our attention to the most important part of the design process, which is the collaboration between the ethnographers and the application designers. The results of ethnographic studies do not stand on their own but must be interpreted by both the ethnographer and the application designer. Just handing a report of the ethnography to the designers is not sufficient. The two groups must work together as a team when the data are being collected and analyzed. It is also crucial that there be reciprocal appreciation and respect of others' viewpoints. The need to overcome different world views, cultures and perspectives is a recurring theme in these studies.

An especially good example of a successful collaboration comes from a study by Linde, Pea and others at IRL (Institute for Research on Learning) for the design of an interactive multimedia communication device [Gog96, All93, deV91]. In a close examination of actual work sites, a multidisciplinary team of researchers representing application developers and ethnographers investigated the learning and work practices that emerged as new communication and computational technologies were integrated into ongoing activities. The design and development process was highly iterative. Outcomes of the studies would get translated into mock-ups which would be tested with users, modified and re-tested.

In one phase of this study, the ethnographers were videotaping a small group of graphic designers at work [Lin91]. The graphic designers organized their ideas using folders. But what the ethnographers observed was that during group meetings the folders were placed on the table in a particular way. The placement — close to the owner or toward the middle — was a form of non-verbal communication used to signal permission for others to talk. The ethnographers were able to point out this observation to the application designers who would not otherwise have been aware of the importance of the folders and their placement. Based on the ethnographers' analysis and their own observation, the application designers realized that the design of the software would need to include not just the ability to share folders but those folders would need to be marked as read only, private or open. This is a small example that was repeated many times in the course of the collaboration between the ethnographers and the application designers.

1.3 DESIGN

Part of the appeal of groupware lies in the promise of being able to eradicate barriers of time and place. Using technology, colleagues should be able to collaborate on projects whether their offices are next to each other or in separate countries, whether they work at the same time or different times of the day. Applications that help bridge barriers of time and place include videoconferencing, shared screens, media spaces, electronic mail, shared files/databases, shared authoring, and group calendaring systems. However, subtle social protocols influence the willingness of participants to communicate with others, the candor of their communication, the richness of information they are willing to impart, and the degree of their engagement in the process. If technology is going to mediate communication especially for people who lack opportunities for face-to-face meetings, it must support rather than ignore these protocols. Getting inside this notion of group *work*, a few themes emerge:

- communication is generally ad hoc, *informal* and unplanned
- there is a need to be *aware* of others for communication and in coordinating work
- issues of sharing often hinge on subtle notions of *anonymity*.

1.3.1 Informal Communication

Research on synchronous, informal communication emphasizes its importance and prevalence in most workplace settings [Kra90, Whi94]. These studies suggest that formal communication is used to coordinate routine tasks whereas brief, informal communication such as spontaneous hallway conversations can help to establish trust, promote social relationships and provide background information about the work environment. Moreover, these spontaneous conversations are more likely to occur amongst people who are physically located close to each other; as many as 91% of all conversations recorded in a particular study occurred among people on the same floor [Kra90].

One type of video-based system, known as MediaSpaces, has been developed to provide visual access and opportunity for conversation to people who are not located in the same place [Fis90, Man91, Dou92, Bly93, Fis93] (see also Chapter 3 in this book [Mac99]). These systems provide continuous visual access between sites through large video screens, often placed in public areas such as hallways or informal meeting places. However, despite their careful design, these systems cannot substitute for unmediated face-to-face conversations.

1.3.2 Awareness

Awareness, of the location and activity of other people, is a critical mechanism for regulating and coordinating our behavior with others. We use cues in the physical environment such as a colleagues' open door, the placement of a work-related document [Hug92] or the level of participation in an on-line discussion, to make decisions about whether to initiate a conversation, begin the next sequence of work or anticipate a meeting. The same social protocols still operate when the work is mediated through computer technology. Groupware applications designed to support coordinated work need to find new ways to represent what were physical cues, so that even when online, people can be aware of the activity of their colleagues. Awareness of others usually takes place when there is on-going or anticipated, direct, synchronous communication between people. But there is also a need to be aware of a general level of involvement and participation of a group over time. Both synchronous and asynchronous awareness are explored below.

1.3.2.1 Synchronous Awareness

The Montage desktop videoconferencing system (e.g. [Tan94a, Tan94b]) supports the kind of momentary, reciprocal glances that occur when one person peeks into another's office to see if that person can be interrupted. In Montage the person initiating the conversation selects the name of the person to be contacted, which causes the recipient to receive an auditory signal that a call is about to commence, followed by a gradual fade-in small video image of the caller. Either person can acknowledge the glance by pressing a button to open an audio channel followed by a 2-way audio-video connection. This mediated interruption can get translated into a more extended interaction supported by the full desktop videoconferencing system by pressing the Visit button. If the caller sees from the glance that the other person is not available, the caller can browse the person's calendar, send a short note or send an e-mail message. As in office-based social conventions, Montage users can set their system to display different levels of interruptibility. These range from "locked" which means no interruptions, to "out of the office" and "other", which lets the caller leave a message, to "do not disturb" which still lets the caller glance in to negotiate an interruption, to "available".

There are numerous other studies of awareness including those on awareness as a mechanism to support coordination (e.g. [Dou92]) and social awareness (e.g. [Tol96]). Products to support awareness include "buddy lists" (e.g. [Mic97]) which signal when someone from the list is on line and hence potentially available for an online "chat".

1.3.2.2 Asynchronous Awareness

We also develop awareness of the general work patterns of our colleagues based on cues left in public or semi-public places. If I want to schedule a meeting with my manager, I might ask his assistant about his availability or I might check online sources such as group calendars, e-mail or online discussions to pick up cues about up-coming meetings, trips and so forth. In the case of group calendars, availability of people's schedules is both a strength for scheduling meetings, and a source of noncompliance for those people who feel exposed (e.g. [Gru95]).

1.3.3 Anonymity

In face-to-face communication, whether direct or mediated by computer technology, the contributors to the conversation are known and visible or audible. However, when there is no visual component to the communication, as in the case of electronic mail and asynchronous communication in general, the technology can hide the identity of the sender or the recipient of the message. This feature has interesting and often unexpected affects on the communication. For instance, people are much more likely to engage in antisocial behavior, such as "flaming" in electronic mail, where the sender's identity may be hidden by an obscure e-mail address and where the usual social protocols to discourage such behavior are absent. Sproull and Kiesler [Spr93] argue that social norms are not well established in computer mediated communication in part because social cues, which are normally present in the physical environment, are absent. For example, the physical appearance and dress code of someone we are about to meet clues us in to the level of formality expected.

On the other hand anonymity can have positive effects. Several researchers have observed that anonymity can reduce effects of power, status and attractiveness (e.g. [Zub88, Tur95]) enabling people who might not have participated in social engagements due to lower status or power to do so when they are anonymous. Similarly, Sproull and Kiesler [Spr93] report that junior members of an organization are much more likely to communicate with senior managers or executives using electronic mail than in a face-to-face meeting. As a classic cartoon in the New Yorker put it: "On the internet no-one knows you are a dog".

Similar effects of anonymity on people's social behavior in computer mediated settings have been observed with computer supported meetings (e.g. [Nun91]). Computer supported meetings typically take place in rooms which have been specially configured with computers embedded into desks or tables [Man89]. The software running on these computers support activities such as brainstorming by letting people freely enter their ideas. The software can then display the individual ideas or some aggregated version of those ideas on the individual terminals or on a large screen visible to all participants. These meetings are generally controlled by a trained facilitator who provides some degree of software support and training as well as handling the dynamics of the meeting itself.

The software portion of these systems can be easily configured to control when ideas get shared amongst the group and whether the ideas are marked with the name of the person who contributed them. Nunamaker and his colleagues have observed that in these kinds of settings, anonymity reduces the pressure to conform and reduces apprehension related to evaluation by one's peers. This, in turn, may encourage a more open, honest and freewheeling discussion. On the other hand, anonymity can increase "free riding". If nobody's comments are attributed, there is no way of checking that everyone in the meeting is actually participating.

1.3.4 Application of Design Themes

Themes such as informal communication, awareness and anonymity rightly belong to the category we have described as *group work* in that these features are not readily apparent from a task focused view. Yet, the presence of these features in an application can materially affect *how* the application is used and *whether* it is used. The inclusion of these and other themes into the design of the application depends in part on the type of application. Using the division of applications laid out in the introduction, those that focus primarily on communication such as e-mail, videoconferencing, media spaces and chat, may be designed around themes of informal communication and awareness. Applications designed to support meetings, on the other hand, need to pay attention to whether issues such as anonymity are needed in the design. In the case of applications that support information sharing, one of the main barriers to acceptance is the readiness of the organization in which the application is to be deployed. This topic will be addressed in Section 1.4.

1.3.5 Customization

We may think of groupware applications such as those that support communication, meetings and information sharing as general-purpose applications ready to be used by a wide range of users for a variety of purposes. This is true of individual "productivity applications" which are designed to be used out of the box with little or no customization. However, groupware applications are rarely ready to use "out of the box" but require some degree of customization. How much work is required depends in part on the type of application, whether it was developed for a particular customer and how the application is architected.

It is fair to say that while customization is not exclusive to groupware applications, in practice there are sufficient differences in work, culture and context from one customer to another that most groupware applications require some degree of customization. This is an important topic which has received little public discussion and hence is only covered briefly here and based largely on personal observations.

1. *Content-based customization.* This is a case where the application is merely a shell and doesn't really become useful until someone begins putting content in. Prime examples are discussion databases, news groups, e-mail etc. Examples from outside the realm of communication software include applications for distance learning where the instructor needs to add course material before the software is useful for the students. In all these examples, the "customization" is done by one or more end-users by supplying content. No specialized technical skill is required.
2. *Setting external parameters.* This is also end-user customization but is more intentional. Examples include TeamRoom (see description in Section 1.5) which is an application to support distributed team work through shared documents, etc. TeamRoom defines attributes such as document category and communication type whose values get set by the team. In this way, the team gets to customize the application to suit the way they intend to use it. A research group, for instance, may want to define categories for documents to represent different research projects while a product group may want to define documents in terms of product families. This is still end-user customization but this time it may involve an outside facilitator to guide the thinking of the group around the goals of the project, the group norms and expectations for the level of participation.
3. *Setting internal parameters.* This is where some degree of system administration or macro

level programming comes in. For instance, in an internally developed system to support on-line reviewing of papers submitted to conferences, the level of customization from one conference to the next ranged from inputting a new set of reviewer names to rewriting parts of the interface to recoding the rules that govern who sees which papers and at what stage of the reviewing process. Many of these changes reflected differences in the reviewing process from one conference to another.

4. *Totally customized solutions.* Applications that match the particulars of an organization's work practices, processes and culture often require that a customized application be built either by someone within the organization or by engaging external consultants.

Some issues of customization are addressed by the available development tools and environments which may provide the pieces out of which the customization is done. One example is the use of templates out of which new solutions can be fashioned. In an article on the use of templates for building business applications, Hofman and Rockart [Hof94] provide an example of a template developed by John Wiley, the publisher, to support internal business processes that allowed for customization by each business unit. This approach allowed them to share best practices, both applications as well as knowledge, aggregate data centrally, and "tailor the business process and system to local needs."

1.4 DEPLOYMENT AND ADOPTION OF THE APPLICATION

In addition to the challenges of building a good groupware application, there are significant challenges facing a developer who is trying to get the application adopted by an organization. Unlike single-user applications which can often be purchased by an individual, groupware applications are, be definition, for groups of people. Hence, enough copies of the application need to be purchased and installed at about the same time for the application to be available to more than one person. Moreover, groupware applications often require a sophisticated technology infrastructure which may in turn require skilled technical staff for the system's administration. In addition to the financial cost of purchasing, installing and maintaining a groupware application, there are also organizational implications of deploying the application. These implications vary with the type of application. For instance, deploying a video-conferencing system may require very little preparatory work within the group, assuming the application itself has been well designed and the infrastructure is in place. On the other hand, an application that depends on a high level of information sharing presumes an organization in which information sharing is already well established and rewarded. This section explores a few of the organizational and cultural barriers to successful deployment and adoption.

1.4.1 Organizational Preparedness

Technology can be introduced into organizations through a mandate imposed by senior management. This method has the advantage of ensuring continued financial and technical support through deployment and in helping disperse the technology through the organization to reach a critical mass of users (e.g. [Mar90]). However, this method of adoption can leave end-users feeling that a decision was forced on them. For instance, Orlikowski [Orl92] reports on the adoption by a large consulting company of groupware to support information sharing. Not only was the culture of the group one in which information sharing was not rewarded, but the technology was introduced to the group without sufficient explanation or training, thus giving these end-users no real understanding or motivation for wanting to expend the extra effort to learn and use the technology. As a result, the technology was poorly adopted and only gained in acceptance over time and with considerable investment and push on the part of senior management, who retained strong conviction in the benefits of the technology. An interesting side note is that the same technology was adopted more or less spontaneously by other groups in the same company where there had been no mandate by management.

Technology can also be introduced into an organization by someone within the organization seeing the potential of the technology. This method has the advantage of getting end-users involved early on. But it has the disadvantage of needing to get buy-in from senior management for continued support.

In a recent study, Grudin and Palen [Gru95] examined the adoption of shared calendar applications in two large organizations. They observed widely dispersed use of the application despite no clear mandate from senior management. They argued that

> “ The features ... may attract a critical mass of users, after which technology-abetted social pressure by peers and others extends use” [page 277]

In at least one organization, the adoption was slow when the application was first introduced. Over time, what developed was a more consistent infrastructure that gave wider access to the application, improved functionality and ease of use and peer pressure. Once a critical mass of people begin using the application, there is strong peer pressure to bring others in line. Calendaring is an example of a groupware application that requires near universal adoption to be successful. Once someone uses the tool to schedule meetings with some colleagues, they will want to be able to use the tool to schedule meetings with other colleagues, and will apply pressure to those colleagues not yet using the application to begin to do so.

1.4.2 Incentives and Motivation

In a work setting, most people are persuaded to adopt a new technology by arguments that make it clear how that technology will improve their work. Such arguments may focus on the technology as enabling the person to do something that was previously very difficult or cumbersome to do. For instance, on-line discussions make it easier to share information with a number of colleagues simultaneously than it is to attend face-to-face meetings. This is especially true if colleagues are not all located in the same place or if it is hard to schedule a time when everyone can attend a meeting. New technology in general, especially groupware, will get adopted more easily if it fills a need rather than simply replaces an existing well understood, working process. For instance, videoconferencing technologies got a major push during a recent oil crisis when it got harder and more expensive for people to travel. The need to communicate and collaborate with colleagues didn’t go away, but reaching those people got harder.

Convincing end-users of the benefit of any new technology is challenging — especially so for groupware applications for which there may be no visible examples of use. For example, several years ago, Wang introduced a multimedia communication system which bundled together image capture, voice recording, electronic mail, pen annotation and high-resolution graphics [Fra91]. The system was intended to be used to annotate and route documents through an organization. However, when the system was introduced into client sites it failed, in part because neither end-users nor management were ready to risk new unproven methods of working despite being told of the benefits of the system.

1.4.3 Critical Mass

Groupware applications are principally designed to benefit and reward the group rather than the individual. But most people are not altruistic. They want some personal benefit from using the application. Getting enough early adopters to use a new system is especially challenging for applications which rely on a large number of people to be effective. Many collaborative filtering systems recommend selections (e.g. of video, music, Net News) based on a statistical aggregate of individual ratings (e.g. [Hil95, Sha95, Res94, Gol92]). When the database has been seeded with enough ratings, users can query it to learn which selections are recommended. But where is the incentive for the early adopters to add their ratings? Resnick et al [Res94] argue that some people are altruistic, while others may be motivated by external rewards such as money to be an early adopter. Reaching a critical mass also proved to be the key factor in a study which systematically compared adoption rates of two similar video telephone systems [Kra94].

The potential asymmetry between those who contribute and those who get the benefit has been underscored by Grudin [Gru90]. He points out that with group-enabled systems such as group calendars and shared project management applications, the beneficiary is often the person scheduling meetings or managing the project, rather than the people contributing the information about their schedules or time.

1.5 CASE STUDY: TEAMROOM

To illustrate many of the points in this chapter, this section presents a brief case study of the design, development and deployment of a particular groupware application. The application, called TeamRoom, was initially developed for use by internal task forces at Lotus [Col96] to support discussion and coordination. It was then made available to outside customers as part of a consulting engagement, and is now sold as one of the family of Lotus/Domino applications.

1.5.1 Teams

Before building an application to support discussion and coordinating by members of a team, it is important to understand how teams work. At a very general level, teams of people work collectively and collaboratively to:

- make decisions
- share information
- coordinate actions.

Teams will be high performing, to the degree that they engage in these activities in a deliberate and persuasive manner to produce something of value to the organization such as a tangible product, a process or a service. There is a large amount written about teams and team performance, which will not be addressed here. As Katzenbach and Smith [Kat93a] expressed recently:

> “ Real teams are deeply committed to their purpose, goals, and approach. High-performance team members are also very committed to one another.” [page 9]

However, there is increased pressure on teams to deliver more value in less time with few resources. Moreover, teams are often ad hoc (e.g. [Fin90]); formed “just in time” to solve a particular problem and then disbanded. And, members of the team may be dispersed throughout the organization as well as separately located due to travel or residence. Team members often come from different cultures in terms of professional training, background, tenure with the company, or nationality.

Technology cannot eliminate these barriers on time and place. However, in conjunction with judicious training, technology can help make the team, once formed, more effective and efficient. People working together need: shared context, shared language and shared objectives. They also need a “workplace” in which the majority of work will get done and where shared discussions as well as private conversations can take place. It is not the place of technology to help create the team, but rather, to support the team once it has formed. (An important and debatable question is whether the application is only as effective as the team or whether a well designed application can overcome limitations in the group dynamics.)

1.5.2 Requirements

TeamRoom was developed in response to a request from one of the senior vice-presidents at Lotus for an application that could support the work of internal task forces. Although all the people on a task force worked for the same company, they came from different parts of the organization and also traveled frequently. This meant that face-to-face meetings occurred only occasionally, necessitating the need to have a place to post documents, have discussions, plan meetings and share ideas. The development team was composed of people from development, design, internal information systems, Human Resources/Organizational Development, as well as the main customer and a representative group of users. Detailed user requirements and functional specifications were arrived at through discussions within this team and by exposing early prototypes to the users.

1.5.3 Design Features

Instead of having team members go to one application to retrieve shared documents, another for group e-mail and a third application for coordination, TeamRoom provides a single “place” for these activities by integrating all three functions in a single application.

TeamRoom builds on the Lotus Notes model of a threaded discussion database where messages can be posted and read by anyone who has access to the database. Documents can be entered to start a topic or as a response to an existing topic. In TeamRoom documents are keyed by communication type as well as by category. Common communication types include: discussion, action request, meeting announcement and reference document, and are used to signal the intent and type of communication. For instance, a discussion document signals that the author wants other people to respond, whereas a reference document may require little or no further discussion. These different document types reinforce and simplify communication. Documents can be viewed by topic, communication type and author as well as by date, so that a user can quickly see which documents have been added recently.

One of the distinguishing characteristics of TeamRoom is its embodiment in the software of the communication norms of a group. The translation of norms to features is handled through the *process* of having the group define its goals, mission, categories and communication types, and the *mechanism* of having these instantiated in the software by completing information about the team on a “Mission Page”. This information includes: categories, communication types, participants and events. Once entered, this information becomes available to the team in the form of keyword lists, which a user selects when composing a document, and visible categories, which users see when viewing documents. Documents can also be automatically archived, which reduces the problem of information overload. The Mission Page is the place where the team records details of their processes and norms; such things as when to post documents to particular people rather than have the document default to be seen by everyone; and the meaning and intent of the different communication types and categories. Teams who have spent time filling out the mission page have found the information there to be invaluable as a source of group memory and an excellent vehicle for new members to get oriented. TeamRoom, especially the Mission Page, becomes a *work* space for *group* memory.

1.5.3.1 Relation to Design Themes

Section 1.3 identified three main group work themes: informal communication, awareness of others, and anonymity. Many of these themes can be seen in TeamRoom.

TeamRoom differs from both e-mail and discussion databases in that it is a place where all work — not just discussions — gets posted. Making each person’s work visible to the rest of the team contributes to an awareness of other people’s level of contribution and the current status of their work. By looking across the different categories, it is easy to see which documents are still in process and which are completed. An index view also shows the number of documents per author or communication type. An author view shows documents by type, which provides a view into whether a person is mostly commenting on other people’s work or contributing their own.

TeamRoom supports informal communication by supporting loosely structured discussions. But it doesn’t really support ad hoc informal communication since it is designed for teams that are not co-located (see Section 1.5.4 below).

TeamRoom does not let people participate anonymously, but it does support private, as well as public discussions, and personal as well as shared workspaces. When a document is posted in TeamRoom, the author is required to specify only the communication type and the document type. The author can optionally mark a document as private to be seen only by people who the author lists in the To: field. The To: field is also used to designate people who need to pay particular attention to a document, even if that document can be seen by the rest of the team. TeamRoom constructs personal workspaces for each member of the team, based on documents for which the person is listed in the To: or cc: field or documents that the member has authored. In this way, TeamRoom supports personalized as well as shared views.

1.5.3.2 Relation to Deployment

When it comes time to deploy the groupware application, organizational preparedness, incentive and motivation, and critical mass are some of the factors which influence the ultimate adoption of the application.

TeamRoom addresses issues of organizational preparedness by accompanying the introduction of the technology with a facilitated meeting during which the members of the team go through the exercise of deciding on their mission, communication and work styles as a team. In this way, all members of the team can participate in setting the goals. The mission and the technology can be seen as being in service of the core work of the team.

One of the ways in which TeamRoom addresses the problem of critical mass — that is making sure that there is enough activity to promote more activity — is by having a facilitator as one of the designated roles in the TeamRoom. This person, who is also a team member, monitors traffic in the TeamRoom and encourages participation if the discussion and postings are becoming reduced.

1.5.4 Lessons Learned

TeamRoom has been deployed in a wide range of companies and settings. Based on informal feedback there are several themes that emerge that are critical success factors: 1) strong leadership; 2) a distributed team who need TeamRoom to overcome barriers of time and place and for whom face-to-face meetings are often scheduled, rather than ad hoc; 3) a well defined team. Below are examples where these factors were *absent*.

Strong leadership. Strong leadership is needed, especially early on, to get people to submit postings. For some teams, collaboration and sharing was a new way of working and if the team leader didn't demonstrate and lead by example, the team generally did not take to it. This was lacking at one company where it seems the team leader set the tone/behavior for the group. In a few cases, the workers still took to the tool, seeing its value and needing the communication that it provided.

Geographically distributed team. TeamRoom is a good alternative to voice or videoconferences for teams whose members are far apart. Especially for complex projects, TeamRoom becomes an information repository to facilitate analysis. However, there can sometimes be delays in replicating TeamRoom to the different sites for these distributed teams. On the other hand, when team members were co-located, TeamRoom was just another thing to have to worry about and it wasn't used.

Well defined team. A well defined team has a common mission and a shared context, language and objectives. Team membership is limited and definable. Well defined teams are not about longevity; a team could be just forming or be together for a long time. One team that used TeamRoom that wasn't really a team, but just a department, failed in their use because there was no real team mission, team norms, or team deliverables. In this case, TeamRoom served as a place to communicate meeting agendas and some marketing announcements. TeamRoom is a mirror on the team. If the team is chaotic, then so it will appear in TeamRoom and people's experience with the tool will be frustrating. A well-organized team takes a lot of work.

1.6 SUMMARY

This chapter takes the position that whereas single-user applications are about *tasks*, groupware applications are about *work*. Tasks are generally explicit, observable, concrete. Work is generally tacit, invisible and amorphous. Work is about people, habits and culture.

Generating a product concept and design specifications for a groupware application demands a methodology that can capture these invisible work practices. The methods, derived from social and management sciences, that are most commonly used in groupware or CSCW

(Computer Supported Cooperative Work) are often descriptive rather than prescriptive, leaving it up to the design team to fashion requirements, functions and architecture themselves. A multidisciplinary team is essential for the design and development of groupware applications.

Getting inside the notion of group "work", a few themes emerge: communication is generally ad hoc, informal and unplanned. There is a need to be aware of others for communication and in coordinating work. Issues of sharing hinge on subtle notions of anonymity, which play out in different ways.

Finally, deploying a groupware application is perhaps the most difficult step in process. First, the application itself will need some level of customization to fit in each customer's work context. Second, groupware applications are rarely ready to go "out of the box" but need to be accompanied by some measure of training in organizational behavior to ensure a fit between the tool and organizational processes. Factors such as motivation, incentives and critical mass are potential show-stoppers when it comes to rolling out the application to the entire group.

The notion of work and translating it into an application is put into perspective by describing a Lotus Notes application that was designed to provide a "place" on line to support discussion and coordination of work amongst members of a distributed team.

Groupware and CSCW are still in their infancy compared with more established practices in the development of single-user applications. Yet, technological developments such as the World Wide Web seem to lead to more need for groupware applications where people spread across the globe, across the country, or just across the street can use technology to coordinate their work and communication with each other.

ACKNOWLEDGEMENTS

There are many people who have contributed to this chapter through discussions, especially those enlightening me on various methodological issues. For their time and patience in talking with me about methodology, I extend my appreciation to: Barbara Katzenberg, Charlotte Linde, Bonnie Nardi, Roy Pea, and Lee Sproull. My colleagues at Lotus Institute, especially Barbara Kivowitz, Linda Carotenuto, and Nicol Rupolo helped me understand many of the nuances and tacit features of TeamRoom. I extend a special thanks to my colleague Debra Cash with whom I have had many engaging and heated conversations and who took the time to read and comment on several drafts of this chapter. Thanks are due to Paul Cole, Sal Mazzotta and especially an anonymous reviewer who read and commented on an earlier version.

REFERENCES

[All93] Allen, C., The reciprocal evolution of technology, work practice and basic research. In D. Schuler and A. Namioka (Eds.) *Participatory Design: Perspectives on System Design*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.

[Arg82] Argyris, C., *Reasoning, Learning and Action: Individual and Organizational*. Jossey-Bass, San Francisco, 1982.

[Arg78] Argyris, C. and Schon, D., *Organizational Learning*. Addison-Wesley, Boston, 1978.

[Ban96] Bannon, L. Ethnography and design. In D. Shapiro, M. Tauber, and R. Traunmuller (Eds.), *The Design of Computer Supported Cooperative Work and Groupware Systems*, pages 13–16. Elsevier Science, Amsterdam, 1996.

[Bar95] Barreau, D. and Nardi, B., Finding and reminding: File organization from the desktop. *SIGCHI Bulletin*, July 1995.

[Bel96] Bellotti, V. and Bly, S., Walking away from the desktop computer. Distributed collaboration and mobility in a product design team. In *Proceedings of the Conference on Computer Supported Work, CSCW '96 (Boston, MA)*, pages 209–219. ACM Press, New York, 1996.

[Ben92] Bentley, R., Hughes, J.A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D. and Sommerville, I., Ethnographically-informed systems designs for air traffic control. In *Proceedings of the Conference on Computer Supported Work, CSCW '92 (Toronto, Canada)*, pages 123–129. ACM Press, New York, 1992.

[Blo93a] Blomberg, J., Giacomi, J., Mosher, A and Swenton-Wall, P., Ethnographic field methods and their relation to design. In D. Schuler and A. Namioka (Eds.), *Participatory Design: Perspectives on System Design*, pages 123–154. Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.

[Blo93b] Blomberg, J., McLaughlin, D. and Suchman, L., Work-oriented design at Xerox. *Communications of the ACM*, 36(4):91, June 1993.

[Bly88] Bly, S., A use of drawing surfaces in different collaborative settings. In *Proceedings of the Conference on Computer Supported Work, CSCW '88*, pages 250–256. ACM Press, New York, 1988.

[Bly93] Bly, S., Harrison, S., and Irwin, S., Media spaces: Bringing people together in a video, audio and computing environment. *Communications of the ACM*, 36(1):28–45, January 1993.

[Bly97a] Bly, S., Field work: Is it product work? *interactions*, pages 25–30, January+February 1997.

[Bly97b] Blythin, S., Rouncefield, M, and Hughes, J.A., Ethnography in the commercial world. *interactions*, pages 38–47, May+June 1997.

[Bow95] Bowers, J., Button, G., and Sharrock, W., Workflow from within and without: Technology and cooperative work on the print industry shopfloor. In H. Marmolin, Y. Sundblad, K. Schmidt (Eds.), *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work, ECSCW '95 (Stockholm, Sweden)*, pages 51–66. Kluwer Academic, Dordrecht, 1995.

[Bro91] Brown, J.S., Research that reinvents the corporation. *Harvard Business Review*, page 330, January-February 1991.

[Bro97] Brown, E.G., Dolberg, S. Boehm, E.W. and Massey, C., Beyond groupware. *Forrester Report: Software Strategies*, 8(4), July 1997.

[Cam95] Cameron, B., DePalma, D.A., O'Herron, R. and Smith, N., Where does groupware fit? *The Forrester Report: Software Strategies*, 6(3), June 1995.

[Car95] Carroll, J.M., Introduction: The scenario perspective on system development. In J.M. Carroll (Ed.), *Scenario-Based Design: Envisioning Work and Technology in System Development*. John Wiley and Sons, New York, 1995.

[Col96] Cole, P. and Johnson, E.C., Lotus development: TeamRoom - A collaborative workspace for cross-functional teams. In P. Lloyd and R. Whitehead (Eds.), *Transforming Organizations Through Groupware: Lotus Notes in Action.*. Springer-Verlag, New York, 1996.

[Cle93] Clement, A and Van den Besselaar, P., A retrospective look at PD projects. *Communications of the ACM*, 36(6):29–37, June 1993.

[deV91] de Vet, J. and Allen, C., *Picasso System Design Rationale*. IRL Technical Report, November 1991.

[Dou92] Dourish, P and Bly, S., Portholes: Supporting awareness in a distributed work group. In *Proceedings of Human Factors in Computing Systems, CHI '92 (Monterey, CA)*, pages 541–547. ACM Press, New York, 1992.

[Ehr94] Ehrlich, K and Cash, D., Turning information into knowledge: Information finding as a collaborative activity. In *Proceedings of the Conference on Digital Libraries (College Station, TX)*, pages 119–125, 1994.

[Ehr97] Ehrlich, K and Cash, D., *Communication and Coordination in Workers Compensation Cases: Implications for Extended Enterprises*. Internal Report, 1997.

[Ehr98] Ehrlich, K. and Cash, D., The invisible world of intermediaries: A cautionary tale. *Computer Supported Cooperative Work: An International Journal.*, In Press.

[Ell99] Ellis, C.A., Workflow technology. In Beaudouin-Lafon, M. (Ed.), *Computer Supported Cooperative Work*, Trends in Software Series 7:29–54. John Wiley & Sons, Chichester, 1999.

[Fin90] Finholt, T., Sproull, L., and Kiesler, S., Communication and performance in ad hoc task groups. In J. Galegher, R.E. Kraut, C. Egido (Eds.), *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, pages 291–326. Lawrence Erlbaum, Hillsdale, NJ, 1990.

[Fis90] Fish, R., Kraut, R.E., and Chalfonte, B., The videowindow system in informal communication. In *Proceedings of the Conference on Computer Supported Work, CSCW '90 (Los Angeles, CA)*, pages 1–12. ACM Press, New York, 1990.

[Fis93] Fish, R., Kraut, R.E., Root, R., and Rice, R., Video as a technology for for informal communication. *Communications of the ACM*, 36(1):48-61, January 1993.

[Fra91] Francik, E., Rudman, S.E., Cooper, D., Levine, S., Putting innovation to work: Adoption strategies for multimedia communication systems. *Communications of the ACM*, 34(12):53-63, December 1991.

[Gog96] Joseph Goguen, J and Charlotte Linde, C., Techniques for requirements elicitation. In R. Thayer and M. Dorman (Eds.), *Software Requirements Engineering, Second Edition*. IEEE Computer Society, 1996.

[Gol92] Goldberg, D., Oki, B., Nichols, D., Terry, D.B., Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61-70, December 1992.

[Gre91] Greenbaum, J and Kyng, M. (Eds.), *Design at Work: Cooperative Design of Computer Systems*. Lawrence Erlbaum, Hillsdale, NJ, 1991.

[Gru90] Grudin, J., Groupware and cooperative work: Problems and prospects. In B. Laurel (Ed.), *The Art of Human Computer Interface Design*. Addison-Wesley, Reading, MA, 1990.

[Gru95] Grudin, J., and Palen, L., Why groupware succeeds: Discretion or mandate? In *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work, ECSCW '95 (Stockholm, Sweden)*, pages 263–278. Kluwer Academic, Dordrecht, 1995.

[Hea91] Heath, C. and Luff, P., Collaborative activity and technological design: Task coordination in London Underground control rooms. In *Proceedings of the Second European Conference on Computer Supported Cooperative Work, ECSCW '91*. Kluwer Academic Publishers, Amsterdam, 1991.

[Hil95] Hill, W. Stead, L., Rosenstein, M., and Furnas, G., Recommending and evaluating choices in a virtual community of use. In *Proceedings of Human Factors in Computing Systems, CHI '95, (Denver, CO)*, pages 194–201. ACM Press, New York, 1995.

[Hof94] Hofman, J.D. and Rockart, J.F., Application templates: Faster, better and cheaper systems. *Sloan Management Review/Fall*, pages 49–60, 1994.

[Hol93] Holtzblatt, K. and Beyer, H., Making customer-centered design work for teams. *Communications of the ACM*, 36(10):93–103, October 1993.

[Hug92] Hughes, J.A., Randall, D., and Shapiro, D., Faltering from ethnography to design. In *Proceedings of the Conference on Computer Supported Cooperative Work, CSCW'92, (Toronto, Canada)*, pages 115–122. ACM Press, New York, 1992.

[Ish99] Ishii, H., Integration of shared workspace and interpersonal space for remote collaboration. In Beaudouin-Lafon, M. (Ed.), *Computer Supported Cooperative Work*, Trends in Software Series 7:83–102. John Wiley & Sons, Chichester, 1999.

[Joh88] Johansen, R., *Groupware: Computer Support for Business Teams*. The Free Press, New York, 1988.

[Jor95] Jordan, B. and Henderson, A., Interaction analysis: Foundations and practice. *J. Learn. Sci.*, 4(1):39-102, 1995.

[Jor96] Jordan, B., Ethnographic workplace studies and CSCW. In D. Shapiro, M. Tauber, and R. Traunmuller (Eds.), *The Design of Computer Supported Cooperative Work and Groupware Systems*, pages 17–42. Amsterdam: Elsevier Science, Amsterdam, 1996.

[Kat93a] Katzenbach, J. R. and Smith, D. K., *The Wisdom of Teams: Creating the High-Performance Organization*. Harvard Business School Press, Boston, MA, 1993.

[Kat93b] Katzenberg, B. and Piela, P., Work language analysis and the naming problem. *Communications of the ACM*, 36(4):86–92, April 1993.

[Kra90] Kraut, R.E., Fish, R.S., Rot, R.W., and Chalfonte, B.L., Informal communication in organizations: Form, function and technology. Reprinted in R.M. Baecker (Ed.), *Readings in Groupware and Computer-Supported Cooperative Work*, pages 287–314. Morgan Kaufmann, 1990.

[Kra94] Kraut, R.E., Cool, C., Rice, R.E., and Fish, R.S., Life and death of new technology: Task, utility and social influences on the use of a communication medium. In R. Furuta and C. Neuwirth (Eds.), *Proceedings of Conference on Computer Supported Cooperative Work, CSCW '94 (Chapel Hill, North Carolina)*, pages 13–21. ACM Press, New York, 1994.

[Kuk92] Kukla, C., Clemens, E.A., Morse, R.S. and Cash, D., Designing effective systems: A tool approach. In Paul Adler and Terry Winograd, (Eds.), *Usability: Turning Technologies into Tools*, pages 41–65. Oxford University Press, New York, 1992.

[Lin91] Linde, C., What's next?: The social and technological management of meetings. *Pragmatics* 1(3), 1991.

[Mac99] Mackay, W.E., Media spaces: Environments for informal multimedia interaction In Beaudouin-Lafon, M. (Ed.), *Computer Supported Cooperative Work*, Trends in Software Series 7:55–82. John Wiley & Sons, Chichester, 1999.

[Mal95] Maltz, D. and Ehrlich, K., Pointing the Way: Active Collaborative Filtering. In *Proceedings of Human Factors in Computing Systems, CHI'95 (Denver, CO)*, pages 202–209. ACM Press, New York, 1995.

[Man89] Mantei, M., Observations of executives using a computerized supported meeting environment. Reprinted in R.M. Baecker (Ed.), *Readings in Groupware and Computer-Supported Cooperative Work*, pages 695–708. Morgan Kaufmann, 1989.

[Man91] Mantei, M.M., Baecker, R.M., Sellen, A.J., Buxton, W.A.S. Milligan, T., and Wellman, B., Experiences in the use of a media space. In *Proceedings of Human Factors in Computing Systems, CHI '91 (New Orleans, LA)*, pages 203–208. ACM Press, New York, 1991.

[Mar95] Mark, G., Haake, J.M., Streitz, N.A., The use of hypermedia in group problem solving: An evaluation of the DOLPHIN electronic meeting room environment. In *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work, ECSCW '95 (Stockholm, Sweden)*, pages 197–213. Kluwer Academic, Dordrecht, 1995.

[Mar90] Markus, M.L. and Connolly, T., Why CSCW applications fail: Problems in the adoption of interdependent work tools. In *Proceedings of Conference on Computer Supported Cooperative Work, CSCW '90*, pages 371–380. ACM Press, New York, 1990.

[McG84] McGrath, J.E., *Groups: Interaction and Performance*. Prentice-Hall, 1984.

[Mic97] Michalski, J., Conversation on the Net. *Release 1.0 newsletter*, January 1997.

[Min91] Minneman, S.L. and Bly, S., Managing à trois: A study of multi-user drawing tool in distributed design work. In *Proceedings of Human Factors in Computing Systems, CHI '91 (New Orleans, LA)*, pages 217–224. ACM Press, New York, 1991.

[Mul93] Muller, M.J. and Kuhn, S. (Eds), Special Issue on Participatory Design. *Communications of the ACM*, 36(6), June 1993.

[Mul95] Muller, M.J., Carr, R., Ashworth, C., Diekmann, B., Wharton, C., Eickstaedt, C., and Clonts, J., Telephone operators as knowledge workers: Consultants who meet customer needs. In *Proceedings of Human Factors in Computing Systems, CHI '95 (Denver, CO)*, pages 130–137. ACM Press, New York, 1995.

[Nar93] Nardi, B., *A Small Matter of Programming: Perspectives on End User Computing.* MIT Press, Cambridge, MA, 1993.

[Nar96] Nardi, B and O'Day, V., Intelligent agents: What we learned at the library. *Libri*, 46(3):59–88, September 1996.

[Nar98a] Nardi, B., A web on the wind: The structure of invisible work. *Special issue of CSCW*, 1998.

[Nar98b] Nardi, B., Miller, J.R. and Wright, D.J., Collaborative, programmable intelligent agents. *Communications of the ACM*, In press.

[Nun91] Nunamaker, J.F., Dennis, A.R., Valacich, J.S., Vogel, D.R., and George, J.F., Electronic Meeting Systems to Support Group Work. *Communications of the ACM*, 34(7):40–61, July 1991.

[OHa94] O'Hara-Devereaux, M and Johansen, R., *Global Work: Bridging Distance, Culture and Time*. Jossey-Bass, San Francisco, 1994.

[Orl92] Orlikowski, W.J., Learning from Notes: Organizational issues in groupware implementation. In J. Turner and R. Kraut (Eds.), *Proceedings of the Conference on Computer Supported Cooperative Work, CSCW'92, (Toronto, Canada)*, pages 362–369. ACM Press, New York, 1992.

[Orl97] Orlikowski, W.J. and Hofman, J.D., An improvisational model of change management: The case of groupware technologies. *Sloan Management Review/Winter*, 38(2), 1997.

[Plo95] Plowman, L. Rogers, Y, and Ramage, M., What are workplace studies for? In H. Marmolin, Y. Sundblad, K. Schmidt (Eds.), *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work, ECSCW '95 (Stockholm, Sweden)*, pages 309–324. Kluwer Academic, Dordrecht, 1995.

[Pra99] Prakash, A., Group editors. In Beaudouin-Lafon, M. (Ed.), *Computer Supported Cooperative Work*, Trends in Software Series 7:103–133. John Wiley & Sons, Chichester, 1999.

[Pyc96] Pycock, J. and Bowers, J., Getting others to get it right: An ethnography of design work in the fashion industry. In *Proceedings of Conference on Computer Supported Cooperative Work, CSCW '96 (Boston, MA)*, pages 219–228. ACM Press, New York, 1996.

[Res94] Resnick, P., Iacovou, N., Suchak, M. Bergstrom, P., and Riedl, J., GroupLens: An open architecture for collaborative filtering of Netnews. In *Proceedings of Conference on Computer Supported Cooperative Work, CSCW '94 (Chapel Hill, North Carolina)*, pages 175–186. ACM Press, New York, 1994.

[Rog97] Rogers, Y. and Bellotti, V., Grounding blue-sky research: How can ethnography help? *interactions*, pages 58–63, May+June 1997.

[Sac95] Sachs, P., Transforming work: Collaboration, learning and design. *Communications of the ACM*, 38(9):36–44, September 1995.

[Sch93] Schuler, D. and Namioka, A., *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, NJ, 1993.

[Sha95] Shardanand, U and Maes, P., Social information filtering: Algorithms for automating "Word of Mouth". In *Proceedings of Human Factors in Computing Systems, CHI'95 (Denver, CO)*, pages 210–217. ACM Press, New York, 1995.

[Sny98] Snyder, W.M., Communities of practice: Combining organizational learing and strategy insights to create a bridge to the 21st century. *Organization Development and Change*, 1998.

[Spr93] Sproull, L. and Kiesler, S., *Connections: New Ways of Working in the Networked Organization*. MIT Press, Cambridge, MA, 1993.

[Sta94] Star, S.L. and Ruhleder, K., Steps towards an ecology of infrastructure: Complex problems in design and access for large-scale collaborative sytems. In *Proceedings of Conference on Computer Supported Cooperative Work, CSCW '94 (Chapel Hill, North Carolina)*, pages 253–264. ACM Press, New York, 1994.

[Suc83] Suchman, L., Office procedures as practical action: Models of work and system design. *ACM Transactions on Office Information Systems*, 1(4):320-328, 1983.

[Suc95] Suchman, L., Making work visible. *Communications of the ACM*, 38(9):56–64, September 1995.

[Tan91] Tang, J.C., Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies*, 34(2):143–160, February 1991.

[Tan94a] Tang, J.C., Isaacs, E.A. and Rua, M., Supporting Distributed Groups with a Montage of Lightweight Interactions. In R. Furuta and C. Neuwirth (Eds.) *Proceedings of Conference on Computer Supported Cooperative Work (Chapel Hill, North Carolina)*, pages 23–34. ACM Press, New York, 1994.

[Tan94b] Tang, J.C. and Rua, M., Montage: Providing teleproximity for distributed groups. In *Proceedings of Human Factors in Computing Systems, CHI '94 (Boston, MA)*, pages 37–43. ACM Press, New York, 1994.

[Tol96] Tollmar, K. Sandor, O. and Schomer, A., Supporting social awareness@work: Design and experience. In *Proceedings of Conference on Computer Supported Cooperative Work, CSCW '96 (Boston, MA)*, pages 298–307. ACM Press, New York, 1996.

[Tur95] Turkle, S., *Life on the Screen: Identity in the Age of the Internet*. Simon and Schuster, New York, 1995.

[Whi94] Whittaker, S., Frohlich, D., and Daly-Jones, O., Informal workplace communication: What is it like and how might we support it? In *Proceedings of Human Factors in Computing Systems, CHI '94 (Boston, MA)*, pages 131–137. ACM Press, New York, 1994.

[Zub88] Zuboff, S., *In the Age of the Smart Machine*. Basic Books, New York, 1988.