As the information highway becomes a reality, and appropriate hardware and software become readily available, anyone using a computer will be able to engage in some form of computer-supported cooperative work within the next few years. Computer-supported cooperative work refers to people working together on a product, research area, topic, or scholarly endeavor with help from computers. This field is also known as computer-supported collaboration, GroupWare, Workflow, and Group Decision-Support Systems. No matter what the name, it is a very active, fertile area for research, development, and production.

But much work remains to be done to bring computer-supported cooperative work (CSCW) into everyday use, including:

- developing user-friendly software,
- addressing the social dynamics of group activities,
- standardizing various terms, and
- handling the difficult interactions between multiple tasks performed by multiple groups.

A primary key to success in CSCW activities lies in user comfort with system operations. The heart of this comfort is the user interface. Present interfaces need to become more intuitive. We expect benefits in this area from ongoing Human-Computer Interface research in the multimedia field.

In terms of social dynamics, users need to learn the best way to work in interactive group situations. There is no universally accepted protocol to guide CSCW groups. This research area links computer science to sociology.

Definitions are also needed in the field of language comprehension. Spoken and written language, without standard terminology, is frequently ambiguous and often leads to misunderstandings. Using computer-based language translators and interpreters can help develop common understanding. This area needs a great deal of work.

The issues that confront a single group are multiplied when multiple groups tackle the same problems. New needs for consistency across groups, concurrency of information or simulations, timely updates, and language comprehension arise. Emerging solutions appear to be moving in the direction of multigroup networking to compensate for lack of information concurrency.

**CSCW modes**

Understanding and solving the main issues presented by CSCW depend on conceptualizing how people work. People work together in many modes. They may be in the same
room at the same time working on activities ranging from group-decision operations to group authoring to running a CAD program. This is called synchronous mode. If these activities take place at the same time but participants are located at different sites, they are working in distributed synchronous mode (see Figure 1). If these activities are taking place at different times but in the same location, they are asynchronous; if they are taking place at different sites at different times, they are distributed and asynchronous. Each mode requires hardware and software support systems for effective and efficient operation. It is important for users to recognize the distinctiveness of each mode, since their protocols and network and storage requirements vary. (Table 1 presents a matrix of several typical CSCW application areas to show which modes are used the most.)

On the road to solutions

This special issue of *Computer* addresses some of the difficult problems relating to technological developments that will enable people to work in any mode. The issue presents software designs, systems, and applications intended to enhance CSCW in a variety of modes, addressing the problems that arise when people work together and how these interactions are affected by computer interfaces. It includes an approach to handling language-comprehension constraints that capitalizes on information-retrieval work. Finally, it covers some of the problems associated with multiple tasks in a distributed environment and multigroup

---

**General glossary of terms**

**Asynchronous** — Group-functioning mode that occurs at different times in the same place.

**Computer-supported cooperative work (CSCW)** — A system that integrates information processing and communications activities to help individuals work together as a group.

**Distributed asynchronous** — Group-functioning mode that occurs at different times and places.

**Distributed synchronous** — Group-functioning mode that occurs at the same time but at different places.

**Group Decision-Support System (GDSS)** — A computer-based system that supports a group's decision process.

**Homogeneous group** — A group composed of individuals with similar experiences and background information relating to the group's task.

**Loosely coupled group** — A group composed of individuals who normally do not work together or address common problems in a consensus-seeking fashion. These individuals may not be electronically coupled as a group.

**Multigroup Decision-Support System (MGDSS)** — A system designed to assist multiple small groups of individuals (who are frequently loosely coupled and nonhomogeneous) in addressing a common problem.

**Nonhomogeneous group** — A group composed of individuals who do not have similar experiences and background information relating to the group's task.

**Spatially distributed** — Occurring at different locations

**Synchronous** — Group-functioning mode that occurs at the same time in the same place.

**Temporally distributed** — Occurring at different times (see asynchronous).

**Tightly coupled group** — A group of people who interact through a computer over a network to address a common problem.


decision-support systems. Due to the fact that designers are approaching CSCW from several perspectives based on geography and culture, this issue also provides an international variety of approaches.

The introductory article, Computer-Supported Cooperative Work: History and Focus, by Jonathan Grudin, provides an excellent historical overview, stressing two distinct perspectives: European and American. The European perspective focuses on organizational and large project issues. In contrast, the US perspective moves beyond single-user applications to products supporting small-group activity. The Asian perspective, although it may be viewed as a third point of view, is very similar to that of the US.

The European perspective is evident in CSCW Tools: Concepts and Architectures, by Walter Reinhard, Jean Schweitzer, Gerd Völkken, and Michael Weber. They present two approaches, which they combine to provide an advanced CSCW framework. Architectural Support for Cooperative Multiuser Interfaces, by Richard Bentley, Tom Rodden, Pete Sawyer, and Ian Somerville, provides another look at the European perspective by describing a software architecture that supports the development of reusable, tailorable, multiuser interfaces.

The US perspective appears in the next group of articles. Prototyping Synchronous Group Applications, by Ivan Tou, Steven Berson, Gerald Estrin, Yadran Eterovic, and Elsie Wu, deals with synchronous group applications that have a strong sharing style. The authors describe a system that lets application designers prototype a strongly shared application by graphically specifying the application's data model, structure, and behavior. The next article, Collaborative Systems: Solving the Vocabulary Problem, by Hsinchun Chen, examines a system design problem created when multiple users adopt different terms to describe a similar concept and discusses two tools that help alleviate vocabulary differences. Multigroup Decision-Support Systems in CSCW, by James D. Palmer, N. Ann Fields, and Peggy Lance Brouse, describes a distributed synchronous mode environment, including a process, and the software to support the process, for a large group of individuals working in the same problem area.

The Asian perspective is evidenced by the final article. In When Client/Server Isn't Enough: Coordinating Multiple Distributed Tasks, by Ting-Peng Liang, Hsiangchu Lai, Nian-Shing Chen, Hungshiung Wei, and Meng Chang Chen, the authors examine issues involved in coordinating multiple tasks in a distributed group decision environment. They propose a three-layered architecture for flexibility in task coordination and present a prototype system using this architecture.

Acknowledgments

We are deeply grateful to Dave Rine, former associate editor of Computer, for supporting our proposal, to all the reviewers for providing excellent reviews in a timely fashion, and to Ted Lewis, Computer's editor-in-chief, for his support throughout the preparation and realization of this special issue.

Reference


James D. Palmer is BDM International Professor of Information Technology at George Mason University in Fairfax, Virginia. He was Director of Electrical Engineering at the University of Oklahoma, Dean of Engineering at Union College, and President of Metropolitan State College. He served in the US Department of Transportation as Administrator of Research and Special Programs Administration and in industry as vice president at MIT and JJ Henry Inc. His research interests are in software engineering, group-decision support systems, computer-supported cooperative work, multimedia systems, and requirements engineering.

Palmer received a BS and an MS from the University of California at Berkeley in 1955 and 1957, respectively, and a PhD from the University of Oklahoma in 1963. He has authored three books and 75 papers. He is a Fellow of the IEEE, a member of the IEEE Systems, Man, and Cybernetics Society (president, 1976-78), a member of the IFIP Computer Society, and a registered Professional Engineer.

N. Ann Fields is a research assistant professor of systems engineering in the School of Information Technology and Engineering at George Mason University. Her research interests include group-decision support systems, computer-supported cooperative work, and requirements engineering.

Fields received a BS in mathematics from the University of Miami in 1974, an MS in operations research and statistics from the University of Southern Mississippi in 1980, and a PhD in information technology from George Mason University in 1991. She is a member of the IEEE and the Society of Professional Women Engineers.

Readers can contact the authors at the Center for Software Systems Engineering, George Mason University, Fairfax, VA 22030-4444.