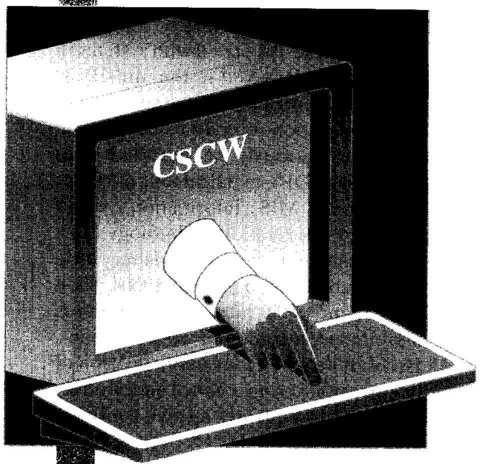


Computer-Supported Cooperative Work: History and Focus

Jonathan Grudin, University of California, Irvine

Ten years ago, Iren Greif of MIT and Paul Cashman of Digital Equipment Corporation organized a workshop that had far-reaching effects. Twenty people from different fields — but with a shared interest in how people work — gathered to explore technology's role in the work environment and coined the term "computer-supported cooperative work" to describe it.

Since then, thousands of researchers and developers have responded to this initiative. Although the first CSCW conferences were held in the United States, the topic was picked up immediately in Europe and Asia, where related work and serious interest already existed.

This article describes the people and the work found under the CSCW umbrella.

CSCW and groupware emerged in the 1980s from shared interests among product developers and researchers in diverse fields. Today, it must overcome the difficulties of multidisciplinary interaction.

Why 1984?

An earlier approach to group support, called "office automation," had run out of steam by 1984. OA's primary problem was not technical, although technical challenges certainly existed; it was in understanding system requirements. In the mid-1960s, tasks such as filling seats on airplane flights or printing payroll checks had been translated into requirements that resulted (with some trial and error) in successful mainframe systems. In the mid-1970s, minicomputers promised to support groups and organizations in more sophisticated, interactive ways, and OA was born. OA tried to extend and integrate single-user applications, such as word processors and spreadsheets, to support groups and departments. But what were the precise requirements for such systems?

Building technology was not enough. OA practitioners needed to learn more about how people work in groups and organizations and how technology affects that. Some engineers, notably Douglas Engelbart, whose early work at SRI foreshadowed much that came later, had made this point all along. Some people in management information systems (MIS) had promoted this approach as a way to improve success rates in large system development. But it had been largely absent from discourse among designers and developers in the vendor companies actually engaged in early efforts to develop group support applications.

CSCW started as an effort by technologists to learn from economists, social psychologists, anthropologists, organizational theorists, educators, and anyone else who could shed light on group activity. It has also become a place for system builders to share experiences and tell others of technical possibilities and constraints. Applications include desktop conferencing and videoconferencing systems, collaborative authorship applications, electronic mail and its refinements and extensions, and electronic meeting rooms or group support systems. Related, but not as strongly represented, application domains include computer-assisted design/computer-assisted manufacturing (CAD/CAM), computer-assisted software engineering (CASE), concurrent engineering, workflow management, distance learning, telemedicine, and real-time network conferences called MUDs (after "multiuser dungeons," although they're now used for more than playing games).

The acronym CSCW has survived a decade of use. It has been criticized for violating the maxim that four words are too many (some use "computer-supported collaboration" or CSC). It has also been criticized because "cooperative" work is often more a goal than a reality. "Workgroup computing," in contrast, shifts the focus from the work to the technology and restricts it to small organizational units. So does "groupware," used by Peter and Trudy Johnson-Lenz prior to 1984 and adopted by the CSCW community.

We now have annual conferences in groupware, focusing on commercial technologies, and in CSCW, addressing research into experimental systems and the nature of workplaces and organizations. I will rely on the terms CSCW and groupware to describe the research and the technology, respectively.

R&D contexts

Each ring in Figure 1 represents one focus of computer systems development and the principal customer or user of the resulting technology. Until recently almost all activity was in the highlighted outer and inner rings. The outer ring represents major systems and applications, primarily mainframe and large minicomputer systems designed to serve organizational goals for transaction processing, order and inventory control, computer integrated manufacturing, and so on. The inner ring represents applications designed primarily for individual users of PCs and workstations. These applications include word processors, debuggers, spreadsheets, games, and so forth. The two middle rings represent large projects and small groups. Large project support includes electronic meeting rooms and workflow automation systems, which are most useful for groups of six or more members. A major focus of small group support — computer-mediated communication (CMC) — includes desktop conferencing and collaborative writing applications, which may not work well with more than three or four users.

Most organizational sofware is unique and produced in house. Single-user applications are the province of commercial developers.

Software development. On the left in Figure 1 are the software development contexts that dominate in each ring. Software systems that support entire organizations (outer ring) are not bought at a local computer store or even at a local mainframe sales office. Some components might be acquired that way, but most software development is unique to the organization and is produced in house. In contrast, single-user applications (inner ring) are the province of commercial off-the-shelf product developers, who rely on the shrink-wrapped software market for sales and do little or no customization for individual customers.

The two middle rings represent groupware development. Government contracts have stimulated project-level software support. Small-group support has been a new focus for commercial product developers, and telecommunications companies are interested in multimedia technologies that create demand for high-bandwidth communication. As the arrows indicate, groupware development has origins in both preexisting contexts.

Research areas. On the right in Figure 1 are the research areas associated with system development and use in each context, and the date by which each area was firmly established. Starting with the outer ring, literature associated with systems in organizations arrived in the mid-1960s with the advent of integrated circuits and third-generation computer systems. The field has variously been called data processing (DP), management information systems (MIS), information systems (IS), and information technology (IT). As Friedman[1] noted, "There is very little on the subject up to the mid-1960s. Then the volume of literature on [computers and] the organization of work explodes. Issues of personnel selection, division of labour, monitoring, control and productivity all subsequently receive considerable attention."

Although the IS field has focused primarily on organizational support (outer ring), it also covers the management of large projects (next ring). In the early and mid-70s, the fields of software engineering (SE) and office automation (OA) emerged, focusing on computer support for large groups and projects. Computer support for software engineering is a specific kind of large project support — and a natural one given the high concentration of technology in SE development environments. The complexity of managing large government software contracts provided further incentive to apply technology to group work. Although OA did not survive as a field, many issues underlying distributed-project-management systems are again being addressed as "workflow."

The inner ring emerged next. With the spread of interactive systems in the late 1970s and early 1980s, research into single-user applications and interfaces blossomed. Work has been presented by the IEEE Human Factors Society and at ACM's SIGCHI (Special Interest Group on Computer-Human Interaction) conferences since 1983. The most recent to emerge is CSCW, with conferences held since 1986 and now alternating between North America and Europe.

The project-level ring is heard from less frequently at CSCW conferences. The most comprehensive collection of readings in groupware and CSCW,[2] with over 70 papers, contains nothing on workflow management or project-level software engineering support. CSCW conferences in the US emphasize small-group support (with some organization-level analysis).

Other potentially relevant work, such as computer-mediated education, is also underrepresented in the CSCW literature, probably because these fields have

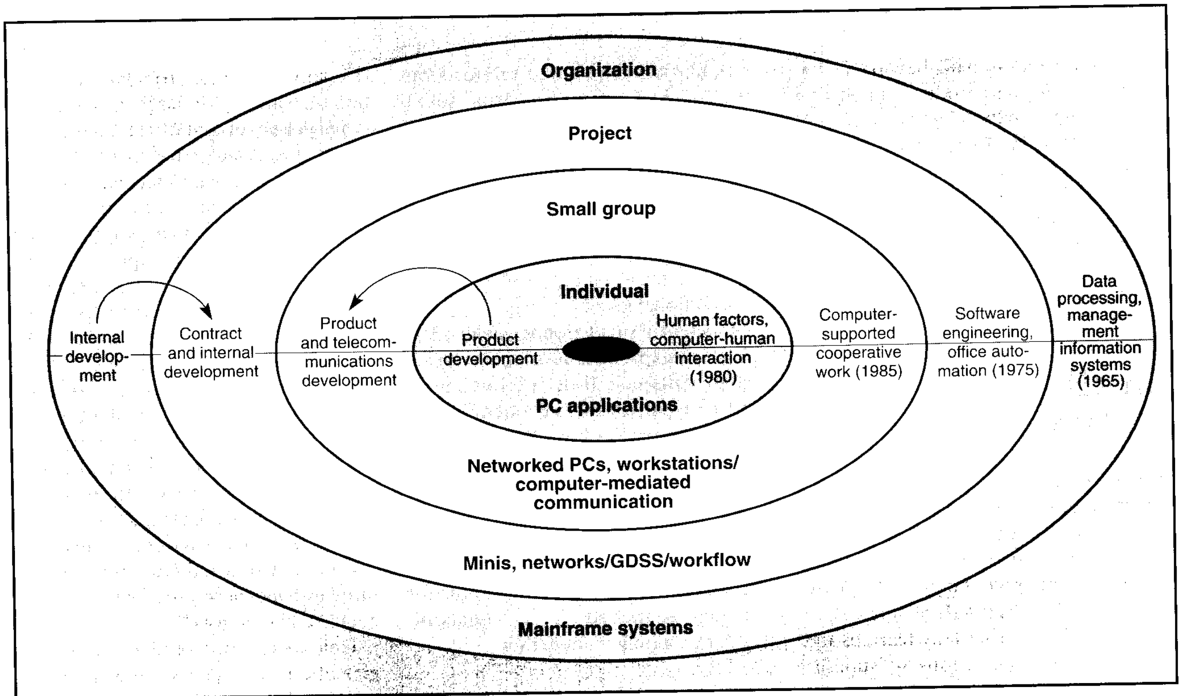


Figure 1. US research and development contexts for computer-supported cooperative work and groupware. Each ring defines a work level (organization, project, small group, or individual) and its corresponding systems (listed directly below the hub of the figure) and software development (left) and research (right) areas.

their own conferences and journals. In addition, their concerns are of less interest to product developers eyeing huge small-group markets and to IS developers focusing on internal and contracted systems development.

CSCW: Research that spans the boundaries. Because Figure 1 represents central tendencies, it obscures other perspectives, notably our ability to examine issues that transcend the divisions. CSCW is not restricted to one ring; it draws from each preexisting development culture. People study, for example, the use, in group and organizational settings, of applications developed for individual users; the ways in which software, developed to support groups, affects individuals and is adapted to different organizational contexts; and systems developed to support organizational goals as they act through individuals, groups, and projects.

CSCW participation. Table 1's analysis of attendance at seven conferences, including five early CSCW conferences, indicates that CSCW in the US and Japan grew primarily from the human-computer interaction field, which explains its focus on small-group applications. European research, on the other hand, stresses organizational and large-project issues.

Column 1 in Table 1 classifies attendees' or presenters' employing organizations. Columns 2 and 3 show a distinct correlation between ACM's CHI conferences and the US-based CSCW conferences. Columns 4 and 5 show an equally strong correlation between European-based CSCW conferences and the International Conference for Information Systems, the premier information-systems-research conference.

The similarity in composition of CHI and the US CSCW conferences suggests that CSCW in the US was fueled by com-

Table 1. Comparing attendance by employing organization for two CSCW conferences (middle columns) and for conferences on human-computer interactions and information systems (outer columns) helps explain the US/Japanese focus on small groups and the European focus on user organizations.

	US/Japanese Small-Goup Focus		European Organization Focus	
Employing Organization	CHI 90 Presenters	CSCW 86-90* Attendees	ECSCW 89, Crete 90 Attendees	ICIS 90 Attendees
Academic	40%	30%	70% **	85%
Product development	30%	40%	10% ***	1%
Telecommunications	10%	7%	5%	0%
Other	20%	23%	15%	14%

*Of 30 Japanese attendees, 55% were from product development companies and 25% from telecommunications. They presented 2 of 30 papers.
**Includes government research laboratories.
***Two thirds of attendees in this category were from the US.

puter companies moving beyond single-user applications to products supporting small-group activity. Thirty CSCW 1990 participants came from Japan. Like the American participants, they were primarily from product development and telecommunications companies, and English-language journal articles by Japanese contributors focus on small-group applications.

Small-group versus systems approach

The growing interest in small-group applications has been a major impetus for CSCW conferences. As PCs and workstations are networked, myriad small groups become potential markets. Vendors are enhancing mature single-user applications with groupware features and devising new applications to support communication and coordination. Simultaneously, telecommunications companies seek to increase the demand for bandwidth through applications that draw on multimedia technologies.

As off-the-shelf single-user product developers expand into computer support for groups, many are confronting issues in group dynamics for the first time. The design of applications such as word processors stressed perceptual and cognitive factors, and developers often succeeded with minimal attention to the workplaces in which single-user applications were used. With groupware, however, the social, motivational, and political aspects of workplaces become crucial.[3]

IS researchers and developers are more familiar with social dynamics, since organizational systems — mainframes and large minicomputers — have been in use for decades. The IS community did not establish the CSCW agenda but has much to contribute. It has other incentives to participate: Networked PCs, workstations, and software products are increasingly important components of organizational information systems. Also, as large systems decline in cost, they can be used by smaller organizational units. An example is the evolution of "group decision-support systems" — once expensive and marketed for use in high-level decision-making — into "group support systems," which are less expensive and flexible enough to support a variety of meeting types.

The small-group application and IS communities share some interests but have striking differences. Most small-group support emphasizes communication because small groups are generally formed to bring together people who need to communicate. Organizational systems focus more on coordination because coordinating the efforts of disparate groups is a major problem at the organizational level.[4]

Members of small groups usually share key goals, so product developers can anticipate relatively little friction among users and can assume a cooperative approach to technology use. This is directly reflected in the second "C" of CSCW. In contrast, researchers and developers focusing on organizational systems must attend to the conflicting goals generally present in organizations.[5,6] For that reason, some in the IS community argue for changing the second "C" to "collaborative" or dropping it altogether.

Product developers focus more on the human-computer interface. Organizational system developers fixate on functionality.

Another contrast is that product developers are more concerned with the human-computer interface, whereas organizational system developers and their customers are more fixated on functionality. Product developers compete in discretionary markets where useful functionality is quickly adopted by others, at which point the human-computer interface can provide an important edge. Internal IS developers often face unresolved design questions based on workplace functionality and cannot justify the cost of fine-tuning the interface.

These differences in priorities cause friction and confusion. I've heard speakers from the IS field berate small-group application developers for focusing on "cooperation" and ignoring conflict, and criticize research that focuses on the thin surface layer of the human-computer interface. On the other side, those working to resolve technical problems may question the value of research into organizational politics distant from their concerns.

It has been suggested that CSCW pits social scientists against technologists, but this may not be the real source of conflict. In large information system environments, decades of experience have brought nontechnological problems to the surface, whereas in small system environments, technological hurdles still predominate. For example, Scandinavian researchers and developers working on tools and techniques for collaborative design are often tied to the "social science" perspective, but they are actually computer scientists who came to realize the importance of social effects in the course of developing large systems. Conversely, many behavioral and social scientists are hired into industry research labs and become "technologists."

Unless we come to understand the origins of our differences, we will not succeed in addressing them.

US and European differences. American and European approaches to CSCW overlap but have marked differences. These partially reflect the small-system/product development and large-system/internal development distinction. In the US, where computer industry research labs and industry support for universities are very influential, research and development are more intertwined than in Europe, where more research is government sponsored and activity focuses on large-scale systems development, often in user organizations.

Many US researchers and developers focus on experimental, observational, and sociological data; others build technology and then look for ways to use it. These US approaches can be described as empirical: experiments by social psychologists looking at group activity among teams of students, anthropological descriptions of activity in schools and businesses, and descriptions of groupware that present interesting technical problems whether or not the technology is used.

European contributions to CSCW are often driven by philosophy or by social, economic, or political theory. Some European contributions to CSCW are explicitly grounded in the writings of philosopers, social theorists, and economists. (This does not characterize all European computer science or informatics, much of which is more formal.) The result may be a broad formulation

The challenge of being multidisciplinary

CSCW represents a merging of issues, approaches, and languages. Whether we view the shared and the disparate interests as a melting pot or a mixed salad, making sense of them is a lively process. Opportunities to learn and to inform generate enthusiasm—which is needed to overcome inevitable obstacles. It is not always apparent why another person's perspective and priorities differ. It takes patience to understand conflicts in priorities and to find mutually advantageous modes of operation.

Although it can be exciting to find a new source of information and a new audience, it can be frustrating when the other group is ignorant of work that you assume to be basic shared knowledge. The groups participating in CSCW are not always aware of the extent to which they have relied on different conferences, journals, and books.

Participants from different areas use the same terms in subtly different ways. What do the words "user" and "implementation" mean to you? In the field of human-computer interaction, "user" refers to a person sitting at a display, entering information and commands and using the output. In the information systems field, "user" refers to a user of the output, a person who might not touch a keyboard. To deal with the ambiguity, the latter field coined the term "end user" to describe a person at a terminal or keyboard. Thus, the term "user" has different connotations that two interlocutors are unlikely to recognize. Nor is that the end of it — in software engineering, the term "user" can mean the users of particular tools, who are themselves developers.

Similarly, "implementation" is synonymous with development or coding in HCI, but to the MIS world it describes the introduction of a new system into an organization. Many, perhaps most, terms are used differently by different research and development communities.

One must attend carefully to meaning when navigating the CSCW literature, including the articles in this issue of *Computer.*

of system requirements or an implementation of a platform to support a range of applications that in concert provide organizational support.

European CSCW also reflects cultural norms, such as national homogeneity, codetermination laws, strong trade unions, and extensive social welfare systems. At the risk of oversimplifying, greater cultural homogeneity can lead to a focus on skill augmentation justified on humanitarian as well as economic grounds: Workers losing automated jobs must be indirectly supported anyway. The Scandinavian participatory or collaborative design approach reflects these priorities.[6]

The work in England partly bridges US and European differences. Due to shared language and culture, perhaps, several US technology companies have active research labs in England. The most notable fusion of US and European approaches comes from Xerox's prolific Cambridge EuroPARC research center. In collaboration with academic researchers, their work includes experimental studies of collaborative writing, sociological analyses of group activity in settings ranging from development laboratories to the London Underground control room, and the construction and use of video communication systems.

CSCW in Europe is supported by an enormous variety of grants. Major European Community projects funded by the European Strategic Programme for Research and Development in Information Technology (ESPRIT) and Research and Development in Advance Communications Technology in Europe (RACE) explicitly bring together researchers and developers from different countries and require both academic and industry partners. Some projects involve tightly coupled work, others consist of more independent efforts at each site. These projects are exercises in cooperative work whose content is CSCW research and development.

Another effort to build cooperation among researchers and developers in the European Community countries is the CO-TECH project, carried out under the Cooperation in Science and Technology (COST) framework. It provides funding for organizing and attending meetings, not for research itself, and has succeeded in building a sense of community.

In addition, many European governments directly fund CSCW research through government research laboratories and specific government projects. One example is a major German effort to develop an infrastructure to support the division of the country's capital between Bonn and Berlin. (NSF is an important supporter of US CSCW projects, but it is less influential than European funding agencies in shaping the research agenda.)

The CSCW 1992 conference illustrated these differences. European presentations included two based on multinational ESPRIT projects and none from computer companies. The ESPRIT presentations described a working "model for automatic distributed implementation of multiuser applications" and a description of the requirements for supporting the Great Belt bridge/tunnel project in Denmark. In contrast, US and Japanese contributions reflected a strong product and telecommunications orientation.

Will the different priorities of the active researchers and organizations in Europe and the US persevere? Should they? Can the groups interact despite the differences?

Conferences have had limited success in drawing from both groups simultaneously. Philosophically oriented European submissions often strike empirically oriented American reviewers as lacking in content; American contributions strike European reviewers as unmotivated or shallow. Differences in terminology block understanding. For example, I listened to a European CSCW researcher criticize an American group's understanding of "task analysis." The Americans used the term to describe a cognitive task analysis based on experimental interface testing, a standard practice in human-computer interface studies. To the European, "task analysis" meant an organizational task analysis based on mapping the flow of information from person to person. He thought the term was "nonsensical" in an experimental setting.

Cultural perceptions of the role of research meetings exacerbate the split. In

Europe, conferences are often gatherings of professionals to interact and share current results; most of those who attend also present. In the US, a conference is often organized for a larger audience, with greater emphasis on polished results. This difference leads to misunderstandings over submission requirements.

CSCW in Japan. Thus far, the principal Asian impact on Western CSCW and groupware research has come from Japan, where government and industry cooperation in technology development includes support for CSCW.

Japanese CSCW research, only recently published in English, indicates widespread interest in technological support for group processes. The "software factory" concept and the interest in process programming (formal workflow management systems) are examples. Contributions to CSCW have come primarily from computer and software companies, including NEC and Toshiba, and telecommunications companies, including NTT and ATR. In this respect Japanese participation matches the US's nonacademic profile.

It is often suggested that Japanese enthusiasm for collaboration and consensus will increase groupware acceptance, but closer examination reveals a more complicated reality. Ishii[7] notes that the importance in Japan of showing consensus in meetings often leads to real decision-making in private discussions, eliminating a role for meeting support software. The Japanese preference for personal contact and direct interaction could even cause resistance to technological mediation. One should avoid a too-precipitous prediction of groupware technology's success in another culture.

Defining groupware

We can now examine a recurring question: What should be included under the rubric of "groupware" or "CSCW applications"?

Figure 2 outlines the controversy. Everything above each arrow is labeled "groupware" by the author or authors to the right. Researchers and developers with a small-system orientation lie at the bottom of this sequence and those with an organizational perspective are at the top.

Starting at the bottom, Crowley[8] argued that the single greatest impediment to computer support for workgroup collaboration is the lack of software permit-

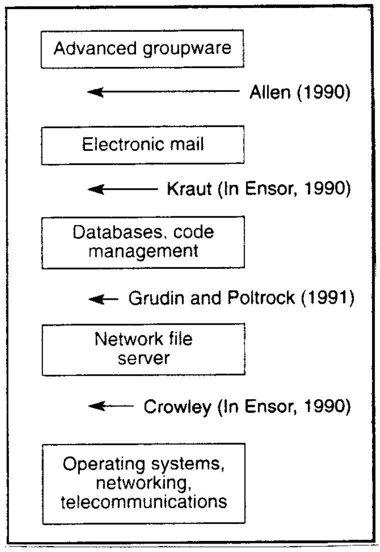


Figure 2. Groupware and its substrate. The authors define all items listed above their names as groupware. Their orientation progresses from the system level (bottom) to the advanced application level (top).

ting interaction across networked PCs. He felt that network file servers and related software are of central importance and should be considered groupware. Grudin and Poltrock[9] considered multiuser software, such as large databases and version control systems, groupware because it provides informative success cases. Kraut[8] argued that electronic mail is the only successful CSCW application, eliminating from consideration successful software below that line.

The argument for excluding multiuser databases is that they support groups by providing the illusion that every user has independent access; however, apart from password control, these databases are not aware of different roles or communications needs in a group. They are not "group-aware." By this definition, a database with triggers to alert specific people to specific changes would qualify as groupware.

Finally, Allen[10] is among those who argue that electronic mail is itself a substrate for groupware applications but should not be called groupware due to its insensitivity to organizational or group qualities.

Perhaps blanket categorization of an application is less helpful than considering its use in a particular setting. E-mail used only to broadcast organization-wide messages is not supporting groups, but an e-mail system with users who create aliases, distribution lists, and complex patterns of use that differ across projects does qualify as group support technology. This position is intellectually defensible, but the reality is that researchers, developers, and marketers want general, site-independent labels. Thus, no formulation will satisfy everyone engaged in CSCW research and development.

Groupware typologies

A multidisciplinary perspective lets us elaborate a familiar groupware typology. Figure 3, a variant of DeSanctis and Gallupe's[11] widely used space and time categorization, places representative applications in different cells. Activity can be carried out in a single place (top row); in several places known to the participants, as in electronic mail exchanges (middle row); or in numerous places, not all of which are known to participants, as in a message posted to a netnews group (bottom row). Activity can be carried out "in real time" — that is, in one unbroken interval, as in a meeting (left column). Alternatively, it can be carried out at different times that are highly predictable or constrained, as in sending mail to a colleague and expecting it to be read within a day or so (middle column). Or it can be carried out at different times that are unpredictable, as in open-ended collaborative writing projects (right column).

Activities do not always match Figure 3 precisely — for example, one collaborative writing project could take place in a single session, but another could involve an unpredictable, large set of people assembling a lengthy document. Some cells have enjoyed more computer support than others; for example, interactive multicast seminars are only starting to appear as a "same time, unpredictable place" activity.

Such a typology is easy to learn. It facilitates communication and is widely used, especially by groupware developers, but not without risk: Figure 3 obscures an organizational perspective. Most real work does not fall into one or another category. As we go about our work, we generally engage in some face-to-face meetings and some distributed and asynchronous communication. Most work involves both communication and coordination. Narrow tasks interact with broader work activities, and even the broadest concerns overlap and impact one another.

Technology designed to support activity in one cell can fail by negatively impacting activity in another. For example, a stand-alone meeting support system that provides no access to existing databases or other on-line materials may be useless in some situations. Noting the interdependencies among activities, Robert Johansen of the Institute for the Future calls for "any time, any place" support.

A typology hobbles groupware developers if it focuses attention too narrowly. At the same time, it serves legitimate purposes by, for example, identifying applications that pose common technical challenges, such as those dealing with concurrent activity.

Figure 3. This 3-by-3 map of groupware options categorizes representive applications according to place and time. It facilitates communication and is widely used by groupware developers.

Place \ Time	Same	Different but predictable	Different and unpredictable
Same	Meeting facilitation	Work shifts	Team rooms
Different but predictable	Tele/video/ desktop conferencing	Electronic mail	Collaborative writing
Different and unpredictable	Interactive multicast seminars	Computer bulletin boards	Workflow

An example: The history of GSS

The history of a "same time, same place" technology, meeting support systems, illustrates several points I've made.

These systems were originally a central component of GDSS (group decision-support systems). Unlike most groupware applications, meeting support did not emerge from product development environments, nor did papers on GDSS appear in human-computer interaction conferences. Until recently, there were no electronic meeting room products.

GDSS research and development began over 20 years ago in the IS field in US business schools. To understand its history, consider the "D" in GDSS. Decision-making was emphasized because until recently, management-as-decision-making was the dominant perspective in schools of business and management.[12] In addition, expensive early systems could be justified in organizations and in a management school curriculum by emphasizing high-level decision-making.

In the mid-1980s, the first CSCW conferences drew GDSS researchers from the IS field. Conflicting use of terminology went unrecognized. The IS community construed GDSS broadly to include all technology that contributes to decision-making, including electronic mail and other common applications. In fact, some in the IS field considered GDSS to be a synonym for CSCW. Encountering the term GDSS for the first time, many from the HCI field assumed it referred only to electronic meeting support, the one technology feature unfamiliar to them. (They also thought in terms of applications, not systems.)

As the cost of the technology fell, GDSS use was no longer restricted to high-level "decision-makers." It could be used to support meetings of various kinds. In addition, the trend toward corporate downsizing has lessened the emphasis on high-level decision-making. As rungs are removed from an organizational ladder, responsibility for decisions often shifts to the groups that implement them. As a result, the "D" was dropped to form GSS (group support systems). The reduced cost, together with improved technology and a better understanding of the process of effective use,[3] led to electronic meeting rooms becoming commercial products around 1990.

GSS is support for projects or large groups — meeting support is not as useful with fewer than 5 or 6 participants. The small-group application developers who play a central role in CSCW have different priorities than the system developers working on GSSs, and few GSS papers have been accepted for CSCW conferences. In addition, GSS researchers observed that small-systems researchers were unfamiliar with their literature. Over time, the GSS community has become less involved in CSCW. They have participated in conferences with an IS orientation, initiated a newsletter that rarely mentions CSCW, and spawned their own journals. They have, however, adopted the "groupware" label, as has the workflow management community—another group focused on large group support.

At the moment, the term "groupware" is found in both GSS and CSCW literatures, used to describe overlapping but different technologies. The division is only partial and may be temporary. IS research is still presented at CSCW meetings, and both groups benefit from their interaction. But the fragile nature of participation in CSCW is apparent.

Some writers describe CSCW as an emerging field or discipline, but what we see today resembles a forum, an undisciplined marketplace of ideas, observations, issues, and technologies. We expect to find shared or overlapping interests, but we should anticipate differences in interests and priorities.

If we think of CSCW as an emerging field or common enterprise, we may be frustrated by this mosaic of different pieces, the frequent misunderstandings, and the lack of intellectual coherence. But when understood and respected, the differences form the core of richer, shared understandings. ■

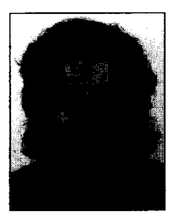
Acknowledgment

Steven Poltrock helped shape these ideas. I also benefited from discussions with Clarence Ellis, Gerhard Fischer, and Brad Hartfield. Mark Ackerman, Jeffrey Blevins, Ruven

Brooks, Alan Dennis, Gerardine DeSanctis, Rebecca Grinter, Suzanne Iacono, Hiroshi Ishii, Peter and Trudy Johnson-Lenz, Stefanie Lindstaedt, Jeanne Pickering, Dianne Murray, Mike Robinson, Gen Suzuki, Richard Taylor, and Doug Vogel provided useful comments or information.

References

1. A.L. Friedman, *Computer Systems Development: History, Organization, and Implementation,* Wiley, Chichester, UK, 1989.
2. R. Baecker, *Readings in Groupware and Computer-Supported Cooperative Work*, Morgan Kaufmann, San Mateo, Calif., 1993.
3. J. Grudin, "Groupware and Social Dynamics: Eight Challenges for Developers," *Comm. ACM*, Vol. 37, No. 1, Jan. 1994, pp. 92-105.
4. T.W. Malone and K. Crowston, "The Interdisciplinary Study of Coordination," *ACM Computing Surveys*, Vol. 26, No. 1, Mar. 1994, pp. 87-119.
5. R. Kling, "Cooperation, Coordination, and Control in Computer-Supported Work," *Comm. ACM*, Vol. 34, No. 12, Dec. 1991, pp. 83-88.
6. M. Kyng, "Designing for Cooperation: Cooperating in Design," *Comm. ACM*, Vol. 34, No. 12, Dec. 1991, pp. 64-73.
7. H. Ishii, "Cross-Cultural Communication and Computer-Supported Cooperative Work," *Whole Earth Review,* Winter 1990, pp.48-52.
8. "How Can We Make Groupware Practical?" R. Ensor (panel moderator), *Proc. CHI 90*, ACM, New York, 1990, pp. 87-89.
9. J. Grudin and S. Poltrock, *Computer-Supported Cooperative Work*, CHI 91 Tutorial Notes, 1991 (available from the author).
10. C. Allen, "Definitions of Groupware," *Applied Groupware*, Vol. 1, 1990, pp. 1-2.
11. G. DeSanctis and B. Gallupe, "A Foundation for the Study of Group Decision Support Systems," *Management Science*, Vol. 33, No. 5, May 1987, pp. 589-609.
12. J. King, K. Ruhleder, and J. George, "ODSS and the Twilight of the Decision Support Movement: Social Segmentation and the Legacy of Infrastructure," *Proc. 25th Hawaii Int'l Conf. Systems Sciences, Vol. IV*, IEEE CS Press, Los Alamitos, Calif., 1992, pp. 472-481.



Jonathan Grudin is an associate professor of information and computer science at the University of California, Irvine. He previously taught at Aarhus University in Denmark, developed software at Wang Laboratories, and carried out research at MCC. He is in the Computers, Organizations, Policy, and Society Group at Irvine, where his interests include CSCW and human-computer interaction. He received a BA in mathematics-physics from Reed College, an MS in mathematics from Purdue University, and a PhD in cognitive psychology from the University of California, San Diego.

The author can be reached at the Information and Computer Science Dept., University of California Irvine, Irvine, CA 92717; e-mail grudin@ics.uci.edu.