1995

A Taxonomy of workgroup Computing Applications

Warren Von Worley
Nova Southeastern University, wvon@comcast.net

A Taxonomy of Workgroup Computing Applications

by

Warren Von Worley

A Dissertation Report Paper Submitted to the
School of Computer and Information Sciences of Nova
Southeastern University in Partial Fulfilment of the Requirements for the
Degree of Doctor of Philosophy

Nova Southeastern University

January 1995

Trademarks

All terms in this thesis that are known to be trademarks are listed below. In addition, terms suspected of being trademarks have been capitalized using the manufacturer's capitalization style. Use of a term in this thesis should not be regarded as affecting the validity of any trademark. All trademarks are the property of their respective owners.

Excel, MS-DOS, Windows 3.1, Windows NT, Windows for Workgroups, and Windows 95, are trademarks of Microsoft Corporation.

LinkWorks Client, TeamLinks, PathWorks, and Alpha are trademarks of Digital Equipment Corporation.

Lotus Notes is a trademark of Lotus Development Corporation.

Macintosh is a trademark of Apple Corporation.

NeXTstep is a trademark of NeXT Corporation.

OS/2 is a trademark of IBM Corporation.

Pentium is a trademark of Intel Corporation.

PowerPC is a trademark of Motorola Corporation.

SPARC is a trademark of Sun Microsystems.

WordPerfect is a trademark of WordPerfect Corporation.

To Ann, my lovely wife, who supported me throughout this undertaking and never waivered in her belief that I would complete this difficult task.

Acknowledgements

The author wishes to express his sincere thanks and appreciation to professor Deitel, dissertation chair, and professors Dringus and Kemper Littman for their dedication, support, insight, and time spent guiding me towards this goal. Also, thanks to my good friend, Mike Brasher. I would have never started at Nova Southeastern University if it were not for his help in telling me about this program.

Abstract

The goal of workgroup computing is to help individuals and groups efficiently perform a wide range of functions on networked computer systems (Ellis, Gibbs, & Rein, 1991). Early workgroup computing tools were designed for limited functionality and group interaction (Craighill, 1992). Current workgroup computing applications do not allow enough control of group processes and they provide little correlation between various workgroup computing application areas (Rodden and Blair, 1991). An integrated common architecture may produce more effective workgroup computing applications. Integrating common support functions into a common framework will avoid duplication of these functions for each workgroup computing application (Pastor & Jager, 1992).

Over 50 research and commercial workgroup computing applications were analyzed to understand and discover their distinctive characteristics and fundamental structure. Using the specified methods, a detailed section of a workgroup computing taxonomy was synthesized for each of 11 workgroup computing functional areas. The detailed taxonomy was the consolidation of all the hierarchical structures. The taxonomy formed the basis for developing an integrated workgroup computing architecture and a set of workgroup computing Application Programming Interface (API) specifications.

The results of this study support the hypothesis that the available workgroup computing literature and application documentation would provide sufficient information to develop a comprehensive workgroup computing taxonomy. By comparing workgroup tasks with workgroup computing functional areas, it was possible to derive a common set of workgroup computing management and support tasks that were based on the detailed workgroup computing taxonomy. Common workgroup computing management and support tasks formed the basis for an integrated workgroup computing architecture. Finally, 86 new API specifications were written for common workgroup computing management and support functions.

This study can be used by workgroup application developers to determine which common workgroup computing functions should be integrated into future workgroup applications. Implementing the results of this study in future workgroup computing systems will lead to flexible and integrated systems that are easier to use and more transparent to workgroup members. Workgroup computing researchers can use this study to identify workgroup computing functions that should be included in their research areas.

Table of Contents

List of Figures

Chapter I. Introduction

A. *Statement of the Problem*

A comprehensive workgroup computing architecture and programming interface specifications for common workgroup functions may be necessary to improve correlation between workgroup application support functions. These elements will also allow for more efficient control of group processes. This study analyzed workgroup computing applications to discover and understand their distinctive characteristics and fundamental structure to develop an in-depth taxonomy of workgroup computing applications and workgroup system services. A new common workgroup computing architecture can be developed based on building blocks or basic workgroup computing functions identified in this comprehensive workgroup computing taxonomy and the literature. Using both the taxonomy and architecture, workgroup computing application programming interface (API) specifications can be developed for common workgroup computing support functions.

B. *Issues*

Workgroup computing application areas were selected using an earlier study completed by the author (Von Worley, 1994) that identified workgroup computing areas and applications. Available literature and applications' documentation provided additional details of workgroup computing application functions and tasks. To develop an in-depth, comprehensive taxonomy of workgroup computing, information from multiple sources had to be reviewed, extracted, and compiled for each workgroup computing function and task. This exercise was tedious and time-consuming.

The second issue was to develop a revised workgroup computing architecture that included all of the basic workgroup computing functions and common application support features identified in the workgroup computing taxonomy. For completeness, the architecture had to include other system functions such as communications, user interfaces, and security. Several comprehensive architectures were assessed (Benford, Mariani, Navarro, Prink, & Rodden, 1993; Pastor & Jager, 1991; Reinhard, Schweitzer, & Volksen, 1994; and Sarin, Abbott, & McCarthy, 1991). A new workgroup computing architecture was constructed using elements from these existing architectures.

A third issue was to develop specifications for programming interfaces for the various workgroup computing primitives identified in the taxonomy and classified in the architecture. Consolidation of common workgroup functions reduced the final number of interface specifications to a manageable number.

C. *Topic Significance*

For workgroup computing to become more effective, individuals and groups must work together more efficiently and transparently using improved tools and integrated architectures. This work to produce a comprehensive workgroup computing taxonomy, architecture, and programming interface requirements is based on a detailed analysis of workgroup computing applications. Study results highlight the correlation and coordination between different workgroup computing applications. By defining common workgroup computing management and support functions, this thesis should also improve the efficiency and effectiveness of workgroup computing applications.

This study is important for the collaborative computing field because it provides a detailed workgroup computing taxonomy and integrated architecture. This structure will allow future workgroup computing research and product developers to define more integrated workgroup computing applications. The study also defines Application Programming Interface (API) requirements that may be implemented for common workgroup computing application areas.

D. *A Guide to the Rest of the Thesis*

The following discussion outlines the structure of the thesis and points to areas that may be of interest to the reader or researcher. The thesis contains much explanatory and descriptive information and an Annotated Bibliography that can point future researchers toward literature appropriate for their studies. The Annotated Bibliography contains 226 sources.

The Introduction chapter contains an explanation of workgroup computing, groupwork, a contrast of single versus multiple user applications, a comparison of several workgroup applications, and a discussion of current and future workgroup supporting technologies including multimedia. The Literature Review links the literature to the methodology and to the workgroup computing taxonomy, architecture and Application Programming Interfaces (APIs). This is followed with a discussion of Workgroup Computing Application Functions and Primitives. The study Methodology is outlined. A sampling of Workgroup Computing Taxonomy and API Specification Results and the Workgroup Computing Architecture can be found in the Results chapter followed by the Discussion, Implications, and Recommendations. Complete details of both the Workgroup Computing Taxonomy and the Workgroup Computing API Specifications are available in Appendices A and D. A matrix showing the relationship of workgroup computing taxonomy elements to workgroup computing functional areas can be found in Appendix C. The entire Workgroup Computing Taxonomy with annotations of each element is included as Appendix B. Recommendations for the Workgroup Computing Field and Recommendations for Future Research are also included.

E. *Background*

Workgroup computing background.

Computer Supported Cooperative Work (CSCW) is the field devoted to the study and theory of how people work together using computers (Greenberg, 1991). The field addresses how computers and similar technologies affect group behavior and what motivates and validates groupware design (Greenberg). CSCW also involves the development of software tools to enhance collaborative efforts (Ellis et al., 1991).

Computer Supported Cooperative Work (CSCW) was pioneered by Douglas Englebart who developed a prototype workgroup computing program in 1968 (Bikson & Eveland, 1990). The Massachusetts Institute of Technology held the first CSCW research workshop in 1984 (Greif, 1988). Major workgroup computing conferences followed in 1988, 1990, 1992, and 1994.

The term "Computer Supported Cooperative Work" was first used in 1988 by L. Bannon (Bannon, Bjorn-Anderson, & Due-Thomsen, 1988). CSCW can be described by the synonyms "groupwork", "collaborative computing", and "workgroup computing." For consistency, the term "workgroup computing" is used in this study.

The goal of workgroup computing is to help individuals and groups efficiently perform a wide range of functions on networked computer systems (Ellis et al., 1991). Early workgroup computing tools were designed for limited functionality and group interaction. Systems that addressed group interaction were often "closed, proprietary systems" (Craighill, 1992, p. 408) that did not integrate with the large number of existing information processing tools and applications. Current workgroup computing applications still do not provide enough control of group processes and they provide little correlation between various workgroup computing application areas. Rodden and Blair (1991) contend that "Existing approaches to control in distributed systems are inadequate given the rich patterns of cooperation found in CSCW [workgroup computing]" (p. 49).

The lack of integrated workgroup applications can cause additional work, work delays, and information loss in group-oriented activities. This problem gets worse as groups become geographically dispersed due to the problems of remote information sharing (Ellis et al., 1991). Effective workgroup computing requires that workgroup functions be integrated into a common environment. The following list shows basic functions required for workgroup support. Functions are: (a) activity management, (b) activity coordination, (c) report forms design, (d) report evaluation, (e) security and access control, (f) time management, (g) scheduling meetings, (h) resource allocation, (i) information storage, (j) communication, (k) information presentation, and (l) information processing (Pastor & Jager, 1991). Without a common support framework these functions must be programmed separately for each workgroup application (Pastor & Jager).

Groupware.

Groupware is the software that supports workgroup computing (Greenberg, 1991). Ellis, et al. define groupware to be "computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment." (p. 40). Groupware's goal is to help groups of users in communicating, collaborating and coordinating common tasks (Ellis et al., 1991). The groupware label differentiates workgroup computing applications from single user products. Groupware features include: (a) synchronous and asynchronous communication , (b) users in the same location, (c) users in different locations, (d) shared windows, (e) document management, (f) version control, (g) multiple comments on documents, (h) data security, (i) multimedia, (j) multiple tasks, (k) structured or semi-structured messages, (l) E-mail links, (m) conversation management, (n) idea organization, (o) negotiation and voting support, (p) group consensus, (q) group planning, (r) group scheduling, and (s) office procedure modelling (Ellis et al., 1991; Greif & Sarin, 1988; Huber, 1990; Power & Carminati, 1991; Sathi, Morton, & Roth, 1988).

Groupware applications include: (a) electronic mail, (b) group meeting support, (c) interactive video conferencing, (d) shared databases, (e) workflow managers, (f) decision support, (g) collaborative writing and editing, (h) drawing, (i) program management, and (j) group scheduling and calendaring products (Ellis et al., 1991; Greif & Sarin, 1988; Huber, 1990; Sathi et al., 1988). The term "groupware" is used in this study when referring to application software that supports workgroup computing.

Groupware growth.

The use of groupware has been steadily growing over the past several years. Figure 1 shows the rate of groupware applications revenue growth from 1992 to 1998. In 1993, groupware revenues were over six billion dollars and revenues are expected to increase tenfold by 1998 (R. Flanagan, personal communication, April 29, 1994). As the workgroup computing area continues to grow, it will become more important to develop more efficient, tightly integrated workgroup computing products that can span multiple work situations and allow better use of group resources (Rodden & Blair, 1991).

Groupware characteristics.

Groupware provides communication protocols and access controls that determine who in a group can use what information and in what manner. Groupware users must be working on a common task; however, access does not have to be simultaneous. Multiple-users

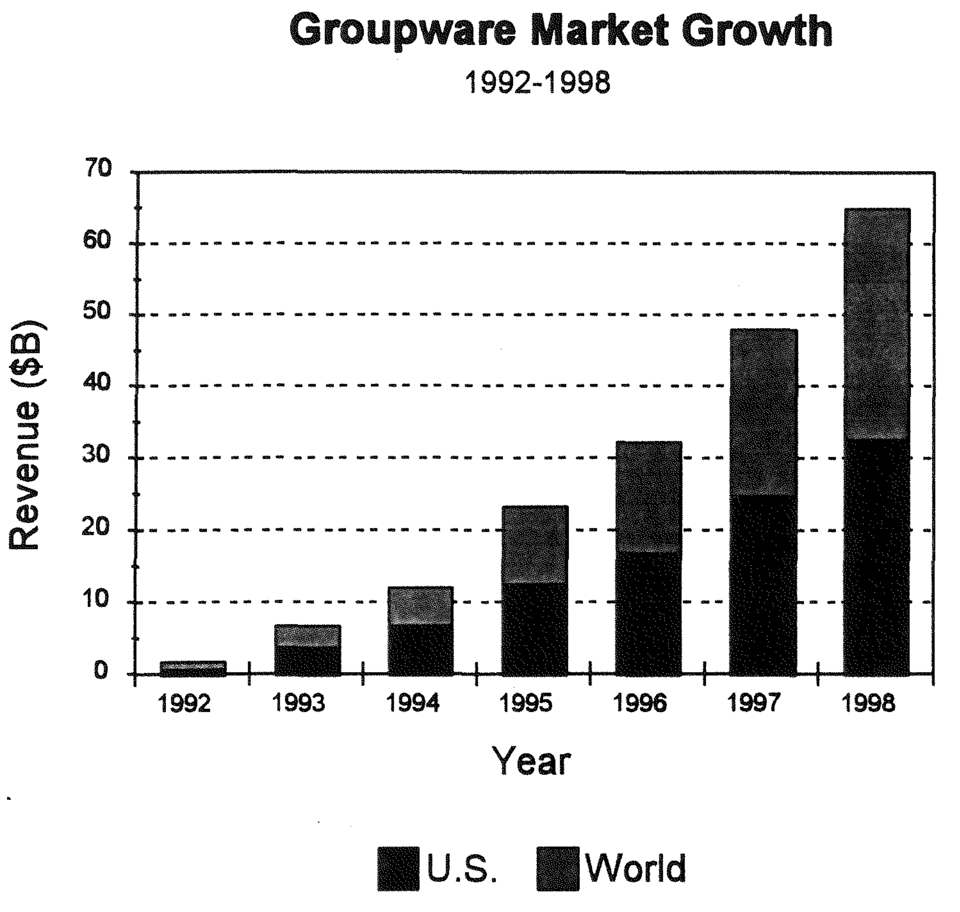

Figure 1. Groupware market growth from 1992-1998.

Note. Data are from R. Flanagan, WorkGroup Technologies, Inc. Copyright 1994 by WorkGroup Technologies Inc. Used with permission (R. Flanagan, personal communication, April 29, 1994).

working on a common task sets workgroup computing applications apart from single-user applications. Groupware excludes time-sharing systems or transaction processing applications where users do not share a common task (Ellis et al., 1991). Groupware deals with structured data and complex relationships between data, personnel, and schedules (Rodden, 1993). Workgroup computing users can be co-located or remotely located and interactions can take place simultaneously or at different times. This is known as synchronous or asynchronous access (Dix, Finlay, Abowd, & Beale, 1993).

Synchronous access requires more tightly integrated applications, and better control of workgroup processes (Dix, Finlay, Abowd, & Beale, 1993). Figure 2 illustrates this concept by showing the overlap of some workgroup applications in the various quadrants of the workgroup computing time/space matrix. This overlap indicates that groupware applications must share a common workgroup structure to avoid duplication of functions (Ellis et al., 1991).

Group process issues and dynamics.

To ensure that a group's functions are integrated, any study of workgroup computing or groupware should address the sensitivity of group applications to group process issues and dynamics. Important issues are defining the roles of group members, understanding how people work together, and the impact of group process gains and losses on group activity. Groupware systems should have many internal communication modes so that users can express themselves in varied and traditional ways. These modes include voice, visual, nonverbal, face-to-face, and text and graphics (McGrath & Hollingshead, 1993). There

	Same Time	**Different Time**
Same Place	Face-to-face interaction Meeting rooms	Asynchronous interaction Video conferencing
	Shared work surfaces and editors Shared personal computers and windows	
Different Place	Synchronous distributed interaction Argumentation tools	Asynchronous distributed interaction E-mail and electronic conferences
	Co-authoring systems and shared calendars	

Figure 2. Time/Space Matrix showing type of access and applicable workgroup computing applications.

Note. Available from two references. Used by permission from "Groupware: Some Issues and Experiences", *Communications of the ACM 34*(1) (p. 41) by C. A. Ellis, S. J. Gibbs, & G. L. Rein, 1991. New York: ACM Press. Copyright 1991 by the Association for Computing Machinery, Inc. Also used by permission from *Human-computer interaction* (p. 448) by A. Dix, J. Finlay, G. Abowd, & R. Beale, 1993. New York: Prentice-Hall. Copyright 1993 by Prentice Hall International (UK) Limited. Reproduced/reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

should be support for spatial separation of group members and the system should allow synchronous and asynchronous interaction (McGrath & Hollingshead). External group communication is also important; however, this subject has been given little attention in the literature (McGrath & Hollingshead).

There are four basic task performance processes that must be considered for groups. They are idea or plan generation, choosing an answer from several defined or undefined alternatives, resolving group conflicts, and executing group and individual tasks (McGrath & Hollingshead, 1993). To accomplish these processes, workgroups need to produce work, workgroup members require support, and there must be consideration for a group's well-being.

An underlying point in group process theory, discussed by McGrath and Hollingshead (1993), is that decision making becomes more difficult as it moves from the individual to the group setting due to "process losses." (p. 104) In a group environment, there can be a reluctance of individuals to participate in the group activity, inconsistent views concerning a problem, or dominance by a single individual. Strong or weak leaders also play a factor in group processes. In addition, some groups come to a premature tendency towards convergence on an issue or there can be an escalation of conflict within the group (McGrath & Hollingshead). A group process structure helps the group to focus on key issues while a task support structure provides necessary information and the ability to analyze it (Nunamaker, Dennis, Valacich, Vogel, & George, 1993).

Common process gains include additional information for the group to use for decision making, group stimulation and synergy, more objective evaluation of problems, and group learning (Nunamaker, Dennis et al., 1993). Process losses include blocking of ideas, lack of focus, conformance pressure, free-riding, single train of thought, domination, information overload, coordination problems, incomplete use of information, and incomplete task analysis. Process gains and losses can be moderated in workgroup systems by using group memory, allowing anonymity, using parallel communication channels, and increasing media speed (Nunamaker, Dennis et al.).

Group memory can be accomplished by recording workgroup sessions or allowing group members to pause and reflect on information and ideas of other group members. Smith (1994) introduced the concept of "collective intelligence" (p. 3) where groups work together over a period of time to produce a product that appears to come "from a single good mind." (p. 3) Anonymity allows a low-threat environment where issues can be discussed more freely. Parallel communication channels permit group members a broader input into a meeting or discussions with less chance of dominance by a single individual. Media speed allows the group to interact and transfer information more naturally without being limited by the speed of the computer system (Nunamaker, Dennis et al., 1993).

All groups must settle on a specific way of working that all members can commit to. The approach does not matter; however, it must be shared or group users will not be able to integrate their activities (Johansen, Sibbet, Benson, Martin, Mittman, & Saffo, 1991). By including users as part of the group, defining users' roles and status, and defining what an individual can get out of the group activity, groupware can help team members recognize their own processes (Johansen et al.). This can lead to expanded individual and group awareness and increased group output and efficiency (Johansen et al.).

Single-user versus multiple-user applications.

To properly identify common workgroup application functions, the differences between single-user and multiple-user applications should be compared. The following is a description of a workgroup application, shared drawing and graphical design, that will show how multiple-user workgroup applications differ from their single-user counterparts.

When using a shared drawing and graphical design application, a group of designers can work synchronously or asynchronously on a shared set of drawings. Each drawing has a lead designer and other designers who may be working together or alone on different sections of a drawing. If two or more designers attempt to work simultaneously on the same drawing section one designer will be granted access and an exclusive lock on that section of the drawing. Once a section of a drawing is locked, other designers cannot access this section of the drawing until the lock is released. Designers will be informed of the lock if they try to access that portion of the drawing. Lock release will occur after a designated period of inactivity or when the current lock holder is no longer working on the drawing. After a lock is released, any designer can request and be granted access to a drawing object (Greenberg, Roseman, Webster, & Bohnet, 1992).

Drawing changes are incorporated periodically or whenever a lock is released. This ensures that the most current drawing update is available for editing by a new designer.

Nonediting designers can view the drawing while it is being edited so that they can take turns editing in near real-time (Greenberg et al., 1992).

In contrast, the single-user drawing and graphical design application can be used by only one designer at a time. This individual is responsible for all drawing editing, coordination, updates, and transfer of the drawing to another designer, as required. If another designer wants to edit a drawing the designer must request a copy from the original designer and transfer the drawing to the correct computer. Two or more separate drawings with different revision status can exist causing confusion over which drawing version is the master. Drawing version control and update are two of important coordination problems with single user drawing applications (Lakin, 1988).

In a workgroup application, all tasks for setting up task configuration control, access control, work coordination and monitoring, and version control are handled by the groupware application (Greenberg et al., 1992). In single-user applications, the lead designer must manually perform workgroup integration functions handled automatically by the groupware application (Lakin, 1988).

A comparison of workgroup applications.

A review of commercial groupware applications can improve understanding of how well groupware application functions have been integrated. This section compares Lotus Notes 3.0, Windows for Workgroups 3.11, and LinkWorks Client for Microsoft Windows, Version 2.1.

Lotus Notes

Lotus Notes 3.0 integrates groups of networked users. It is also a tool for solving group workflow problems. Notes consists of three product components: a distributed document database, electronic mail, and forms generation (Chalstrom, 1993). Notes supports a variety of popular client/server platforms such as DOS, Microsoft Windows, Macintosh, Unix, OS/2, and Novell Netware (Chalstrom). Notes also provides cross-platform access to text, graphics, audio, and images. External documents, such as databases and spreadsheets, can be imported into Notes' documents. (Chalstrom).

The Notes' desktop consists of six workspace pages. The desktop is like a "six-drawer filing cabinet" (Schulman, 1994, p. 7) that contains all of a user's information. Each Notes' workspace page groups broadly related information into a drawer in the filing cabinet. A Notes' database is comparable to a file folder that contains information about a single topic. Documents are like sheets of paper in a file folder that contain information about a specific item (Schulman).

Databases form the main component of the Notes' system. A document in Notes is made up of fields related to each other. Information that can be stored in a Notes' document can be text, numbers, dates, times, lists, styled text, or graphics. Other documents generated outside the Notes' environment can be attached to a Notes' document. Multiple users can share all Notes' information (Gerwirtz, 1994). Views are designed to allow users to navigate through a database. A view is a way to structure database information categories so that users can access and sort documents (Schulman, 1994). Forms display data from Notes' databases on the screen or send data to a printer. The same data can be displayed in many different ways using forms. (Gerwirtz).

A Notes' mailbox is on each user's desktop. Procedures for reading mail and sending mail are the same ones used to prepare other Notes' documents. Users can select, read, print, file, copy, respond to, or delete incoming messages. They can create new messages that can be saved, sent, signed, or be encrypted. Carbon copies of messages are allowed and the delivery priority can be specified (Schulman, 1994).

Database replication is one of Notes' most valuable features. When a user updates or changes a database, the database modifications automatically occur in all copies of that database on the client/server system. Replication enables users at various locations to share database information (Gerwirtz, 1994).

Notes has a built-in encryption tool, the RSA public-key cryptosystem, that lets a user keep information confidential, give documents digital signatures, and provide data integrity features to control document tampering. Electronic messages (E-Mail), database documents, and individual files within fields can be encrypted for security. This feature allows intercompany networks and E-mail systems to be linked while providing security for company sensitive information (Schulman, 1994).

Lotus Notes has an excellent complement of integrated workgroup functions; however, it does not allow multi-user interactive editing of documents (Gerwirtz, 1994). Notes has limited meeting and conference support functions. It does not include computer-aided design functions and it has little workflow or decision support capability (Schulman, 1994).

Windows for Workgroups

Another product, Windows for Workgroups 3.11, is the network version of Windows 3.1. Program manager, file manager, and print manager functions are the same as those in Windows 3.1; however, workgroup computing applications such as networking, chat, mail, and a scheduler have been added (Borland, Lorenz, & O'Mara, 1993).

Windows for Workgroups allows a personal computer user to communicate and exchange files with other users on a network. A variety of network software operates with Windows for Workgroups. Any user with a compatible network, such as Banyan Vines or Novell Netware, can use Windows for Workgroups to access information on other linked networks. Program manager, file manager, and print manager access, investigate, and print information contained on the interfacing networks (Borland et al., 1993).

Windows for Workgroups' E-mail allows users to exchange files and messages with other users on the network. Users can generate and send mail, print messages, organize and store messages in files or folders, and search for existing messages. A variant of the mail function called "chat" will allow users to "ring up" (Borland et al., 1993, p. 191) a team member's computer, send a message, and receive a response in a window. Windows for Workgroups also has built-in fax software that allows users to send and receive faxes. Password security features ensure that only authorized users have read access to particular faxes. Windows for Workgroups does not have a public key cryptosystem like the one in Lotus Notes (Queen, 1993).

The schedule function keeps track of appointments and tasks. The function also records notes. Meetings can be set up with other users on the network. Schedule allows a group member to schedule and share appointments and tasks with other group members (Queen, 1993).

As discussed above, Windows for Workgroups provides basic workgroup computing functions. Like Notes, it does not allow multi-user interactive editing of documents. The application provides limited meeting or conference support, no computer-aided design functions or workflow management, and little decision support capability (Borland et al., 1993).

LinkWorks Client

Digital Equipment Corporation (DEC) has introduced groupware software called LinkWorks Client for Microsoft Windows, Version 2.1. LinkWorks runs under several different operating systems, including Windows 3.1. It provides information administration and management capabilities, document sharing, access control, E-mail, and automation of workflow activities (Krill, 1993). LinkWorks is suited to tightly integrated workgroups and it is different than other DEC groupware products like DEC PathWorks or TeamLinks. PathWorks is DEC's general-purpose PC integration product family suited for ad hoc, informal workgroups that exist in most organizations (Cini, 1993). TeamLinks is a limited function product that enhances team computing. TeamLinks provides electronic mail and document routing in client/server environments (Cini). Neither PathWorks nor TeamLinks will be discussed in this study.

DEC proposes to utilize LinkWorks as a "framework" (Garry, 1993) rather than an application such as Lotus Notes. LinkWorks' users can employ third party wordprocessors and spreadsheet applications at the user level and pass files, documents, and applications as objects on the network to allow workgroups to share this information (Krill, 1993). Future releases of the product will support links to shared network resources using as SCO Unix, Ultrix, Open VMS, Alpha AXP, HP-UX, and IBM AIX. For the user interface, LinkWorks will support Windows, OS/2, Macintosh, and Motif systems (Garry).

LinkWorks has several basic functions including creating, copying, editing, and deleting documents and filing containers. Attributes can be defined for each type of document or filing container so that a specific document can be retrieved using search criteria. A shredder tool can delete a document by placing the document file into an electronic wastebasket where it can be stored for later retrieval or destruction. LinkWorks imports or exports documents from and to other applications (Digital Equipment Corporation, 1994a).

Filing can be performed using a hierarchical container structure composed of cabinets, drawers, folders, and sections. Using several customizable filing levels, users create filing containers on desktops where files are grouped in containers based on type. In addition, a document can also be placed in a LinkWorks' archive where the document can only be retrieved by searching (Digital Equipment Corporation, 1994a).

Users can search for documents by looking in containers on their desktop, on another user's desktop in the same LinkWorks' domain, or in the LinkWorks' archive. The search will only find documents that users have permission to access. A user can specify the search criteria, including keywords, text strings, and deadlines in a workflow and the program can save search criteria for future use (Digital Equipment Corporation, 1994a).

Shared objects are those that appear at more than one location within the LinkWorks' domain. Documents or containers can be converted into shared objects by using mail, placing objects or files into a shared container, or by using the share option on the LinkWorks' menu. Shared documents will show an icon on each user's desktop. All sharing users can simultaneously access the shared object. The system places a lock on the document when it is opened for editing; however, other users can still see the document although they cannot access it. Users can also specify interest in a document so that they will be notified when it is being changed (Digital Equipment Corporation, 1994a).

Version control allows users to keep track of progress and keep various copies of documents that a group has worked on without renaming each version. A new version of a LinkWorks' document is automatically created when a user opens it. All document versions can be saved. In addition, document versions can be copied, edited, read, saved, or deleted (Digital Equipment Corporation, 1994a). For data security, users can set the access rights for each newly created document or filing container. Nine access levels range from public, with no restrictions, to private, which can only be viewed by its owner. An access right describes which actions, such as read, edit, or copy, can be performed by a particular level of user (Digital Equipment Corporation, 1994a).

LinkWorks has an internal mail function that handles messages sent between users in the same LinkWorks' domain. Users can also send LinkWorks' mail to external users or users in other LinkWorks' domains using the messaging protocol available on the network server. Mail can have attached documents and filing containers. Users can request mail delivery confirmation, suggest that a reply is necessary, or set a reply deadline. Users can re-route mail to another location, for any period, while they are absent from work (Digital Equipment Corporation, 1994a).

Workflow allows object routing, the setting of deadlines for different actions, and the signing of documents using password authenticated digital signatures. Any document or filing container can be routed and the flow can be defined by a workflow blueprint. A workflow blueprint defines a series of stages through which an object passes. A stage can be serial or parallel which automatically makes the object a shared object. When users have completed a workflow step they drop the object into the outgoing mail slot on their desk. LinkWorks then sends the object to the next user or organization specified in the workflow blueprint (Digital Equipment Corporation, 1994a).

LinkWorks' documents can be signed at three approval levels. Documents can be initialed as "seen", they can be "signed for approval", or they can be "signed off." A "signed off" document can no longer be modified or deleted (Digital Equipment Corporation, 1994a).

If a document or filing container requires further action it can be placed in a pending box. A deadline and comment must be attached. LinkWorks displays a notification message and comment if a deadline expires (Digital Equipment Corporation, 1994a).

No reviewed application integrates all workgroup computing functions into a single product. The previous applications contain a limited core of workgroup computing functions, including file and document management, E-mail, meeting support, writing and editing, and rudimentary workflow management. Existing application components are integrated; however, important workgroup components are missing.

In the reviewed group, DEC LinkWorks is the most comprehensive workgroup computing product with the best integration of functions. LinkWorks has more workflow management capability than Lotus Notes and Windows for Workgroups. It has multi-user document writing and editing capability, but no computer-aided drafting, drawing, project management, or decision support functions.

Workgroup computing supporting technology.

Workgroup computing systems require more than groupware applications. To make a complete system, there must be adequate support services including networking, access control and security, and a user interface (Schnaidt, 1992). The following describes the current capability in this area.

Computer networks that support workgroup computing are categorized as either local-area networks (LANs) or wide-area networks (WANs). Either type of network is defined as a set of computing devices or nodes that can communicate with one another over a set of communication channels. A LAN connects nodes in a local area and a WAN connects nodes in widely dispersed geographical areas. LANs and WANs provide the capability to share valuable computer and application resources between computer users. LANs and

WANs have allowed workgroup applications to be developed that could not exist in a one-person, single-computer environment (Schnaidt, 1992; Stallings & Van Slyke, 1994).

LAN or WAN nodes can be computers, terminals, or workstations. Network communication channels use either copper-based, optical, or wireless connection media. Copper-based wiring is categorized into voice or data grade twisted pair, shielded twisted pair, or coaxial cable. Optical fibers are thin, flexible glass or plastic fibers that transmit light. Optical fiber-based systems are more difficult to install than copper-based systems; however, optical fiber use is expanding since optical systems can support higher data transmission rates. Wireless systems are also being installed more frequently. Transceiver installation is easy and wireless systems can connect remote nodes without installation of copper or optical cabling. Advantages of wireless systems are currently offset by their limited frequency and channel capacity (Schnaidt, 1992; Stallings & Van Slyke, 1994).

Data transfer between the nodes is controlled by the access method embedded in the network interface card. The two most popular methods are token-ring and Ethernet (Stallings & Van Slyke, 1994). In a token-ring structure an electronic token travels around a ring of nodes in the form of a header. A sender captures this header and adds a message resulting in a message frame that travels to the receiver's node where it is removed and a new token circulated (Stallings & Van Slyke, 1994). Ethernet is based on a bus network topology. Ethernet uses a contention line-control protocol. When a message is sent, the node gains control of the line, sends the message, then relinquishes control of the line to another node (Stallings & Van Slyke, 1994).

Communication servers provide communication links between compatible LANs and WANs. Because networks use a variety of communication protocols and operating systems, incompatible networks cannot communicate directly with one another. Routers bridge the gap between LANs and WANs by performing the necessary protocol conversion and routing the message or data to its destination (Schnaidt, 1992; Stallings & Van Slyke, 1994).

The workgroup network operating system may consist of Netware, UNIX, Windows for Workgroups, IBM's LAN Manager, Banyan Vines, or some other LAN manager. There may also be connections to mainframes and mini-computers (Schnaidt, 1992).

The user interface may be MS-DOS, Microsoft Windows, Windows NT, OS/2, UNIX, NeXTstep or Macintosh software. Many operating systems and interfaces are unique to particular types of machines and there is not interoperability between many platforms. Several new operating system (OS) products including Windows 95, previously known as Chicago, will allow applications to be integrated more easily into the user environment. In addition, Windows 95 provides 32-bit programming support and better networking capability. Windows 95 will include some public application program interfaces (APIs) for functions such as file management, browse, open, and print features. This will help groupware developers access these file management functions without having to rewrite specific interface code for each workgroup computing application (Johnston, 1993).

Workgroup computing and multimedia.

Many future advances in groupware will be in multimedia and workflow applications (Marshak, 1992; Rangan, 1992). Improvements in workflow applications will allow for more efficient management of documents, processes, and information flow in an organization. Multimedia applications will create a need to pass a higher volume of information over networks (Marshak).

According to Rangan (1992), multimedia is a union of voice, text, images, and video data for enhanced output. In the workgroup environment, multimedia combines traditional text-based data processing with live or recorded video displays, voice and sound, computer-generated images, and animations to produce a rich environment for implementing groupware applications (Opper & Fersko-Weiss, 1992). High bandwidth communication network installations such as Broadband Integrated Services Digital Network (B-ISDN) and Switched Multimegabit Data Service (SMDS) are paving the way for digital multimedia for workgroup computing (Stallings & Van Slyke, 1994). Multimedia systems provide any combination of synchronous or asynchronous access (Rangan, 1992).

Storage of megabyte and gigabyte data files for multimedia applications will stress the capability of existing magnetic media data storage and transfer devices. Multimedia storage systems for workgroups must be capable of quickly storing gigabytes of information and transporting it without data loss (Khoshafian, Baker, Abnous, & Shepherd, 1992; Little & Venkatesh, 1994). CD-ROM, WORM, and writable and erasable optical disk drives can be used to meet the higher storage demands of multimedia applications (Little &

Venkatesh). Optical storage systems also provide a 20 to 100 year archival life for data versus 1 to 3 years for magnetic media. This is important for archiving workgroup activities and products in the corporate environment (Koshafian, et al., 1992).

Compression software and hardware are necessary to speed the display and improve the storage efficiency of imagery data (Koshafian et al., 1992; Stallings & Van Slyke, 1994). Until recently, only lossless video compression/decompression schemes have been used. These lossless techniques do not degrade data during the compression or decompression process; however, they are only capable of modest 2:1 to 8:1 compression ratios. Some of the newer lossy compression techniques, which use fractals, do not offer perfect reproduction, but they can have compression ratios of 100:1. For multimedia images and sounds, the recovered decompressed information using lossy techniques is adequate for reconstruction of the source material (Weiss & Schremp, 1993). Bell Atlantic is using lossy compression techniques to transmit full motion video over conventional phone circuits in Virginia (Ganssle, 1994). Compression will also allow synchronous transmission of multiple data streams over existing lower bandwidth communication networks (Rangan, 1992; Stallings & Van Slyke, 1994).

Multimedia imaging systems require high-resolution monitors to display letter-size images without zooming (Little & Venkatesh, 1994). Users can access multiple windows that contain video images, pictures, graphics, or text. Video input, high resolution scanners, optical character recognition (OCR) software, facsimile capability, sound boards with voice software and stereo speakers, and multi-gigabyte storage devices will be required for full multimedia implementations (Koshafian, et al., 1992; Little & Venkatesh, 1994).

Multimedia experiments that combine voice and data with video will revolutionize desktop computer communications (Brittan, 1992). With the introduction of multimedia into the workgroup environment, the workgroup interface should become more like traditional face-to-face group meetings or work situations. Individuals using workgroup products may feel more comfortable with multimedia systems that allow more traditional and natural forms of communication. Multimedia systems will allow voice, sound, video, graphics, and text to be tailored and integrated to meet any work situation or user desire (Koshafian et al., 1992; Little & Venkatesh, 1994). For example, an individual in a workgroup can transmit a video or sound clip that is relevant to the workgroup task while explaining it using the voice link. Documents can be edited or transferred for coordination and approval. When using a video link during a workgroup session, a workgroup member can choose an icon to replace the video picture if they do not desire to be seen on another user's computer screen (Koshafian et al., 1992).

Future technology needs.

As multimedia workgroup applications emerge, they will produce the need for increased data transfer rates due to the need to transfer high resolution graphics and video images. In addition, distributed workgroup architectures with multiple LANs, hundreds of nodes, and multiple sites will require an enhanced network capability. New developments in high-speed network services include frame relay, Switched Multimegabit Data Service

(SMDS) and Asynchronous Transfer Mode (ATM). B-ISDN is another high-speed networking technology that evolved from conventional ISDN (Heldman, 1993; Stallings & Van Slyke, 1994). All of these network services can meet future high-rate data transfer requirements.

Frame relay provides a more efficient approach to wide-area packet switching. This new standard that emerged from ISDN will provide extended network performance for small to medium-sized networks with a minimum of system overhead (Stallings & Van Slyke, 1994).

To accommodate higher data transfer rates and larger networks, ATM, known as cell relay, will be implemented for metropolitan workgroup LANs and WANs. Similar in concept to frame relay, ATM provides faster packet switching that can support data rates several orders of magnitude higher than frame relay (Stallings & Van Slyke, 1994).

SMDS is a high-speed switched data service developed by Bellcore (Stallings & Van Slyke, 1994). SMDS can provide higher data rates required for multimedia workgroup applications (Stallings & Van Slyke). SMDS is a metropolitan network that supports LAN and WAN interconnections, image transfer, and bulk file transfers (Heldman, 1993). SMDS uses streamlined processing to achieve these high data rates (Stallings & Van Slyke).

Another approach to high-speed networking combines B-ISDN and ATM (Stallings & Van Slyke, 1994). B-ISDN uses a Synchronous Optical Network (SONET) that will allow full-motion video for desktop conferencing (Messmer, 1993). Data transport mechanisms such as ATM will allow multiple logical connections to be multiplexed over a single interface. ATM is more streamlined than frame relay or SMDS (Stallings & Van Slyke, 1994). ATM can provide workgroups with a data capacity in the hundreds of megabytes using packets, also known as cells, to transmit data in 53 eight-bit bytes over a B-ISDN network (Schnaidt, 1992).

Another network structure for high-speed information services will be an optical fiber-based voice/data/imaging system. This system will provide high-speed data transfer rates that can handle all combinations of voice, video, graphics, and text (Heldman, 1993). These optical fiber-based systems will support multimedia workgroup computing by covering a large geographical area with a high bandwidth networking system (Schnaidt, 1992; Stallings & Van Slyke, 1994).

Some type of encryption technology will have to be implemented to ensure information confidentiality (Schnaidt, 1992). Primary candidates are the Data Encryption Standard (DES) or the RSA cryptosystems (Russell & Gangemi, 1991). Both systems allow various levels of security depending on user requirements. The DES has been embedded in many commercial products and is the encryption algorithm of choice for many commercial users. DES uses a private-key encryption algorithm and RSA is a public-key system. In a private-key system the same secret key is used to encrypt and decrypt information. The security of this algorithm depends on how well the secret key is protected. To strengthen encryption, there are variations of the DES that use longer key lengths and multiple encryption levels (Russell & Gangemi).

The RSA algorithm uses two keys: a public key and a private key. A user must keep the private key secret, however, the public key does not have to be kept secret. Most versions of the RSA algorithm use 40, 154, or 512-bit keys. Use of keys with large numbers makes it extremely unlikely that factoring can be used to break the RSA security. The primary advantage of public-key systems is increased security. The secret private keys do not need to be transmitted or revealed to anyone. In contrast, in a private-key system there is always the chance that an unauthorized individual could discover the secret key while it is being transmitted (Russell & Gangemi, 1991).

An interesting capability of public-key cryptosystems is the ability to easily use a digital signature for message authentication (Russell & Gangemi, 1991). This allows verification of various transactions on the network where it is important to verify the source of a message. When used for digital signature purposes, authentication allows use of a key to verify origin of a message and identity of a sender. To "sign" a message, the sender encrypts the message using their secret key. The recipient receives the encrypted message and the advice that it came from the sender (Russell & Gangemi). The recipient looks up the sender's public key and uses it to decrypt the message. If the decryption is successful, the recipient knows that the message came from the sender because only the sender has the secret key that matches the public one. Digital signatures are also available with a private-key systems; however, this requires sharing of a secret and sometimes trust of a third party (Russell & Gangemi). A sender could repudiate a previously signed message by claiming that the shared secret was compromised by one of the parties sharing the secret (Russell & Gangemi).

To support integrated groupware applications, hardware will evolve to faster processors including the Intel Pentium, Motorola Power PC, SPARC, and DEC Alpha chips. As the time required to make decisions continues to shorten, future systems will distribute real-time solutions using multiprocessing techniques. Continuing issues will be the speed of communication between processors and the synchronization of tasks between different users (Laplante, 1992).

Despite efforts to build better software, most software today is still built one instruction at a time using high-order programming languages (Booch, 1994). A better way to construct programs and groupware may be object-oriented technology (Taylor, 1990). According to Booch, the object-oriented methods have evolved to help developers "exploit the expressive power of object-based and object-oriented programming languages, using the class and object as basic building blocks" (p. 34).

The three main areas of object-oriented technology are objects, methods, and classes (Taylor, 1990). An object is a software package that "contains a collection of related procedures and data" (Taylor, p. 16). Procedures are called methods. Objects interact with one another by sending messages asking an object to carry out a procedure. A class defines the methods and variables included with a particular object (Booch, 1994; Taylor).

Any object that contains a particular value for variables belongs to an instance class. Classes can be nested to any degree. The resulting structure is called a hierarchy. Classes can also be defined in terms of one another. This is called inheritance (Booch, 1994; Taylor, 1990).

Object-oriented workgroup computing applications and support systems and improved graphical environments and object-oriented operating systems like those being developed by Taligent will improve how users can display and use workgroup information. Taligent will be the first completely object-oriented operating systems (OS), providing developers with preprogrammed collections of code objects called frameworks (Cortese, 1993). Frameworks allow developers to use many preassembled object sets. Rather than programming individual function sequences, programmers use frameworks' object sets to dictate object interactions and modify defined functions, subroutines, or complete programs (Pederson, 1994).

Taligent's modular architecture allows essential operating structures and interfaces to be segregated into the core operating system kernel. Taligent also allows mulitprocessing and multiprogramming on concurrent operating systems. This will make it easier to upgrade to a new operating environment or combine an old and new operating environment. Taligent's structure permits faster data flow to workgroup users by locating other system drivers or workgroup APIs close to the core kernel. This will increase a workgroups's ability to coordinate their activities (Cortese, 1993).

F. *Assumptions*

This study used the following assumptions:

1. Results of this study can be applied to workgroup computing systems that run under Unix, OS/2, or other windows-based operating systems.

2. API specifications developed using the Windows 3.1 format can be rewritten for Windows 95 (Chicago) or other windows-based operating systems.

G. *Limitations*

The following are the limitations of this study:

1. This study uses only the Microsoft Windows 3.1 syntax for developed API specifications. Resulting API specifications can be converted to the syntax for other operating systems discussed in this thesis.

2. The detailed taxonomy addresses only workgroup application areas included in the technical literature or available in commercial products.

3. Detailed programming is not provided for the API specifications.

4. This study does not evaluate the API specifications.

H. *What Was Accomplished in this Study*

1. Based on a detailed analysis of workgroup computing applications, a comprehensive workgroup computing taxonomy was derived for specified workgroup application areas. Sufficient information was available in the reference sources to develop a section of

a workgroup computing taxonomy for each of 11 workgroup computing functional areas.

The analysis did not reveal any added workgroup computing functional areas and no areas had to be combined or divided. Each taxonomy section includes all the tasks and primitives that could be extracted from the available reference sources.

The analysis also confirmed that no single groupware product includes all workgroup computing functions. Each application normally includes its own set of relevant functions that allow the application to perform a necessary activity such as meeting or conference support, scheduling, or project management. The taxonomy allowed identification of common workgroup tasks and primitives that may be incorporated into integrated applications. Repetitive or missing workgroup tasks and primitives were identified by creating a matrix showing workgroup tasks and primitives versus workgroup computing functional areas.

2. Using the detailed taxonomy, an integrated workgroup computing architecture was developed based on common workgroup management and support functions. By comparing workgroup tasks with workgroup computing functional areas, it was possible to derive a common set of workgroup computing management and support tasks that were based on the comprehensive workgroup computing taxonomy. These common workgroup management and support tasks formed the basis for an integrated workgroup computing architecture. The developed workgroup computing architecture consists of four main elements: workgroup applications, workgroup support interfaces, workgroup management interfaces, and the operating system and network interfaces. The architecture provides a common

style of interface for all workgroup activities that permits a user to access shared functions. A user can adapt each workgroup session to a specific set of workgroup styles. Since all workgroup functions are defined and accessible by all users, complex, multiple-session workgroups can be supported.

3. In addition, it was possible to write new workgroup computing API function specifications for common workgroup computing management and support tasks. API specifications were written for 86 workgroup computing APIs that will allow the integrated architecture to be implemented in a Windows-like operating environment. Based on the sorting of workgroup computing tasks and primitives, the API specifications were divided into two groups: workgroup management APIs and workgroup support APIs. Developed API specifications support the defined architecture and they allow maximum flexibility for configuring workgroup sessions.

Chapter II. Literature Review

A. *Introduction*

The literature review includes articles from four workgroup computing areas: background, applications and groupware, taxonomies, and architectures. Much of this material will be used to develop the workgroup computing taxonomy and architecture. Application Programming Interface (API) references provide information to help develop API specifications for workgroup computing tasks and services. Background information concerning researchers and authors was current as of the time they published their articles.

B. *Workgroup Computing Background*

Computer supported cooperative work and workgroup computing.

The material in this first set of sources provided background information about Computer Supported Cooperative Work (CSCW) and workgroup computing. These sources addressed a definition of CSCW, identified core workgroup computing issues, discussed methods for sharing information space, and showed how to adapt workgroup technology to the organization.

Bannon and Schmidt (1991), two workgroup computing researchers from Denmark, examined some of the core issues that have helped to define workgroup computing. These core issues included a definition of cooperative work, sharing information spaces, and adapting workgroup computing technology both to the organization and to the users. The authors concluded that CSCW is a valid representation of the research area and that there is much work to be done in defining the issues. Bannon and Schmidt stated that "We need to develop a theoretical framework that will help us understand the complex interactions between the technological subsystem, the work organization, and the requirement of the task environment" (p. 15).

Greenberg (1991) and Hsu and Lockwood (1993, March) discussed collaborative computing environments that allow people to share information. Greenberg, who was with the Department of Computer Science at the University of Calgary, has edited a book based on CSCW and Groupware papers published on the Journal of Man-Machine Studies in 1991. Greenberg defined CSCW as "the scientific discipline that motivates and validates groupware" (p. 1). Greenberg noted that groupware is neither well defined nor is there consensus on what applications should be included in the field. Hsu and Lockwood discussed three fundamental aspects of collaborative computing systems: common tasks, shared environment, and time and space sharing. Both Greenberg and Hsu and Lockwood emphasized that implementation of workgroup computing systems is difficult both from technical and social aspects of an organization. These systems can meet resistance when they are implemented because they challenge the existing organizational structure. Organizations must change the way they operate for collaborative systems to be successful.

Required changes are not easily defined. Hsu was a computer consultant and Professor of Information Systems at Montclair State College, New Jersey, and Lockwood was a technical editor with *Byte* magazine.

Two CSCW approaches exist. The first approach supports the exchange of information between users and the second develops systems that allows the cooperative sharing of information. Rodden (1993), from Lancaster University, United Kingdom, reviewed technologies exploited by each approach. Rodden also discussed electronic meeting and conference systems that allowed information sharing and group communication. Rodden provided details on shared and private views, displaying related information, on-line voting schemes, participant autonomy, conference roles, and a conference command system. Rodden contended that the key to successful systems is the manipulation of text and graphical images to support writing and argumentation.

Important features of workgroup computing systems include access, version control of documents, commenting capability, confidentiality of personnel, and security. Power and Carminati (1992) worked on implementing a European groupwork project called PECOS (PErspectives on Cooperative Systems). They provided a history of CSCW and key groupware features. The authors specified that CSCW systems are required to support collaboration in complex situations. Power and Carminati stated that "Most existing CSCW tools have been developed in order to try out a new technological idea such as semi-structured messages or of graphs that represent possible speech-act sequences" (p. 24). The tool was developed first and then the possible uses were identified. Integration is necessary. Basic workgroup computing functions that are common to many activities should not be duplicated for each application tool; however, shared functions should be developed. This type of environment allowed faster development of new workgroup computing tools and easier integration with existing tools.

Rodden and Blair (1991) researched the problem of control in distributed systems by investigating forms of cooperation and the geographical diversity of the participants. Although extensive patterns of cooperation are found in CSCW, the authors contended that existing methods of control are inadequate. Rodden and Blair saw the need for high-speed networks that could provide the delivery of multimedia information to multiple users.

Bowers and Rodden (1993) studied different views of the computer interface in the context of workgroups. Traditional concepts of the interface were modified to provide richer conceptualizations of the relationships between users and computer systems. These alternative concepts were examined within the framework of actual organizations to determine applicability and usefulness. This thesis will use this work to help define workgroup computing interfaces in various workgroup settings.

Poole and DeSanctis (Wagner, Wynne, & Mennecke, 1993), researchers at the University of Minnesota, theorize that group support technology is not an object that is adopted equally by every group. Known as the Adaptive Structurization Theory or Give-N-Take, the main idea of the theory is that the group support structure should not be imposed on the group. Instead, the group should be exposed to the workgroup technology and they should be allowed to selectively adapt it to their individual group structure (Wagner et al.). From the beginning, SAMM, a University of Minnesota group support system was designed to be fun and provide friendly support rather than impose an arbitrary structure on the group. The SAMM system was designed to give each group member full access to all aspects of the workgroup software rather than giving control to a group leader, facilitator, or single individual (Wagner et al.).

Group dynamics issues and organizational impacts have been explored during the development of systems for organizations. According to Grudin (1991a), these issues have not been adequately researched for small groups. Grudin (1991a), a computer science faculty member at the University of California, Irvine, contended that "in order to effectively integrate these experiences, interests, and approaches [into small workgroups]. . .we have to go beyond what is shared and explore the differences." (p. 91). As these development areas converge, they are creating a common language that will allow developers to gauge their position in a rapidly changing systems world. Grudin (1991a) argued that the concepts of CSCW and groupware lack definition. Although the field has a strong technology component, workgroup computing has mainly been concerned with the behavior of people and organizations.

Groupware.

Groupware, the software component of workgroup computing, is defined and discussed in this section. Selections include writings, studies, and interviews with various organizations that determine the basis for groupwork definitions.

This article by Ellis, Gibbs, and Rein (1991), who were researchers in the groupware area, provided a broad overview of groupware. According to Ellis et al., groupware is a merging of computers, large information databases, and communication technology. Ellis et al. considered groupware to be "computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment" (p. 40). This team of researchers, from University of Texas, the Microelectronics and Computer Technology Corporation, and the University of Geneva, looked at several groupware design issues that included the ability to communicate, collaborate, and coordinate using selected software packages. In the area of communication, Ellis et al. saw the goal of groupware as an interface to a shared environment that could help groups to communicate and perform other comparable functions. Ellis et al. also provided information on a time/space workgroup computing taxonomy that will be discussed later in this section.

Bullen and Bennett (1991) interviewed 223 people in 25 organizations to determine how groupware systems were being utilized. They defined groupware as "computer-based tools that can be used by workgroups to facilitate the exchange and sharing of information" (p. 71). Their research attempted to understand the value that groupware brings to the office environment. Based on this survey, Bullen and Bennett found that implementation challenges in all surveyed organizations were the same and that problems of coordination and computerization were solved using similar approaches. They concluded that "experiences gained from studying people as they use new tools can benefit the designers of the next tool generation, thereby helping to accelerate the process of acceptance and use of these tools" (p. 81).

Dix, Finlay, Abowd, and Beale (1993), human-computer interaction (HCI) researchers from the United Kingdom, provided a time/space framework for groupware. They classified groupware into computer-mediated communications that supported direct communication between participants, meeting and decision support systems that captured common understanding, and shared applications that supported user interaction with shared work. The authors' groupware time/space matrix showed various categories of workgroup applications spread across a spectrum of asynchronous to synchronous activity. Their matrix provided insight into the structure of groupwork applications that was discussed in the introduction to this study. Dix and Finlay were from the University of York, Abowd was Carnegie-Mellon University, and Beale was from the University of Birmingham, United Kingdom.

Using a Macintosh computer system, Dale (1993) studied views of groupware compared to real experiences. Dale, the Advanced Technology Manager for Strategic Programs at the Government Center for Information Systems in the United Kingdom, discussed both technical issues and human-computer interaction. The paper concluded with benefits and problems of building this type of system. Based on these studies, Dale defined groupware as "the networked hardware and software which allows people to support each other in their efforts to achieve their work goals irrespective of when or where they might want to do this" (p. 317). Dale proposed that groupware can increase communication between individuals and groups while changing organizational hierarchies. Adopting groupware produced more frequent and easier communications. This can lead to better teamwork. Groupware can also have a positive effect on the organizational structure. These effects included reduced management levels, simplified control, and workgroups being created and dissolved in response to changing organizational requirements and goals.

Opper and Fersko-Weiss (1992), an independent organizational environment consultant and a computer industry editor and writer, respectively, defined groupware as "any information system designed to enable groups to work together electronically" (p. 4). Groupware ingredients must include electronic communication, information management procedures, and group administration. Opper and Fersko-Weiss defined groupware categories as administration, information management, communication management, and real-time meeting management and control.

Administration included meeting scheduling, group calendars, name and address files, and simple document control. Information management included document editing, project tracking, and filtering systems. Communication management comprises computer conferencing and bulletin boards. In the last category, Opper and Fersko-Weiss (1992) saw a need for real-time multimedia meeting support that included consensus building and voting systems.

In a recent article, Grudin (1994), highlighted the differences between groupware and traditional computer support. Grudin outlined the origins of groupware, discussed eight challenges for groupware developers and examined groupware successes. Conclusions were that groupware has targeted smaller system problems rather than problems affecting larger organizational goals. Groupware has been primarily off-the-shelf and it has not been integrated. Grudin contended that groupware will be more successful if it is integrated with existing features that support individual computer activity. For example, co-authoring documents required most work to be completed alone. Individuals do not want to give up their favorite word processor to use a co-authoring application. Rather than developing new groupwork applications, groupwork functions should be integrated into familiar individual user applications.

Like Dix et al. (1993) and Opper and Fersko-Weiss (1992), Allison (1992), with HaL Computer Systems, defined groupware to be "the software that facilitates group efforts" (p. 231). Groupware can be used for decision making or the collaborative development of large systems. Allison stressed that groupware must be tailored to the needs of the organization using it. Allison also discussed underlying support systems including networks, databases, and file transfer. The author called this software "glueware" (p. 232) when it was used to hold together a number of different applications.

The term "groupware" has been used to describe various software products from simple electronic mail programs to complicated workflow automation software or complex computer-aided design programs (Higgins, 1992). Many experts, including Higgins from Enable Software, did not think that groupware should be considered a distinct category of software. Higgins explained that industry analysts recognize applications that deserve to fall under the groupware label. Higgins discussed six categories of groupware: messaging systems, database-enabled messaging applications, message-filtering applications, workgroup application development software, calendaring and scheduling products, and document management and imaging systems.

Messaging systems allowed E-mail exchanges between users on a LAN or WAN. Most available groupware applications included E-mail. Database-enabled messaging applications let users organize, distribute, and file messages. Message-filtering applications permitted users to specify criteria for filtering messages based on content, subject, or source. Workgroup application development software allowed users to build workflow management and automation applications. Calendaring and scheduling products allowed users to schedule meetings and use shared resources. Lastly, document management and imaging systems permitted routing, retrieval, and filing of text or images (Higgins, 1992).

Robinson (1991) of Sage Force Limited, United Kingdom, illustrated the content of the groupware field by reviewing first generation workgroup computing applications. Robinson discussed group authoring, calendar management, meeting scheduling, action coordination in organizations, and large meeting support.

The Group Outine Viewing Editor, GROVE, is a group authoring tool that was developed at the Microelectronics and Computer Technology Corporation. GROVE included an outliner and a real-time group editor. Co-authors can remain in their offices while using a computer link to access the system. A voice link was also provided. This early group authoring system had no method to avoid conflict between authors on the same piece of text. Despite its limitations, the GROVE system enjoyed user acceptance because of its simplicity (Robinson, 1991).

Calendar management and meeting scheduling are groupware functions that have not enjoyed success because there has been a disparity between those who enjoy the benefits and those who must do the work. To successfully schedule meetings or other events, a group calendar must be maintained by all users who may participate in an organization's meetings or activities and personnel must agree to allow the computer to schedule their free time. Studies have shown that neither of these conditions are normally satisfied. Managers have secretaries that can maintain their calendars. In contrast, non-management personnel must update their own schedules; however, they rarely do. A commitment is required by all personnel to make this type of system work (Robinson, 1991).

The Coordinator is one of the more recognized workgroup E-mail applications. The Coordinator was designed to allow exchange, clarification, and negotiation of commitment in an organization. By using cooperative work techniques, there was intent to improve democracy and respect for people in an organization. Hancock of EDS Corporation reported a favorable experience with the Coordinator (Robinson, 1991); however, after a six month study, Grantham and Carasik (Robinson) agreed that the Coordinator has not been a successful groupwork application. Although the system supported the process of communication, the Coordinator failed to provide adequate support for conversation or work negotiation processes in an organization (Robinson).

Large meeting support was pioneered by the University of Arizona Management Information Systems Department. This group developed a Group Decision Support System (GDSS) than included a "Meeting Environment" (p. 66). The system supported a wide variety of group meeting functions including voting, issue analysis, and brainstorming. Studies showed that GDSS meeting support was effective for medium to large size groups. (Robinson, 1991).

Many of these early applications that were designed for small groups had limited results. Robinson (1991) saw a need for fewer small group systems and more systems that can aid large groups. Large meeting support included in the University of Arizona's GDSS is one of the most successful examples of an early groupwork application (Robinson).

Another area of workgroup computing that is becoming important is multimedia support. Brittan (1992), an associate editor of *MIT Technology Review,* reviewed multimedia experiments that successfully combined voice and data with video. Brittan described several experimental systems that provided elements of this advanced communications capability. Multimedia will revolutionize the way information is presented in workgroup computing settings; however, as discussed in the introduction, higher bandwidth communication networks will be required to support multimedia technology.

Rangan (1992) discussed the higher bandwidth communication systems that are required for multimedia communications. Implementation of these systems will result in a large number of multimedia collaborative applications. In addition, there is a need to develop medium transmission protocols that will allow synchronous transmission of multiple data streams. Rangan is a researcher with the University of California, San Diego, Computer Science Department, Multimedia Laboratory.

Current research in high-speed networking and multimedia addresses how multiple data streams can be synchronized and transmitted to support multimedia groupware applications. Yavatkar and Lakshman (1994) developed a protocol suite called multiflow conversation protocol that addresses issues of concurrency control and temporal and causal synchronization of messages and other network traffic. Concurrency control provides for consistent information views and smooth transitions among workgroup computing users. Temporal synchronization allows the related data streams to be moved through the network as a group and be delivered in the correct order to the destination workgroup user. Causal synchronization preserves the context in which the message is sent. Messages must be received in an order that makes sense to the receiver. The goal of this research is to integrate these areas into a UNIX system to build multimedia groupware applications. Both researchers were with the Department of Computer Science, Kentucky University.

Another current research area is the video display of workgroup collaborations. Gershman and Sato (1994) discussed the video wall that will provide both a multimedia work channel and a multimedia discussion channel. Both channels would require wide bandwidth network transmission systems. The researchers desire to make workgroup interactions ubiquitous. The goal is to make these systems easy to operate, equally accessible to all participants, and allow rapid and efficient information transfer. Gershman and Sato were with the Human Systems Integration Laboratory, Anderson Consulting.

C. *Workgroup Computing Applications*

The literature review identified 11 workgroup application areas and 51 applications or journal articles where there was sufficient detail to allow extraction of workgroup tasks and primitives. A review of application manuals for leading workgroup computing provided detailed information on how these products function in a workgroup computing environment. These groupware applications will provide sources for details of workgroup computing functions, tasks, and primitives.

Computer-aided design, drafting, and rendering.

Glickman and Kumar (1993), from Enterprise Integration Technologies, collaborated with Stanford University on a groupware mechanical engineering project called SHARE. An investigation of traditional methods for capturing engineering information in paper reports, computer programs, letters, telephone conversations, face-to-face meetings, and video conferences was used to develop a set of groupware tools that accomplished the similar functions in a groupware environment. Two collaboration tools called Notemail and XShare included application sharing, asynchronous text chat, shared whiteboards, audio/video conferencing, and session archiving and retrieval.

Kyng (1991), an Associate Professor of Computer Science at Aarhus University in Denmark, discussed collaborative design processes used for system design. Kyng viewed cooperation as a factor that must be integrated into computer support efforts. Systems must be designed so that they apply both to users and designers. System use must stimulate the flow of knowledge while allowing personnel to apply this knowledge to the design situation. Active participation in the design process is supported by development of groupware tools.

Shu and Flowers (1992) described development of a graphically rich, three dimensional, groupware design tool. Experiments determined that users preferred the simultaneous mode of editing versus turn taking. In addition, independent points of view optimized synchronous activity. Experiments led to development of another tool, Viewpoint, that allowed effective resolution between contrasting and arbitrary viewpoints. Shu and Flowers were researchers at the Computer-aided Design Laboratory, Department of Mechanical Engineering, MIT.

Decision support.

Huber (1990), a Fondrer Foundation Centennial Chaired Professor of Business at the University of Texas, outlined the effect of computer-assisted communication on decision making. Huber addressed the forms of communication and the effects these communication technologies could have on change in the workplace. Huber also discussed the effects of computer-assisted communications on decision making. Although there may be less face-to-face communication, managers and professionals can choose the medium that best fits the task and required decision. It is also possible that the decision making level will shift to a lower organizational level where personnel have the data necessary to make appropriate decisions.

A report by the National Research Council (1990) described theories of distributed decision making. It included definitions and models for decision making and principles for designing distributed decision making systems.

Drawing, graphical design, and presentation.

Drawing processes are as important to the design process as the drawings themselves. Bly and Minneman (1990), researchers at the Xerox Palo Alto Research Center (PARC), discussed a shared drawing system, called Commune, that included drawing, writing, and gesturing. Commune allowed designers remote access to shared drawing spaces and it provided a set of tools that allowed group interaction. Support is similar to face-to-face sessions. Minneman moved to the Center for Design Research, Stanford University.

Greenberg, Roseman, Webster, and Bohnet (1992) described issues and hurdles that surfaced during the design of two real-time, multiple-user drawing systems, Groupsketch and Groupdraw. Differences between the single and multiple-user applications were detailed. The authors discussed architectural schemes, participant registration, multiple cursors, network requirements, and the structure of the workgroup drawing primitives. Based on design and implementation results there was no compelling reason to use a centralized versus replicated architecture. Greenberg, Roseman, and Webster were with the Department of Computer Science, University of Calgary. Bohnet was with MPR TelTech, Vancouver, Canada.

A common display assists writing and drawing for workgroups. There must be a computerized system to more efficiently manipulate text and graphics. The system must also refine work and record it for future reference. Lakin (1988) presented an approach for the design of an appropriate drawing medium. The author discussed new computer examples, key features of text and graphic manipulation, and a proposed architecture. Lakin also discussed how graphics and text can be manipulated in a shared drawing system. The author was a researcher with the Center for the Study of Language and Information Center for Design Research, Stanford University.

Rimmer (1991), a microcomputer consultant and author, wrote the manual for the CorelDraw drawing program. This manual provided background on the structure of conventional computer-based drawing program. Software Publishing Corporation's (1991) Manual for Harvard Graphics drawing program also provided background on the structure of a conventional computer-based presentation graphics program. Both sources will be used to help structure primitives for a workgroup graphics application.

Tang (1991), from the System Sciences Laboratory, Xerox PARC, studied the work activity of small groups using a shared drawing space. Video-based interaction analysis techniques were used to study collaborative drawing activity. Design assumptions for individual users were examined to help build tools for collaborative drawing processes. Experiments showed that participants required more than the shared work environment for effective collaboration. To express ideas, users interactively create representations of ideas in the shared drawing space. The group will perceive, react, and build on these ideas to form drawing concepts. Two other important elements to emerge from this study were hand gestures for getting attention and the use of a drawing space to mediate user interaction.

Electronic mail.

Sproull (1991) described E-mail technology including system and social features. An E-mail system must provide access, naming conventions, transport, and group communication. The author also provided a listing and discussion of E-mail functions including editing, message selection, searching, deleting a message, copying, replying, automatic reply, displaying messages by category, finding addresses, maintaining group lists, and determining if a message has been read. Sproull is with the College of Humanities and Social Sciences, Carnegie-Mellon University.

Turoff (1991) performed research on group support systems at the New Jersey Institute of Technology. Turoff presented a historical perspective of computer-mediated communication and its relationship to designing for group support. The author divided messages into several parts. The message abstract includes name, identification of the author, data and time of creation, title of the item, keywords, and message status. The body of the message was included in the content section. Attachments could include graphic or binary files. Activities included viewing, voting, reviewing, doing, organizing, and notifying.

Interactive communication.

Benest and Dukic (1993), researchers from the Department of Computer Science, University of York, United Kingdom, described the Automated Office Metaphor (AOM) that turns information sharing into a conferencing system. Emphasis was placed on the ability to share captured and computed information in small project teams. AOM is the electronic equivalent of office forms. The authors described a number of electronic forms and implementation problems.

Successful collaboration requires awareness of both individual and group activities. Dourish & Bellotti (1992) discussed a study of passive awareness mechanisms that allowed users to be aware of and exploit information in shared environments while avoiding problems with active approaches. CSCW systems support this with information generation systems that are separate from the shared workspace. Users could move between loose and close collaboration and coordinate work dynamically. Both authors were researchers with Rank Xerox EuroPARC.

Ohkubo & Ishii (1990), from the NTT Human Interface Labs, Kangawa, Japan, proposed an approach to shared workspaces called TeamWorkStation (TWS). The TWS design included integrated real and virtual workspaces, a shared drawing surface, and smooth transitions between individual and shared workspaces. The system allowed development of objects and ideas by using a shared whiteboard surface. The authors described design objectives, implementation methods, and experimental results.

Stefik, Bobrow, Foster, Lanning, and Tatar (1987) were from the Intelligent Systems Laboratory, Xerox PARC. Their significant early work, supported by the Defense Advanced Research Projects Agency (DARPA), discussed collaborative software called Boardnoter that provides a multi-user interface emulating a chalkboard. Users input data and choices to interact with programs to retrieve information and solve problems The authors discussed multi-user interfaces for access to various workgroup computing information systems. They also discussed synchronous and asynchronous interchanges between multiple users.

Meeting and conference support.

Crowley, Millazzo, Baker, Forsdick, and Tomlinson (1990), from Bolt, Beranek, and Newman, Inc., worked on a DARPA sponsored system called MMConf. Computers have traditionally allowed asynchronous group work through shared file systems. MMConf explored how computers can support real-time, distributed, group collaborations. The authors discussed floor control policy that manages the distribution of events, object locking, message and file transfer, video tools, and a slide show implementation. The architecture has been implemented on UNIX systems as both a toolkit and conference manager. Applications that support real-time cooperative work have been produced and tested.

Written by Jay (1976) before desktop personal computers, this article furnished essential background on how a meeting can be structured and run. It discussed meeting functions and objectives. This is an old, but useful, reference that will allow workgroup meeting functions to be developed and understood. Jay was a producer of computer-based training films for industry.

Nunamaker, Briggs, and Romano (1993) discussed the integration of three levels of workgroup meeting technologies and how they can improve organizational productivity. Nunamaker, Briggs et al. presented a vision for a future meeting environment that included visual maps, graphics models, and specialized applications. Specialized applications included brainstorming comments, action item lists, and graphics of new products. Nunamaker, Briggs et al. were researchers with the Group Systems Development Group at the University of Arizona.

Nunamaker, Dennis, Valacich, Vogel, and George (1991) described an Electronic Meeting Support (EMS) system developed by the authors at the University of Arizona. EMS attempted to make meetings more productive through the application of information technology. Almost all important decisions in organizations are made by groups. Group meetings may have lack of clear goals and focus, lack of participation, and individuals may have hidden agendas. Often meetings end without a clear understanding or record of what was discussed. This paper presented results of EMS research that improved on these deficiencies. This excellent paper defined group meeting processes.

Rodden (1993) discussed electronic meeting systems that allow information sharing and group communication. Two CSCW approaches existed. One supported the exchange of information between users. The other developed systems that allow the cooperative sharing of information. The author considered technologies exploited by each approach.

Rodden also discussed iterative meeting systems that allowed information sharing and group communication. Rodden was a CSCW researcher at the CSCW Research Centre, Department of Computing, Lancaster University, United Kingdom.

Sarin and Greif (1988) conducted significant early research on the use of real-time conferencing systems to support joint work in numerous application areas. Conducted under DARPA and Office of Naval Research (ONR) sponsorship, this research developed a prototype system for real-time conferences. The research also identified design and implementation principles for group conference systems.

Stefik, Foster, Bobrow, Kahn, Lanning, and Suchman (1988) completed significant early research that defined the use of an electronic chalkboard to support workgroups and meetings. Meetings provided coordination or mediated intellectual decisions between groups of people. This article discussed the advantages of using computers to support functions that used to be supported by chalkboards. An experimental meeting room known as Colab was established at Xerox PARC. Collaborative processes were studied using face-to-face meetings. The project resulted in a usable meeting room and several tools to support meeting collaboration.

Borenstein and Thyberg (1991) described the operation of a successful multimedia mail and bulletin board called Messages. The "Messages" program was an interface to the Andrew Message System (AMS). Although it was easy to learn, Messages was extremely powerful and it satisfied both novice and expert users. The Messages system which is part of the Andrew Advisor System supports cooperative work that is not possible with other mail systems. The system has been used weekly by about 5300 people at Carnegie-Mellon University to read bulletin boards. This successful user interface is described in this paper. Borenstein worked at Bellcore and Thyberg was a researcher at Carnegie-Mellon University.

Vin and Chen (1992) presented a model for multimedia collaborations. The model consisted of a hierarchal abstraction of data streams, sessions, and conferences. The model supported both asynchronous and synchronous workgroups and it provided sophisticated access control and building blocks for multimedia applications and collaborations. Interfaces included joining a group, sending and receiving information, changing access, restructuring sessions and changing meeting control. Vin was a researcher in the Multimedia Laboratory, Department of Computer Science and Engineering, University of California, San Diego. Chen was affiliated with the IBM Thomas J. Watson Research Center.

Project management.

Kerzner (1992) presented computerized approaches for project management that can be extended to workgroup situations. In chapter 17 of this reference, Kerzner discussed computerized aids for project managers. Project management software features included planning, tracking, monitoring, report generation, project calendars, "what-if" analysis, and multi-project analysis. Kerzner, a Professor at Baldwin-Wallace college, has authored over 60 papers and 13 texts on program and project management.

Sathi, Morton, and Roth (1988) discussed project management and the Callisto project whose goal was to support large projects requiring interaction and cooperation. Callisto uses a program requiring significant interaction and cooperation between users. Project management tasks are broken down into three areas: activity management, product configuration management, and resource management. Elements in activity management include planning, scheduling, chronicling, and analysis. Product configuration management involves product management and change management. Resource management includes projection and assignment of resources, responsibility assignment, and critical resource control. Several models emerged from the early experiments. The authors conducted this research at the Intelligent Systems Laboratory, Carnegie-Mellon University.

***Scheduling and calendaring*.**

Greenwood (1992), from WordPerfect Corporation, described personal calendaring and group scheduling functions. The author considered both calendering and scheduling to be an extension of electronic mail. The review included an explanation of calendaring, file management, notebook, editor, E-mail, and scheduling functions.

Lange (1992) discussed the essential elements for the successful introduction of a groupware product in an organization. The implementers found that there were two highly related sets of factors for the use of groupware. First, the product, electronic calendaring, had to have well defined expected uses and clear guidelines for usage. Second, the organization had to revise procedures as the product was introduced to encourage and stimulate usage. Elements included calendaring, proposal enclosures, interface to E-mail, "In/Out" board, time management, and resource management. Lange was with the Center for Technology Research, Anderson Consulting.

Shared databases.

Celentano, Fugini, and Pozzi (1991), Italian researchers at the Universita di Beslia, provided an approach to document classification, filing, and retrieval using semantic model documents. The approach covered a classification scheme and retrieval system based on document roles. The paper included a discussion of the Kabiria project that used knowledge techniques for document management. The authors also discussed an approach to document classification, filing, and retrieval.

Greif (1992) described a case study of the design of a groupware spreadsheet. The author believed that the next generation of workgroup products would come from the union of communication and data sharing capabilities with desktop tools. For this to happen, both groupware tools and application software had to change. This paper described a case study for the design of a spreadsheet example that responded to the need for an integrated product. Greif, from Lotus Development Corporation, participated in developing Lotus Notes 1.0.

A major concern in workgroup computing is how shared data can be accessed. Greif and Sarin (1988) defined the scope of data sharing requirements for workgroup computing systems. Three cooperative work systems are discussed along with limitations of implementation techniques. Features for modifying roles and working relationships are examined. Due to the diversity of programs investigated, the authors concluded that their requirements were representative of a wide range of workgroup applications. This is a significant early work defining the scope of shared data requirements in workgroup computing systems. Sarin is with the Computer Corporation of America.

Holtham (1993), Bull Information Systems Professor of Information Systems, City University Business School, London, discussed information sharing tools for collaborative environments. The author identified five workgroup drivers: change, coordination, collaboration, control, and connectivity. A new grid provided a way of classifying groupware tools. The author argued that "bundles" (p. 292) of different groupware tools are required to support different business situations.

Malone, Grant, Lai, Rao, and Rosenblitt (1989) described a prototype information-sharing system called the Information Lens. The authors discussed Information Lens' information sharing capabilities. The Lens' system is based on a set of semi-structured messages for different message types. Each message type has a template with fields to hold information. To share information, rules are used to process messages within the system. The authors showed how Lens' information sharing capabilities could be used to support task training, meeting scheduling, and the distribution of engineering documentation. All five researchers were in a research group at MIT.

As workgroup computing matures, a major area of discussion will be how data is stored, maintained, and accessed in workgroup settings. Schwartz (1992), from Borland International, Inc., addressed a wide range of issues surrounding this subject. Schwartz states that "Close attention should be paid to the underlying technologies and the data delivery models each approach uses so that there is a close fit with the data needs. . . an organization can support." (p. 240).

Workflow management.

Bair (1993) reviewed several workflow software models and included a discussion of workflow elements. Workflow elements included forms, documents, activities, users, access, and graphics. A key consideration was "Does the model allow me to specify all that is necessary for the work to flow?" (p. 230). Bair was employed as a cooperative systems consultant with New Science Associates.

Bock (1992), a DEC Project Manager for collaborative systems, discussed workflow concepts and "coordination environments." The author contended that workflow concepts may be part of groupware. Groupware applications can enhance organizational effectiveness. Bock provided information on the development of a groupware language used to build "coordinated environments" (p.168).

Workflow software can be considered a type of groupware. Marshak (1992) defined a context for workflow software. The paper also provided measurement criteria for workflow software. Marshak was the Editor-in-Chief of the *Office Computing Report: Guide to Workgroup Computing.*

Palermo and McCready (1992), workflow software producers, defined and discussed different types of workflow software that can be used by groups in an integrated environment. Workflow software can be used by both individuals and groups to manage a wide variety of business processes in an integrated environment. Workflow software provides a language and tools for structuring work processes.

Schael and Zeller (1993) developed a functional architecture and a pilot workflow management project for financial reviews in an Italian bank. The architecture included input/output systems, workflow coordination, application processing, and a shared database. The authors discussed several different functional modules including message handling, data management, and document management.

Writing and editing.

Baecker, Nastos, Posner, and Mawby (1993) presented a design for user-centered collaborative writing software including a taxonomy and design requirements. The taxonomy detailed author roles, writing activities, document control methods, and writing strategies. The authors discussed results from several tests. Test results suggested that, although conflict resolution could be resolved by locking schemes, many times only a voice link is necessary to resolve writing conflicts. Both the taxonomy and conflict resolution techniques will be factored into the workgroup computing taxonomy development. Baecker's research team was with the Dynamic Graphics Project, Computer Systems Research Institute, University of Toronto.

Baydere, Casey, Chuang, Handley, Ismail, and Sasse (1993), from the Department of Computer Science, University College, London, described an investigation of how multimedia conferencing systems support collaborative writing. Multimedia conferencing offers different tools to perform collaborative writing. The result is similar to a face-to-face meeting. The authors discussed details of the collaborative writing process.

Denley, Whitefield, and May (1993) discussed issues related to design of a multimedia collaborative writing system. They progressed from basic concepts to the final design and its utility. Denley and Whitefield were with the University College, London. May was with Standard Elektrik Lorenz-AG, Germany.

Knister and Prakash (1990) developed a toolkit called DistEdit that allowed interactive group editors to be built for workgroup computing environments. They described details of editing primitives that can be used in this study. Both were with the Software research Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan.

Neuwirth, Kaufer, Chandhok, and Morris (1990) reported on a project to develop a "work in preparation" (p. 183) editor that allowed co-authoring and commenting to be studied. The study identified design issues for computer support processes. The authors included support for social interaction, cognitive aspects of authoring, and practicality in both types of interaction. Neuwirth et al. were researchers at Carnegie-Mellon University.

Posner and Baecker (1992) presented a taxonomy of group writing that will help groupware builders understand the collaborative writing process. The authors discussed the four areas including collaboration roles, the writing process, document control, and writing strategies. The paper also listed a set of design requirements for collaborative writing.

Sharples (1993) described basic collaborative writing strategies. The author suggested simple techniques for structuring and coordinating writing. A case study explored collaborative writing for an academic paper.

Sharples, Goodlet, Beck, Wood, Easterbrook, and Plowman (1993) studied collaborative writing issues and indicated that strategies related to writing are hard to uncover and analyze. In addition, it is difficult to design computer systems that support these processes. Implementors need to understand the broad issues concerning cognitive and social processes that motivate collaborative writing. The authors, from the School of Cognitive Sciences, University of Sussex, United Kingdom, provided information about collaborative writing issues that will be useful for this study.

Existing access models for collaborative writing are not satisfactory. Shen and Dewan (1992) from the Department of Computer Science, Purdue University, developed a new model for access control. The model described access control and rights that are applicable to collaborative writing and other workgroup computing areas. The model also associated displayed data with a set of collaborative rights and it provided a scheme for specifying access.

Hypertext is a promising approach for handling large, complex document sets that are used and maintained over a long period of time. Sobiesiak & Myopoulos (1991) described a system called ThyDoc that allowed knowledge-based software engineering technologies to document the authoring process. ThyDoc treated the authoring process as a knowledge acquisition process that formally and informally captured all of the knowledge needed by authors to design, develop, and maintain large hypertext documents. A prototype ThyDoc model was developed to illustrate document engineering. Sobiesiak and Myopoulos were from the University of Toronto. Sobiesiak also worked for the IBM Canada, Ltd. Laboratory.

Specific workgroup computing applications.

Digital Equipment Corporation's (1994a) paper on DEC LinkWorks Client for Microsoft Windows, Version 2.1 provided details of all LinkWorks workgroup software functions. DEC's (1994b) paper on DEC TeamLinks for Microsoft Windows, Version 2.0 provided details of TeamLinks workgroup software functions.

Gerwirtz (1994), who has published over 50 commercial software products for the MAC, Windows and UNIX, described Lotus Notes 3. The book is designed to help information managers and network administrators determine how Lotus Notes fits into their organization. The book described details of all Lotus Notes 3.1 application functions. Schulman (1994), a computer programmer, also described the Lotus Notes 3.1 application functions in extreme detail. Schulman included a complete menu and function reference.

Borland, Lorenz, and O'Mara (1993) described the details of Microsoft Windows for Workgroups 3.11. This document provided clear explanations and examples of workgroup functions. Borland has been a technical writer for over 12 years. Queen (1993) was a certified netware engineer with over 10 years of networking experience. Queen also described the Windows for Workgroups application. This reference details how use Windows for Workgroups to build a LAN.

D. *Workgroup Computing Taxonomies*

This section describes existing workgroup computing taxonomies along with their limitations and relationships to this study.

Ellis et al. (1991) defined an application taxonomy that is based on application level functionality. The taxonomy is not comprehensive and workgroup application categories overlap. Their taxonomy was intended to describe the diversity and scope of the groupware domain. This work will help define workgroup computing functional categories such as message systems, multi-user editors, decision support, conferencing, intelligent agents, and coordination systems.

Easterbrook, Beck, Goodlet, Plowman, Sharples, and Wood (1993), from the University of Sussex, United Kingdom, provided a top level structure for workgroup computing. Easterbrook et al. included generally recognized applications in their taxonomy; however, they provided little detail. This work will also help define workgroup applications such as E-mail, text conferencing, concept development tools, decision support, and collaborative writing.

Kedzierski (1988), with the Kestral Institute, Palo Alto, California, created a simple workgroup computing taxonomy based on initial studies of user interaction with the system. In Kedzierski's taxonomy, workgroup functions were based on "Communication Acts" (p. 253) such as questioning, griping, planning, requesting, and informing. Kedzierski also included system interaction and automation of these functions. This work is of little use for this thesis.

This work by Johansen, Sibbet, Benson, Martin, Mittman, and Saffo (1991) analyzed the time/space workgroup computing model presented in the introduction to this thesis. Johansen et al. charted the terrain and discussed workgroup computing building blocks. The authors also defined groupware categories that included electronic copyboards, decision making, polling systems, video conferencing, screen sharing, team rooms, shared files, group writing, conversational structuring, forms management, and voice mail. This work structured and validated workgroup computing categories defined in this thesis.

Dyson (1992), an author for Lotus Notes 1.0, also provided a framework for groupware. The author saw groupware as a "tool for change" (p. 10) in an organization. Dyson's groupware structure centered around the classification of groupware as information or workflow oriented. Categories were user-centered, work-centered, or process-centered. Although this was an innovative approach to groupware classification, it is of little interest to this thesis.

Egan (1993), a software consultant, provided another view of how to categorize workgroup applications. Egan broke down groupware into three groupings: decision making; workflow; and work management. This work was useful in defining broad groupware categories, but it provided little detail for individual application functions.

E. *WorkGroup Computing Architectures*

Workgroup computing taxonomies categorize workgroup computing functions while architectures explain how all of the pieces are linked together. This section discusses several existing architectures and some interesting approaches and interfaces that can link applications in an integrated fashion.

Englebart (1992), with the Bootstrap Institute, considered groupware as a means to an important end: "Creating truly high performance human organizations" (p. 77). Englebart proposed four architectural requirements for groupware systems. They were global and individual vocabulary control, multiple look-and-feel interfaces, shared-window teleconferencing, and linkage between hyperdocuments and other data systems. The author discussed a workgroup computing system called CODIAK that supported these requirements. This work will be used to help define workgroup computing architectural elements for this thesis.

Workgroup computing systems must support collaboration in complex situations. According to Power and Carminati (1992), these systems must function both synchronously and asynchronously at a single or multiple locations. The authors defined a set of workgroup computing system features that may be useful in defining an updated workgroup computing architecture.

Rhyne and Wolf (1992) also discussed workgroup computing architectures in the context of synchronous or asynchronous collaborative processes. If asynchronous capability is missing, additional users cannot join a collaboration in process. Asynchronous collaboration allows activity by a subgroup or individual to be documented in a permanent record that can be commented on by other members of the group. This point must be considered when designing a collaborative computing architecture. Rhyne and Wolf were researchers with the IBM Thomas J. Watson Research Center.

A project by Cook, Birch, Murphy, and Woolsy (1991), groupwork researchers with Object Designers ltd. and IBM, explored ideas on how a distributed system could enhance the effectiveness of communications, support planning, and complex cooperative tasks. The authors' approach created software models that were used in group experiments. These models will be used to develop a revised workgroup computing architecture.

Sarin, Abbott, and McCarthy (1991), with Xerox Advanced Information Technology, presented another type of model that is based on units of work. Their concept included flexible routing of work to personnel who will do the work including presentation and manipulation of documents. The model was implemented as an object-oriented network service. This model was important because it is one view of the collaborative process that can help explain the group dynamics in a workgroup computing environment.

An architecture developed by Benford, Mariani, Navarro, Prinz, and Rodden (1993), a multinational European research group, described a workgroup computing environment called MECCA. This environment facilitated interaction between different groupwork functions. The architecture was an open approach that had many similarities to the model proposed by Sarin, Abbott, and McCarthy (1991).

Roseman and Greenberg (1992) presented another flexible approach based on open protocols. The authors described a workgroup computing architecture called Groupkit. Three strategies included an extendable, object-oriented, run-time architecture; transparent overlays for adding general components of groupware; and open protocols to allow designers a wide range of interface and interaction policies.

In follow-up research, Roseman and Greenberg (1993) described implementation techniques for their workgroup computing approach and they illustrated examples of floor control, conference registration, and brainstorming. The authors were from the Department of Computer Science, University of Calgary.

Malone, Lai, and Fry (1992), from The Center for Coordination Science, MIT, provided basic components of a groupware architecture. Users can create applications by combining objects, views, agents, and links in a building block approach. Functionality is provided that is equivalent to well known systems such as The Coordinator, Lotus Notes, and The Information Lens. Established primitives provide a tailoring language for application construction.

Pastor and Jager (1991), from the PECOS project, identified a list of workgroup computing system requirements and building blocks that were used to define a workgroup computing architecture in a European design project. According to Pastor and Jager "The list is neither complete nor accurate. It served as a starting point for the identification of a set of common support functions" (p. 110). This list will be used to define workgroup computing functions for this thesis.

The Assistant for Cooperative Work (ASCW) was a system for managing cooperative work. Developed by Kreifelts and Prinz (1993), it included a task manager, information system, and conference tool. This system overcame predefined and rigid structures imposed by other types of groupwork systems. Elements of this flexible model can be used in this thesis. Kreifelts and Prink are researchers with the GMD, German Research Center for Computer Science, Institute for Applied Information Technology.

The lack of a consistent set of data services across corporate platforms has inhibited the expansion of groupware. To reduce the impact of this problem, Lepick Kling (1992) described a homogeneous environment using TCP/IP or PIPES Platform, an object-oriented operating system. Lepick Kling's work provides insight into architectural requirements for enterprise-wide groupware developments.

Enterprise-wide workgroup systems also require support for interaction by multiple users. Bentley, Rodden, Sawyer, and Somerville (1994), another multinational European workgroup computing research group, designed and developed a software architecture that supported this concept. They discussed centralized, replicated and hybrid architectures, and problems associated with each approach. Their research adds to enterprise-wide workgroup computing research completed by Power and Carminati (1992) and Roseman and Greenberg (1993).

Reinhard, Schweitzer, and Volksen (1994), a German research group, presented two different approaches to a workgroup computing architecture based on workgroup application criteria and requirements. They proposed an architecture that can be either centralized or distributed. Their applications research included systems in three different frameworks: collaboration transparent applications, a centralized workgroup implementation, and a common workspace interface.

F. *Application Programming Interfaces (APIs)*

Harrison (1994), a computer system analyst and programmer based in California, discussed application program interfaces (APIs) for networking architectures. These APIs allowed local and wide area networks to interface seamlessly. Various API layers performed different interface functions. This article will be used to define workgroup computing APIs or interface standards.

Several sources list details of Windows' APIs. They will be used to help structure program interfaces for functions defined in this thesis. Works by Harrison (1994), Schildt, Pappas, and Murray (1994a & b), and Conger (1992) describe several hundred of these APIs. Used as source material, these APIs will provide guidelines for structuring new workgroup program interfaces.

G. *Summary*

The first set of sources addressed background information concerning CSCW and workgroup computing. Groupwork was defined and core workgroup computing issues were identified. The authors also discussed methods for sharing information space and how to adapt workgroup technology to the organization.

According to Ellis et al., groupware is a merging of computers, large information databases, and communication technology. There is a need to integrate workgroup computing functions to support more complex implementations. Today, no single groupware product addresses all workgroup computing needs. How groupware applications are used and how they provide control for the processes they support must be factored into design of workgroup computing systems. In addition, elimination of duplicate workgroup computing functions can simplify the software programming and groupware implementation process. Highlighting the differences between groupware and traditional computer support, Grudin (1994) concluded that groupware has targeted smaller system problems rather than problems affecting larger organizational goals. Groupware has been primarily off-the-shelf and it has not been integrated.

Information exists in the literature to support development of a detailed workgroup computing taxonomy. The literature review identified 11 workgroup application areas and 51 applications or journal articles where there was sufficient detail to allow extraction of workgroup tasks and primitives. A review of manuals for leading workgroup computing applications provided detailed information on how these products function in a workgroup computing environment. Groupware application descriptions, application manuals, and product description documents will provide enough detail on application functions and tasks to develop a comprehensive workgroup taxonomy.

Most workgroup computing architectures only look at a subset of workgroup applications. Alternative architectures are available. Several authors discussed features and design requirements for workgroup computing systems that provided a starting point for design of an integrated workgroup computing architecture. In addition, API function specifications developed as a result of the taxonomy and architecture can be patterned after Windows APIs.

Chapter III. Workgroup Computing Application Functions and Primitives

A. *Introduction*

The literature review presented a discussion of available references in a number of workgroup computing application functional areas. It was useful to review typical workgroup applications that could be performed in each of these areas to further assist in defining detailed workgroup tasks and primitives. When combined with information available in the literature, this review helped define a complete set of workgroup computing tasks and primitives for inclusion in the taxonomy.

A Workgroup Computing Applications List compiled for a practicum report, completed by the author (Von Worley, 1994), was a source for typical workgroup applications. Work and job reference sources compiled by job experts, including Goldstein and Healey (1990), Krantz (1992), and Petras and Petras (1993), helped to define workgroup functional areas.

B. *Communication*

***Interactive storyboarding and bulletin boards*.**

Using networked computers and television, users access and share video, audio, data streams, and sessions to perform storyboarding or share bulletin boards. Multiple users access interactive communication systems for entertainment, home shopping, education, problem solving, and other uses (Stefik et al., 1987).

Any authorized user can logon to the interactive communication workgroup system and request a workgroup session. The format of the session can be preplanned or tools can be used to tailor it to individual preferences. The workgroup session leader defines and approves the access list. The workgroup system notifies other users that a workgroup session is underway. The system then invites users to join the session. Users logon as necessary. Session leaders or designated users can request that workgroup sessions be merged or ended. When a user leaves a workgroup session, a message stating that the user is gone is broadcast to remaining workgroup members. Various users can input data and choices to the bulletin board or storyboard that allows other users to access and retrieve the data and jointly solve problems. Conflicts are mediated by allowing data file access to only one user at a time.

In some systems, gesturing is performed with workgroup software support tools. Gesturing is a specific movement that allows a user to write on or direct attention by pointing to some portion of the shared workgroup area during a session. With this type of workgroup application, only synchronous interchanges are possible. As discussed earlier, synchronous exchanges take place at the same time and asynchronous exchanges take place at different times (Benest & Dukic, 1993; Stefik et al., 1987; Turoff, 1991; Vin & Chen, 1992).

This form of interactive communication allows users to prepare slides and small documents, do pre and post-session file transfer, and perform on-line presentations. Multimedia objects can be added if the network bandwidth will support it. Real-time tools are used for pointing, marking, and manipulating text and graphics (Stefik et al., 1987).

Patient monitoring and remote diagnosis.

Nurses and trained staff monitor a patient's status via networked computer links. Authorized users can establish links from either end of the communication link and group access rights can be designated. Users logon using passwords. This type of interactive communication system can link several local or remote hospitals, doctor's offices, or homes. Paramedics transfer patient data to hospitals where doctors and nurses remotely diagnose a patient's status and provide diagnosis and guidance to the field unit (Vin & Chen, 1992). With this type of workgroup application, only synchronous interchanges are possible (Benest & Dukic, 1993).

Communication workgroup primitives.

Communication workgroup primitives include opening the storyboard, bulletin board, or monitoring session, configuring the workgroup session, determining access rights and control, logon, logoff, sending or receiving data, editing, filing transfer, workgroup editing functions, message queries, gesturing functions, merging multiple sessions, changing access, changing session relationships, and changing session controller (Ohkubo & Ishii, 1990; Vin & Chen, 1992).

C. *Computer-aided Design (CAD), Drafting, and Rendering*

Engineering design.

Engineering designers and drafting support personnel use computer-aided tools to assist in the design process. Local or geographically separated shared drawing spaces are used to help capture and manipulate design information. A workgroup session leader can open or close a workgroup design session. The leader can also open or restrict access to the session by preparing access lists. The system notifies users that they can join an established session. Logons occur and users interact using software tools for pointing, gesturing, writing, and attracting attention (Glickman & Kumar, 1993; Kyng, 1991).

Drawing spaces are configured by the session leader and they can include public or private design spaces. During the drawing and edit phases, conflicts are resolved by a drawing control system that assigns each object to an owner. Users must request permission to access and change drawing objects they do not own or to change object ownership. With this type of workgroup application, either synchronous or asynchronous interchanges are possible (Benest & Dukic, 1993). When a user logs off from a session the system notifies other users that they are gone. Users also use shared whiteboards, shared text and drawing spaces, audio-video conferencing and meeting support, and asynchronous text chat to enhance the design process (Glickman & Kumar, 1993; Kyng, 1991).

Rendering.

Graphic artists jointly render architectural drawings, automotive designs, and advertising layouts without being in the same location. Artists share a common drawing area using remote access and selected drawing tools (Bly & Minneman, 1990). The lead artist must open or close a workgroup drawing space and determine who can access that space. Users logon to a session by using a password for access control. Pointing and design tools allow pointing, gesturing, and manipulation of the workspace text and graphics. Drawing objects are controlled to determine who can access them. Any user can request permission to transfer control of a drawing object to another user's control so that an object can be edited (Glickman & Kumar, 1993; Kyng, 1991). Either synchronous or asynchronous interchanges are possible (Benest & Dukic, 1993).

CAD, drafting, and rendering workgroup primitives.

Workgroup primitives include initiating and closing a workgroup session, access designation and control, logon, logoff, gesturing including pointing, writing, erasing and directing attention, workgroup conflict resolution functions, retrieving and storing images, switching between shared and personal workspaces, merging workgroup sessions, changing access, changing session relationships, and changing session controller, message queries, and gesturing functions. Editing functions include moving to personal workgroup space, moving to shared workgroup space, and drawing object ownership and transfer. Filing primitives include importing and exporting data or objects and snapshots or printouts from the workgroup session (Ohkubo & Ishii, 1990; Vin & Chen, 1992).

D. *Decision Support*

Problem solving, decision making, and intelligent assistants.

Management and technical staff use computers, databases, and computer conferencing and meeting support to support problem solving and decision making. Any manager can initiate a workgroup session and designate workgroup members. Sessions can be open or restricted or merged with other workgroup sessions. Workgroup members with access rights are notified about the session. Users logon to participate in information collection, brainstorming, organizing ideas, polling, consensus building, and voting stages of the problem solving process. Selected messages can be encrypted. Using selection criteria defined by the user, artificial intelligence front-ends use declarative rule sets to sort information. An example of this type of rule set is described below under "Intelligent E-mail Front-ends." Intelligent functions also assist in outlining alternatives and narrowing decision choices. Users can meet synchronously or asynchronously on the system to discuss choices and arrive at mutually agreeable decisions (Benest & Dukic, 1993; Huber, 1990; Stefik et al., 1987; Vin & Chen, 1992).

Decision support workgroup primitives.

Workgroup primitives include session opening and closing, access rights, access lists and password control, logon, logoff, sharing data streams, data handling and archiving functions, message encryption, polling, consensus building, and voting. Primitives for intelligent functions include filters, sorting, and declarative rule guidelines (Borenstein & Thyberg, 1991; Khoshafian, et al., 1992).

E. *Drawing, Graphical Design, and Presentation*

Package design.

Graphic artists and illustrators jointly design advertising and other product layouts without being in the same location. Graphic designs can be rendered for video and motion pictures including animation. Artists share a common drawing area using remote access and selected drawing tools. Locking mechanisms control access to drawing objects or shared drawing spaces (Bly & Minneman, 1990).

The lead illustrator can open or close a workgroup drawing space and determine who can access that space. Once notified of the session, users logon by using a password for access control. Pointing and graphical design tools allow pointing, gesturing, and manipulation of the workspace text and graphics. Drawing spaces are configured by the session leader and they can include public or private design spaces. To reduce conflicts, drawing objects are controlled by the system to determine who can access them. Any user can request permission to transfer control or ownership of a drawing object to some other user for editing. Users must request permission to access and change drawing objects they do not own or to change object ownership. Either synchronous or asynchronous interchanges are possible (Benest & Dukic, 1993). When a user logs off from a session the system notifies other users that the user is gone (Glickman & Kumar, 1993; Kyng, 1991).

Multimedia presentations.

Graphics designers produce multimedia presentations using a shared work area. After the workgroup session is set up by the lead designer, notified users logon to a session by using a password for access control. Key authentication can also provide access control. Pointing and design tools allow pointing, gesturing, and manipulation of the workspace text and graphics. Multimedia files are passed to remotely located users over high bandwidth networks. Design objects are controlled to determine who has permission to access them. Any user can request permission to transfer control of a drawing object to their control (Glickman & Kumar, 1993; Kyng, 1991). Either synchronous or asynchronous interchanges are possible between multiple users with this type of workgroup application (Benest & Dukic, 1993). When a user logs off from a session, the system notifies other users that they are gone (Glickman & Kumar).

Drawing, graphical design, and presentation workgroup primitives.

Workgroup primitives include initiating or closing a workgroup session, access designation and control, logon, logoff, gesturing including pointing, writing, erasing, and directing attention, retrieving and storing images, switching between shared and personal workspaces, conflict resolution, merging workgroup sessions, changing access, changing session relationships, changing session controller, and message queries. Editing functions include workgroup drawing editing functions, moving to personal workgroup space, moving to shared workgroup space, and drawing object ownership and transfer. Filing primitives include importing and exporting data or objects and snapshots or printouts from the workgroup session (Ohkubo & Ishii, 1990; Vin & Chen, 1992).

F. *Electronic Mail (E-mail)*

Multimedia messages.

Individuals at multiple locations prepare, send, and receive multimedia messages using networks, text, and embedded objects including graphics, video, and sound. Text content is sent in a message with attachments that can be embedded objects, libraries of objects, or databases. Object Linking and Embedding (OLE) allows embedded objects to be viewed in each mail program without using separate viewers (Khoshafian et al., 1992). Any user with authorization can access the E-mail system. Responses are normally asynchronous (Benest & Dukic, 1993). The pervasiveness of these types of workgroup systems demands implementation of access control and security measures to prevent inadvertent or deliberate access to sensitive information. Access and distribution lists allow users to logon to the system to extract incoming mail from their mailboxes and send mail to predefined individuals or groups. Sensitive messages can be encrypted using RSA, DES, or other public or private-key cryptosystems (Russell & Gangemi, 1991).

E-mail messages can be displayed by category (Sproull, 1991). Available message windows include sending, reading, selecting, deleting, copying, or replying to other messages. Multimedia messaging systems should be transparent to users by allowing them to read, print, and manipulate text and multimedia objects using identical procedures. Other functions include voting, return receipts, redistribution notices, and folder subscription notifications and invitations (Borenstein & Thyberg, 1991).

Intelligent E-mail front-ends.

Intelligent front-ends can enhance E-mail by providing lists of addressees based on attributes or relationships. Messages can be forwarded based on interests, expertise, background, "reports to" or "works with" relationships, or other attributes. Users search, filter, abstract, and store messages using declarative rules that specify what an individual wants to do or how to forward a message to different users. Filing rules can also exist (Khoshafian, et al., 1992). A short example of a declarative rule set follows:

"If message subject is marketing; then copy message to marketing folder."

"If message subject is multimedia; then copy message into multimedia folder and forward copy to 'gsmith'."

"If message subject contains a deadline; then keep in mailbox and provide alert one week before deadline date."

E-mail workgroup primitives.

Workgroup primitives include global and personal E-mail group designations, access rights, access lists and password control, logon, logoff, message distribution, handling, and archiving functions, OLE attachments, message encryption, permissions for others to access a user's mailbox, search, return receipt, deferred message delivery, and voting. Primitives for intelligent functions include filters, sorting, and declarative rule guidelines (Borenstein & Thyberg, 1991; Khoshafian et al., 1992).

G. *Meeting and Conference Support*

Meetings and conferences.

Multiple participants hold conferences between two or more locations. Shared and private spaces, polling, consensus building, voting, and participant anonymity are available. Communication mediums include text, graphics, video, and voice over networked links. An electronic whiteboard can be used for note-taking (Sarin & Greif, 1988).

A group leader creates a computer meeting room and schedules a session by accessing group calendars. The leader sends a message to all invited group members. The group leader develops a meeting agenda and transmits it to the group via E-mail for comment. Group members respond asynchronously. Invited group members are prompted by the workgroup system to join the meeting at the scheduled time. Logons occur and the work session begins. Activities can include document review, issue discussion, or multimedia presentations. Depending on the type of session, participants, at their discretion, can remain anonymous (Nunamaker, Briggs et al., 1993). When a user logs off from a session the system notifies remaining users that the user is gone. (Glickman & Kumar, 1993).

Remote training.

Using networked computers, students and teachers input lesson information and share data streams, sessions, and conferences to provide remote training classes. Students and the instructor use interactive text, video, audio, or whiteboards to display information, perform exercises, answer questions, and evaluate responses to questions or test situations. Meetings and conferences have the same system controls. Meeting control is used to allow access to the meeting tools and prevent collisions between participants. Synchronous or asynchronous interchanges are possible (Benest & Dukic, 1993; Stefik, et al., 1987; Vin & Chen, 1992).

Meeting and conference support workgroup primitives.

Workgroup primitives include session planning, group messaging and calendar functions, session management, group definition, access control, definition of shared and private spaces, group interaction, and meeting control. Idea generation, idea organization, alignment of related information, prioritization, participant anonymity, conference roles, presentation of status information, and message and file transfer are other important primitives (Jay, 1976; Nunamaker et al., 1991; Sarin & Greif, 1988).

H. *Project Management*

Planning and resource management.

Managers use planning, tracking, and monitoring software on networked computer systems to perform project management functions including resource control between geographically separate locations. Individual and shared workspaces will be available. Locking mechanisms allow users access to data while preventing other users from simultaneously changing the same data. Asynchronous and synchronous access will be available (Kerzner, 1992; Sathi et al., 1988). Users also use shared whiteboards, audio-video conferencing and meeting support, and asynchronous text chat to enhance the project management process (Kyng, 1991; Glickman & Kumar, 1993).

Managers will develop resource driven schedules that can be distributed to the workgroup for review and comment. Access lists will allow specified parties to access and change the resource schedule. Intelligent project management assistants can track project progress and ensure that the appropriate workgroup is notified when they must monitor or change a resource allocation (Kerzner, 1992).

Project management workgroup primitives.

Workgroup primitives include schedule and resource planning, schedule and resource management, group definition, access control, locking controls, group interaction, and conflict resolution. Other important primitives are idea generation, idea organization, alignment of related information, prioritization, participant anonymity, schedule and resource control, presentation of status information, and message and file transfer (Kerzner, 1992; Kyng, 1991).

I. *Scheduling and Calendaring*

Meeting planning and group calendars.

Managers and staff use networked computers to maintain personal and group calendars, automate repetitive tasks, propose meetings, send meeting invitations, block time, and prevent inadvertent scheduling of overlapping events and resources. User controls include the capability to add personal activities to a schedule and accept or decline meeting invitations. The system tracks confirmations and updates user schedules and calendars. A proxy feature allows individuals to view a user's private schedule (Lange, 1992).

In addition to scheduling meetings, managers can schedule people, groups, rooms, and equipment. The scheduling system automatically highlights conflicts and updates schedules when invitations are accepted or declined. The system can generate prioritized "to do" lists based on workgroup or individual inputs. Work tracking and automatic scheduling of repetitive events and tasks is also possible (Greenwood, 1992; Lange, 1992).

Scheduling and calendaring workgroup primitives.

Workgroup primitives include defining group calendars, group definition, access control, logon and logoff, group interaction and conflict resolution, locking-out and sharing information, access to private schedules, proxy authorization, automation of repetitive tasks, invitations and acceptances or declinations, time blocking, scheduling of overlapping events and resources, and file management. Other primitives are workgroup and individual "to do" lists, responding to "to do" requests, reminders, and automatic date programming (Greenwood, 1992; Lange, 1992).

J. *Shared Databases*

Item control databases.

Users remotely access, share, and update computerized databases that support inventory control, stock management, parts control, personnel or medical records management, order or product management, package tracking or other work areas. Sharing and updating between several adjacent or remote locations can be either asynchronous or synchronous. OLE allows embedded objects to be viewed in each database without using separate viewers (Khoshafian, et al., 1992). Database owners set access control and security levels. Both owners and users access complete or partial data sets and determine inconsistent data content (Greif, 1992; Greif & Sarin, 1988).

Database managers must insure access control and security are defined for each database. Access lists and passwords allow users to logon and logoff while allowing selective read-only or read and write access to portions of the database. Most transactions in these systems are asynchronous and conflict resolution is a small problem. When adding to or editing a database, a larger problem is concurrency control where each transaction must preserve consistency of the data file. The effect of running many transactions is that they must be queued to avoid conflict and preserve data consistency. This can slow down the database system and restrict access. Longer transactions increase the probability of "transaction abort" (Greif & Sarin, p. 497). If transactions are short, data locking and time-stamping can help resolve conflicts quickly (Greif, 1992; Greif & Sarin, 1988).

Shared database workgroup primitives.

Workgroup primitives include user definition and approvals, access and password control, database segmentation, logon, logoff, and editing procedures. Other important workgroup primitives are OLE, concurrency control, locking procedures, time-stamping, and data viewing (Greif, 1992; Greif & Sarin, 1988).

K. *Workflow Management*

Workflow systems.

In workflow systems, users design workflow setup, assign work, monitor flow, create deadlines, resolve status, determine current activity, and get coordination and approvals on work packages. The system keeps track of the status of multiple work packages and prompts users to complete required actions. The system returns completed actions to the initiator for distribution, filing, and archiving. Users interface asynchronously with multiple personnel to determine work flow status (Bair, 1993; Bock, 1992; Palermo & McCready, 1992). Intelligent agents monitor the workflow system and provide users with workflow routing, branching, and status information for relevant items (Bock, 1992).

Workflow management workgroup primitives.

Workgroup primitives include workflow routing design, access control and permissions, logon, logoff, routing and branching, serial or parallel editing, and approvals. Other important primitives are signature validation and authentication, workflow rules, status monitoring, OLE, distribution, filing and archiving, and information filtering. (Bair, 1993; Bock, 1992; Greif, 1992; Greif & Sarin, 1988).

L. ***Writing and Editing***

Multimedia reports and other correspondence.

Authors initiate multimedia collaborative writing sessions to brainstorm, research, plan, write, edit, and review reports and other correspondence using text, multimedia objects, and annotations. Shared and private workspaces using video and audio links can be available. The lead author defines and approves the access list and the style and content rules. The workgroup system notifies other authors that a workgroup session is underway and it invites users to join the session. Users logon as necessary. The lead author or a designated author can request that workgroup sessions be merged or ended. Control strategies and locking mechanisms are used to manage conflicts and the concurrent writing efforts of several co-authors. Both asynchronous and synchronous access is available (Baecker et al. 1993; Denley, Whitehead, & May, 1993; Sobiesiak & Myopoulos, 1991).

Writing and editing workgroup primitives.

Workgroup primitives include session control, access control and permissions, logon and logoff, merging sessions, routing and branching, and serial or parallel editing. Other important primitives are document segmentation, style and content rules, editing rules, concurrency and version control, revision status, status monitoring, links between objects, and information filtering (Bair, 1993; Bock, 1992; Greif, 1992; Greif & Sarin, 1988).

M. *Summary*

The preceding review provides an initial set of workgroup computing tasks and primitives that must be included in a workgroup computing taxonomy. The number of different workgroup applications is extensive and this review was intended to provide only a sample for comparative purposes. It was useful to review typical workgroup applications that may be performed in each of the functional areas to further assist in defining detailed workgroup tasks and primitives.

A Workgroup Computing Applications List compiled for a practicum report, completed by the author (Von Worley, 1994), was used to define typical workgroup applications. Work and job reference sources compiled by job experts, including Goldstein and Healey (1990), Krantz (1992), and Petras and Petras (1993), were used to define workgroup functional areas.

This review illustrates that many common workgroup computing tasks and primitives exist in the various workgroup application functional areas. When combined with information available in the literature, this review helped define a complete set of workgroup computing tasks and primitives for inclusion in the taxonomy.

Chapter IV. Methodology

A. *Hypothesis*

Review of the available workgroup computing literature and the review of applications and primitives strongly suggested that sufficient information was available to develop a detailed workgroup computing taxonomy. Several workgroup computing taxonomies existed in the literature; however, they were limited in scope and depth. The taxonomy that will be developed later in this study will comprehensively cover a broad range of workgroup functions and it will include details at the task level.

The literature review also suggested that analysis of the taxonomy could provide a set of common workgroup tasks or primitives and common workgroup support functions. These common functions and primitives were used in this thesis to develop a functional workgroup computing architecture that included necessary workgroup computing tasks and support tasks. The new taxonomy and architecture permitted preparation of a set of workgroup computing Application Programming Interface (API) specifications.

B. *Limitations of Current Workgroup Computing Architectures*

Workgroup computing architectures discussed by Kreifelts and Prinz (1993), Pastor and Jager (1991), Reinhard et al. (1994), and Roseman and Greenberg (1992) are applications oriented. A few, including those by Benford et al. (1993) and Malone et al. (1992), are task oriented. No single architecture discusses every one of the reviewed workgroup computing application or task functions. Commercial workgroup computing products present a tightly structured approach to workgroup computing and they do not include all workgroup computing functions. In order to effectively integrate workgroup experiences, interests, and approaches, Grudin (1991a), a primary workgroup computing researcher, states that "we have to go beyond what is shared and explore the differences" (Grudin, 1991a, p. 91). In a later article, Grudin (1994) contends that groupware will be more successful if it is integrated with features that support individual activity. Frequently used features should be readily available to the user and groupware components should not interfere with these frequently used features (Grudin, 1994).

C. *A Different Approach to Workgroup Computing*

A review of existing workgroup computing taxonomies and architectures indicated that there may be a better way to structure a workgroup computing architecture. If all workgroup functions and primitives could be identified and common elements consolidated for both application and support functions, an architecture could be specified that would allow an individual to select functions or tasks that they wanted to perform.

This approach differs from existing architectures that provide an interface to a number of different workgroup applications. Instead of being application dependent, the workgroup system would offer a full range of tasks and primitives to address a complete range of workgroup computing tasks and support functions.

If the workgroup architecture includes a comprehensive set of workgroup functions and tasks and the users are networked together, the architecture allows any defined workgroup task to be accomplished. This architecture will allow new or different types of work functions to be accomplished within the existing structure by linking together basic tasks or primitives. With the availability of a complete set of basic workgroup computing tools, there will be no need to produce another workgroup application to perform the new functions. Users only need to select the tasks that they want to perform. If required, new APIs can be developed to easily add profoundly different tasks that did not exist when the architecture was developed.

D. *Creating a New Architecture*

Creating the new architecture required that a detailed workgroup computing taxonomy be developed. This was necessary so that all existing workgroup computing tasks and primitives could be placed into a single structure that, when analyzed, showed common workgroup and workgroup tasks and support functions. The taxonomy described workgroup tasks and primitives in fine detail so that common elements could be identified, categorized, extracted, and combined. The reviewed literature and tasks and primitives identified in the application reviews were used as source material to develop this taxonomy. Tasks and primitives compiled from the taxonomy were included in the new architecture. Implementation of this architecture required development of workgroup application programming interface (API) specifications for newly defined common workgroup computing tasks and primitives.

E. *Workgroup Computing Taxonomy Development Methods*

Using the following methods, this study developed a comprehensive workgroup computing taxonomy, architecture, and workgroup computing API specifications.

The first task was to develop a workgroup computing taxonomy functional structure. This structure was based on previous work completed by the author (Von Worley, 1994), analysis of workgroup computing applications included in the literature, and the earlier discussion of workgroup computing application tasks and primitives. The taxonomy structure includes the following workgroup computing functional areas.

1. Computer-aided Design, Drafting, and Rendering
2. Decision Support
3. Drawing, Graphical Design, and Presentations
4. Electronic Mail
5. Interactive Communication
6. Meeting and Conference Support
7. Project Management
8. Scheduling and Calendaring
9. Shared Databases

10. Workflow Management

11. Writing and Editing

As the taxonomy development continued, this initial structure was reviewed for possible consolidation of functional areas, splitting of functions into more than one area, or adding new functional areas.

The second task was to use application information in the literature to extract and develop detailed workgroup computing tasks and primitives. Detailed workgroup computing tasks and primitives were derived for each listed workgroup computing functional area. Review and analysis of technical literature, application literature, and manuals for each specified workgroup computing functional area was used in conjunction with the applications review presented earlier to derive workgroup computing or workgroup support tasks and primitives. The taxonomy included these tasks and primitives. Nonworkgroup tasks that did support workgroup activities were included in the taxonomy only if they added value or clarity to the analysis.

To accomplish this, each workgroup computing functional area was divided into major workgroup computing functions. Further, each major workgroup computing function was broken down into one or more primary functions. Each primary function was composed of subfunctions that included one or more items that allowed the primary function or subfunction to be carried out. Each primary workgroup or supporting function was broken down into the lowest level of subfunction. If a primary function could not be divided into additional subfunctions it became the lowest level.

Each of these lowest level subfunctions performs a single independent task. Figure 3 illustrates this breakdown structure. This breakdown resulted in a hierarchical structure that progressively divided workgroup application functions and tasks from the highest to the lowest level. A definition describes each primary function or subfunction.

This procedure is illustrated below for the Shared Drawing and Graphical Design "Workgroup" major function.

Workgroup (Major function)

Start workgroup session (Primary function): Workgroup session leader initiates unique workgroup session and is prompted to establish session registration criteria allowing selected participants to register by logging on using user IDs and passwords.

Unrestricted (Level 1 subfunction): No list of attendees is required. Any authorized user is allowed to access the shared workspace.

Restricted (Level 1 subfunction): Allows the session leader to designate users that can access to the workgroup session.

Define access list (Level 2 subfunction):

Workgroup session leader defines a list of users that are authorized access.

Enter "User IDs" (Level 3 subfunction): Workgroup session leader enters a list of authorized user IDs.

Logon to workgroup session (Level 2 subfunction): Users can logon to shared workspace. Host name, port, and user name pass to the session registrar.

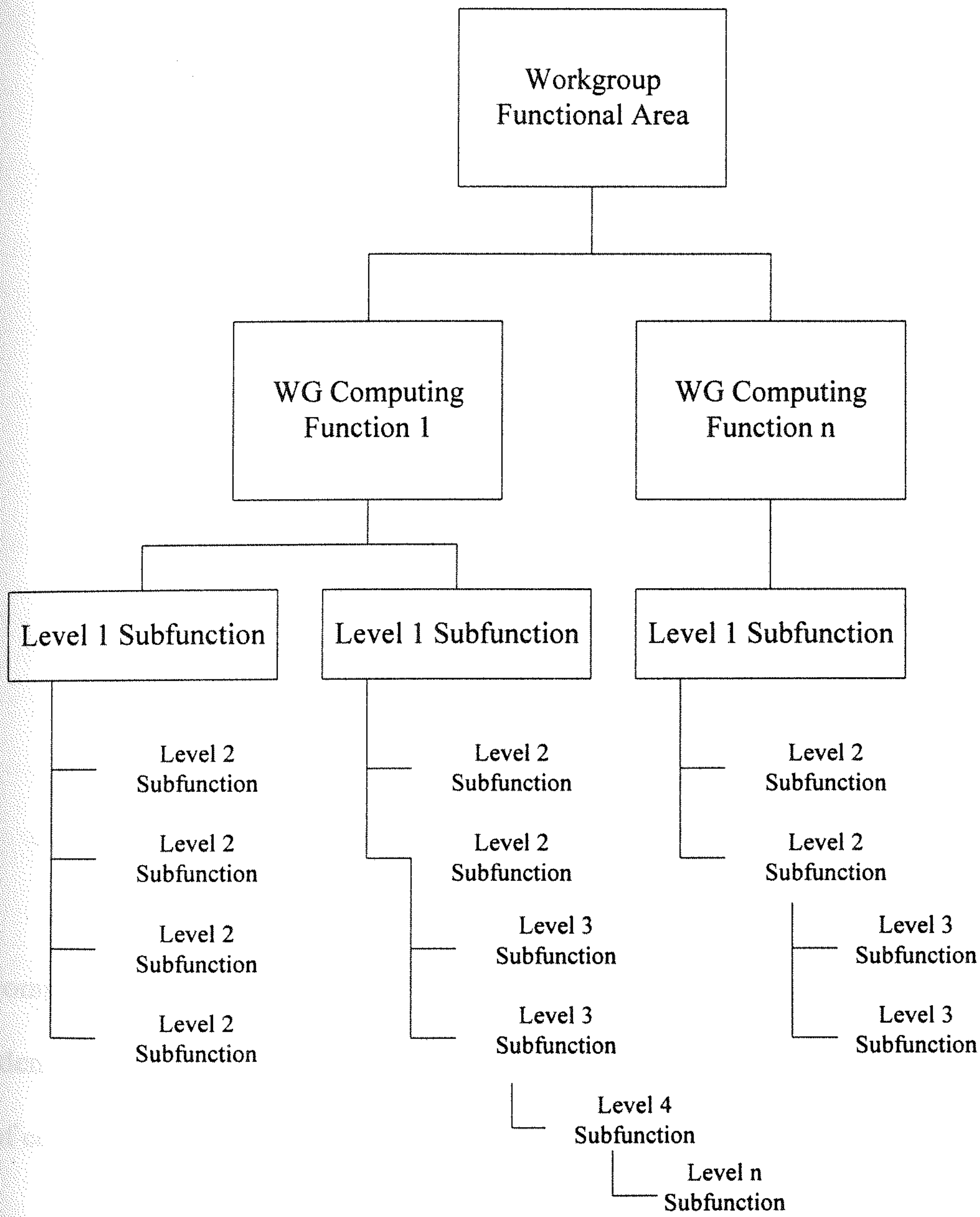

Figure 3. Workgroup computing function hierarchical breakdown.

Enter "User ID" (Level 3 subfunction): Self-explanatory.

Enter "Password" (Level 3 subfunction): Self-explanatory

Logoff workgroup session (Primary function): Users logoff of shared workspace. The system sends a logoff notification message to remaining users.

Gesture on (Primary function): User can use the following gestures on the shared drawing workspace.

Point (Level 1 subfunction): User can point at an object or text.

Write (Level 1 subfunction): User can write a note.

Erase (Level 1 subfunction): User can erase a note.

Direct attention (Level 1 subfunction): User can make a motion to direct attention.

Gesture off (Primary function): Gesturing is turned off.

After all workgroup application areas were reviewed and assessed, the results were compiled into a table that listed the workgroup computing application areas versus all identified workgroup computing tasks and primitives. This table allowed identification of common tasks and primitives that were consolidated into the architecture.

F. *Classification Structure and Architecture Development Methods*

Figure 4 is an application-oriented workgroup computing architecture synthesized from several sources (Bentley, Rodden, Sawyer, & Sommerville, 1994; Pastor & Jager, 1992; Reinhard, Schweitzer, & Volksen, 1994; Roseman & Greenberg, 1992, 1993).

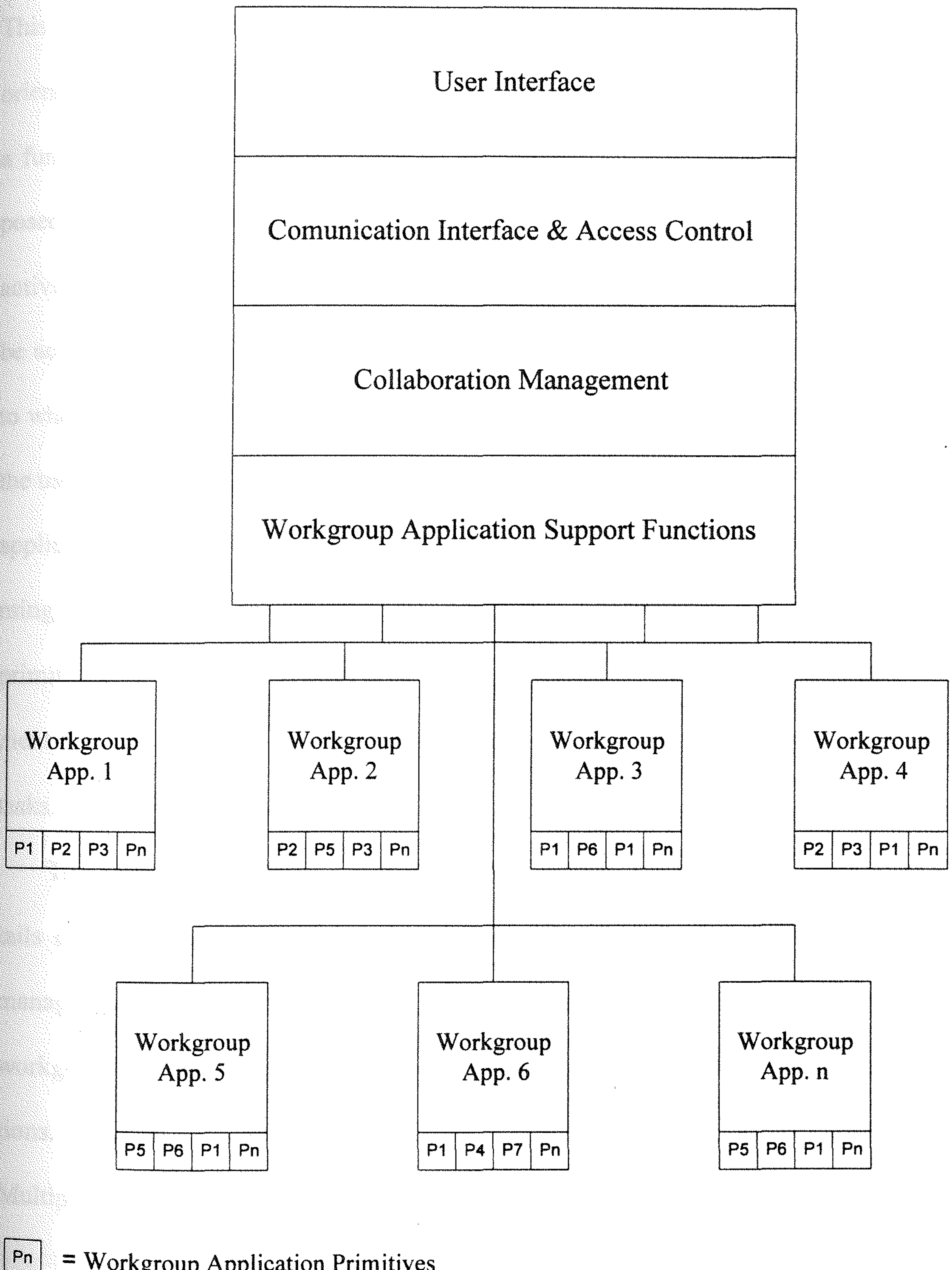

Figure 4. Application-oriented workgroup computing architecture.

This architecture contains the basic elements that must be included in any application-oriented workgroup computing architecture. Figure 5 shows the preliminary structure for a function-oriented architecture. In this architecture, each workgroup function is composed of tasks or primitives that can be selected by a user to perform various workgroup activities. These two architectures are different. In the function-oriented one, work can be accomplished by selecting tasks or primitives that make up functions that correspond to whatever function the user wants to perform. In the applications-oriented architecture, the user must select one or more applications to perform the desired work. These selected applications may contain duplicate tasks and primitives and require duplicate programming code. This study concentrated on identifying the workgroup functions, tasks, and primitives that must be included in the new architecture. In addition, workgroup management functions were identified that support common workgroup support functions or tasks.

The completed workgroup computing taxonomy provided information to define details of the workgroup functions, application programming interfaces, and workgroup management function blocks. Based on a review of the completed taxonomy table, all workgroup functions were classified as workgroup support or management specific functions. Analysis of tasks identified duplicate tasks that existed in each application area. Multiple occurrences were consolidated into a single task that was properly categorized.

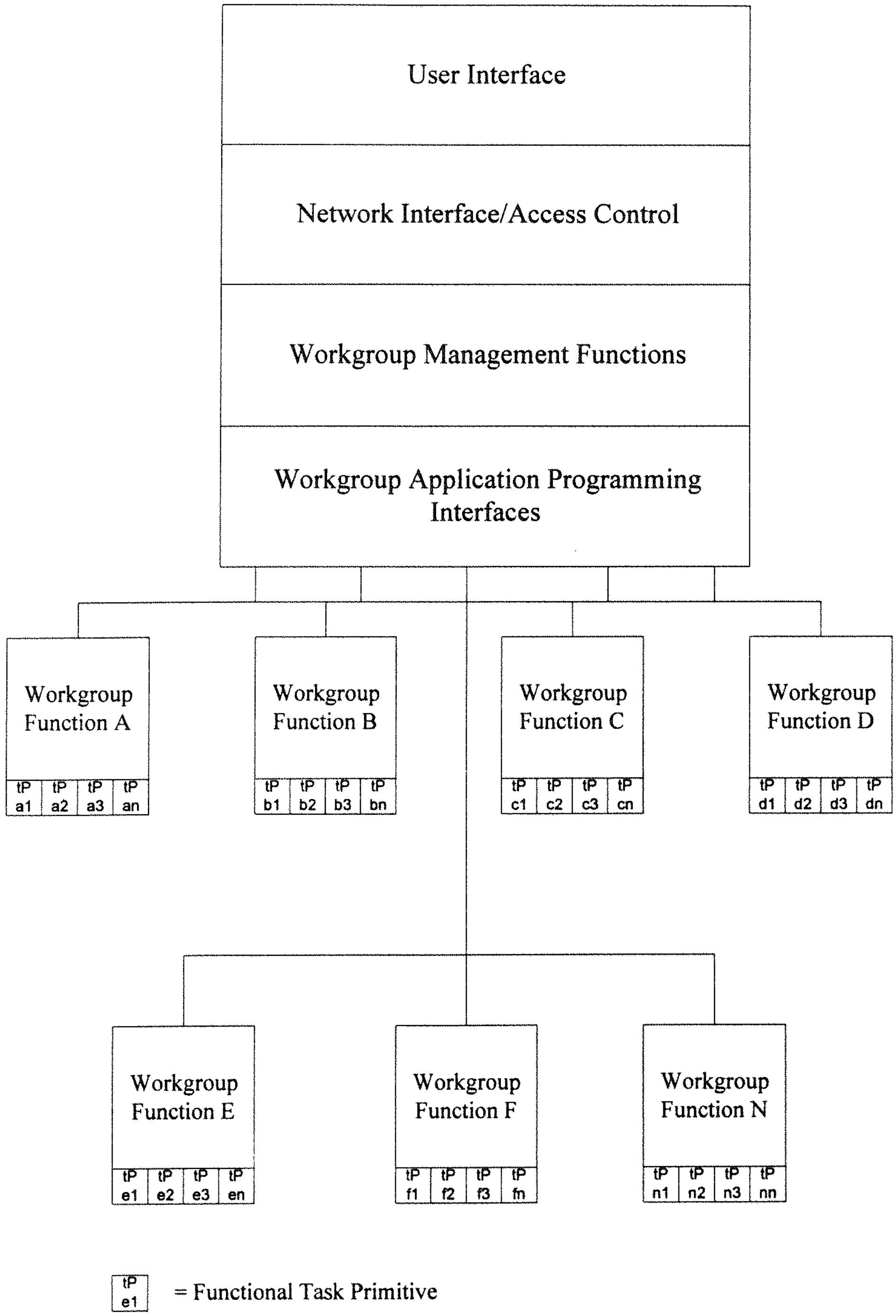

Figure 5. Function-oriented workgroup computing architecture.

A definition of all workgroup common and support functions allowed design of a new architecture that included details of the above elements. The main emphasis of this study was to provide details in the common workgroup support and management areas.

Standard network and user interface elements extracted from the literature were also included. The user interface is either Windows 3.1 or Windows 95. Both Windows 3.1 and Windows 95 are structured to wait until a program receives a message from an application to execute a task. Windows functions, or APIs, are used to interface with the Windows user interface to make the task happen for a program. Workgroup computing APIs will cause workgroup functions to occur when they are called by the workgroup program (Conger, 1992).

The architecture includes a hierarchy of protocols that provide networking interfaces. Figure 6 illustrates this networking protocol hierarchy. The physical layer defines the electronic, radio frequency, or optical transmission medium and the manner in which bits are signaled on the medium. The network layer determines how sequences of bits are framed into chunks. The network layer deals with packet communication and it is the lowest computer communication level (Schnaidt, 1992).

The transport layer provides congestion control between communicating programs. Some applications require that data be delivered in sequence and with high reliability. Others only require that data be delivered quickly. Some information can be lost. The layers above transport are closer to the application functions and must reflect their needs.

Application	Detailed information about data being exchanged
Presentation	Conventions for representing data
Session	Management of connections between programs
Transport	Delivery of sequences of packets
Network	Format of individual data packets
Data Link	Access to and control of transmission medium
Physical	Medium of transmission

Figure 6. Network communication layers.

Associations between communicating programs are established at the session layer. A networking protocol was selected that satisfied the above requirements. Two candidates were the Windows or Netware networking protocols. (Schnaidt, 1992).

Users at all locations must be satisfactorily identified. Security was implemented using passwords and digital cryptographic systems. A user will be able to proceed only if their password matches the password listed in the system file or the password listed in the logon file for the specified workgroup session (Russell & Gangemi, 1991). In addition, an RSA algorithm was implemented to provide message authentication and inadvertent access to data files. Use of the public-key RSA algorithm allows users to digitally sign messages (Russell & Gangemi, 1991). Encryption selections can be made at the discretion of individual users or workgroup session leaders (Russell & Gangemi, 1991).

Users and workgroup session leaders can specify who may access and edit individually owned or specified workgroup files or programs. Access rights can be granted on a global basis or to individual users. Three types of access rights will be supported: read, write, and execute. Read allows users to read a file they have access to. Write allows users to write to a file they have access to. Execute applies to programs and to users that are authorized to start workgroup sessions. If a user is granted permission to execute a program they can start a workgroup session (Russell & Gangemi, 1991).

G. ***Workgroup Computing API Specification Development Methods***

A review of the detailed taxonomy and comprehensive architecture provided the basis for new workgroup computing API functions. Using the format for the Windows API, new API descriptions and specifications were developed for each new workgroup task, primitive, or common workgroup support function included in the taxonomy and architecture. Applicable workgroup APIs that existed in the literature were not rewritten; however, they were listed and the source cited in the study.

The Windows API format was chosen because the author is more familiar with the Windows interface. If required, the Windows workgroup API specifications can be converted into the format for OS/2 or any other operating system programming interface format. Developed workgroup computing API specifications can be translated to many different operating systems; however, for each operating system, the specific API structure and programming syntax must be followed.

The following outlines the API specification structure that was used in this study (Conger, 1992; Schildt et al., 1994a, 1994b).

1. API title: This is the title of the workgroup API function. Titles are normally parts of one or more words linked together without spaces that briefly indicate the API function (e.g. StartWGSess).
2. Purpose: Explains how the API function is used to define a workgroup task.

3. Description: Describes what the API does and how it is used in a workgroup context. The description can define a range for the function and state how often the function is called.

4. Uses: Explains when and in what sequence the API is used by the workgroup program.

5. Returns: Defines a value or set of values that is returned to the program as the API function is executed. Returned values can be integers, a previous value, no returned value (a null or void), Boolean True if a value or configuration for a function is determined or successful, or Boolean False if an error or alternate configuration occurs. The API may return a dialog box, a window handle, or a handle to a pop-up menu.

6. See Also: Specifies other API functions that support the defined function (e.g. CloseWGSess)

H. *Summary*

Based on the literature review and analysis, methods were outlined to develop a comprehensive workgroup computing taxonomy that included detailed workgroup computing tasks and primitives. A workgroup computing taxonomy functional structure was developed. This structure was based on previous work completed by the author (Von Worley, 1994), analysis of workgroup computing applications included in the literature, and the earlier discussion of workgroup computing application tasks and primitives. The taxonomy structure included 11 workgroup computing functional areas.

A review of existing workgroup computing taxonomies and architectures indicated that there may be a better way to structure a workgroup computing architecture. If all workgroup functions and primitives could be identified and common elements consolidated for both application and support functions, an architecture could be specified that would allow an individual to select desired functions or tasks.

The taxonomy provided sufficient information to develop an integrated workgroup computing architecture based on workgroup computing support and management tasks. The main emphasis of this study was to provide details in the common workgroup support and management areas. This architecture will allow new or different types of work functions to be accomplished within the existing structure by linking together basic tasks or primitives.

The taxonomy and architecture structure formed the basis for the workgroup computing API specifications for workgroup support and management functions. These API specifications will allow the integrated architecture to be implemented using either Windows 3.1 or Windows 95. If required, new APIs can be developed to easily add profoundly different tasks that did not exist when the architecture was developed.

Chapter V. Results

A. *Workgroup Computing Taxonomy*

Using the specified methods, a detailed section of the workgroup computing taxonomy was synthesized for each of the 11 workgroup computing functional areas shown in Figure 7. The analysis did not reveal any additional workgroup computing functional areas. Review and analysis of technical literature, product literature, and application manuals for each workgroup computing functional area, in conjunction with the workgroup applications review, provided sufficient source material to derive workgroup computing tasks and primitives. To provide clarity or continuity within a section, some non-workgroup tasks were included in the taxonomy.

For each functional workgroup area, only workgroup primitives that were identified in the source material were included in the taxonomy. Although a task or primitive could exist in another related section of the taxonomy, if a task was not identified in any of the sources for a particular workgroup computing functional area, then the task or function was not included in the taxonomy for that section. An example is encryption. This function was only found during the analysis for E-mail, interactive communication, and workflow management taxonomy sections. If data security and privacy are issues, encryption can be applied to any workgroup functional area. In the architecture section, common workgroup management or support functions have been extended to each work-

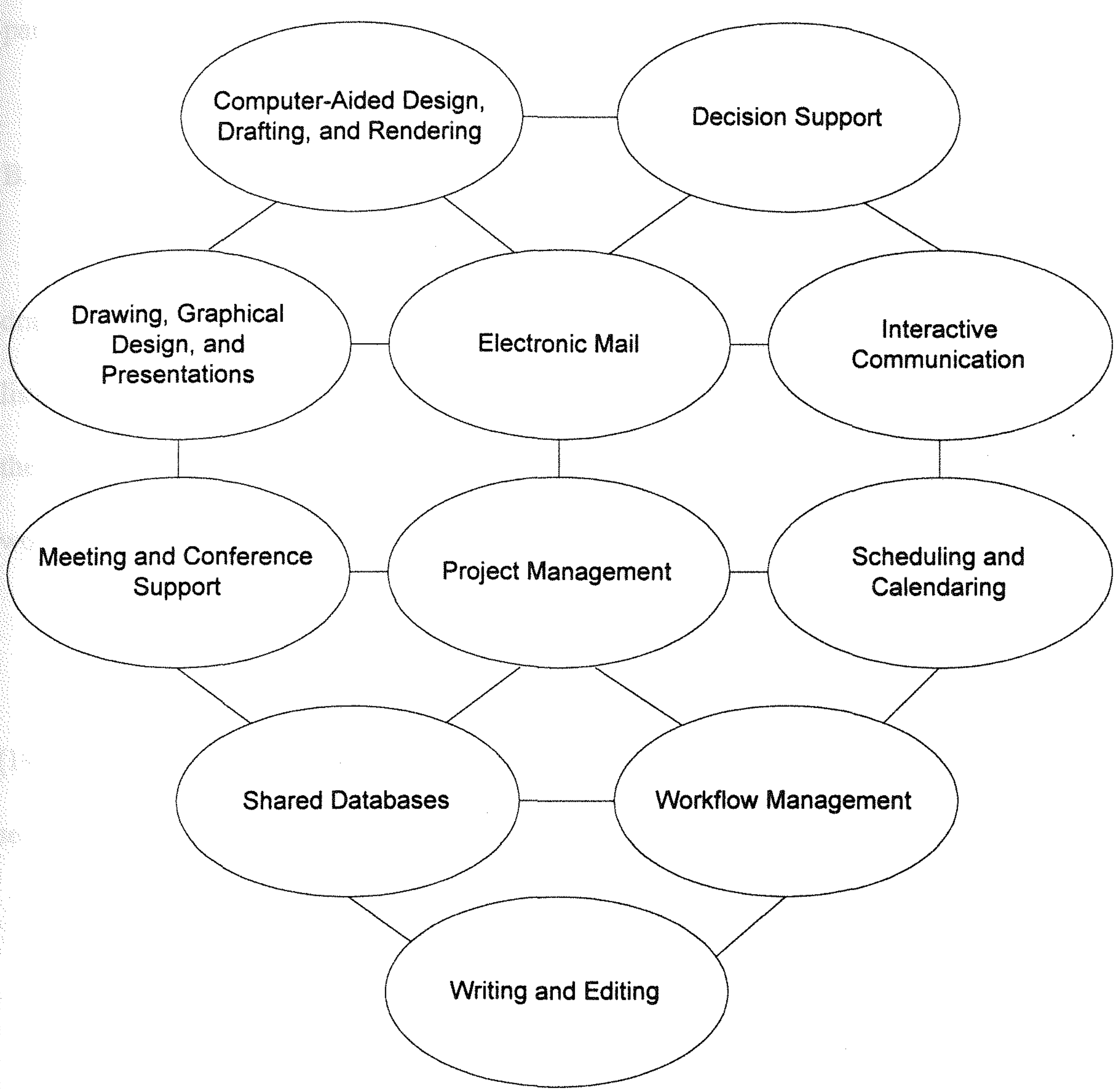

Figure 7. Workgroup computing functional areas used to develop the taxonomy.

group computing functional area. Citation of source material is included in the lead-in paragraph for each section of the taxonomy.

B. *Workgroup Computing Taxonomy Examples*

As examples, the following are portions of the taxonomy for computer-aided design, drafting, and rendering; interactive communication; and writing and editing. Appendix A lists the complete workgroup computing taxonomy without annotation. Appendix B lists the complete workgroup computing taxonomy with annotations that explain each workgroup computing task or primitive.

Computer-aided design, drafting, and rendering.

Glickman and Kumar (1993), Goetsch (1986), Kyng (1991), and Shu and Flowers (1992) provided source material for the computer-aided design, drafting, and rendering section of the workgroup computing taxonomy.

Design session setup

- Start workgroup session
  - Unrestricted
  - Restricted
    - Define access list
      - Enter "User IDs"
  - Change access
    - Add "User ID"
    - Delete "User ID"
  - Send invitation
  - Request user information
- Logon to workgroup session
  - Enter "User ID"
  - Enter "Password"
- Logoff workgroup session
- Request to join group

- Approval to join group
  - Join
  - Do not join
- Change session leader
- Editing Mode
  - Designated
  - Baton mode
  - First-come, first-served
  - Free
- Gesture on
  - Point
  - Write
  - Erase
  - Direct attention
  - List workgroup notes
- Gesture off
- Status information
  - Who is on-line
  - Topic
  - Group leader
  - Who is in control
- Audio/Video on
- Audio/Video off
- Voice mute
- Video mute

File

- New
- Open
- Layout
  - Single window
    - Specify drawing size
  - Multiple windows
    - Specify number
    - Specify drawing size
  - Work in
    - 2D
    - 3D
    - Layers
      - Add layer
      - Delete layer
- Recall Session
- Merge Session

Split session

Save as

Save current snapshot

Retrieve a snapshot

Save session

Archive session

Compression on

Compression off

Import file or object

Export file or object

Import private view into workspace

Drawing setup

Draw

All session design objects have the following:

Send description

Coupling status

Owner ID

Acquisition status

Request change status

Do change status

Request change owner

Do change owner

Request to grab

Do grab

Where grabbed

End grab

Owned objects list

Text

Type text

Shared workspace text

Request select text

Do select text

Change text

Arrange

Clear workgroup surface

Display

Access private view

Move to empty workgroup view

Move to common workgroup view

Show virtual position

Interactive communication.

Benest and Dukic (1993), Dourish and Bellotte (1992), Ohkubo and Ishii (1990), Stefik et al. (1987), Vin and Chen (1992), and Rodden (1993) provided source material for the interactive communication section of the workgroup computing taxonomy.

Communication session setup

Start communication session

- Unrestricted
- Restricted
  - Define access list
    - Enter "User IDs"
- Change access
  - Add "User ID"
  - Delete "User ID"
- Assign roles
  - Organizer
  - Primary user
  - Contributor
- Send invitation

Logon to workgroup session

- Enter "User ID"
- Enter "Password"

Logoff workgroup session

Request to join group

Approval to join group

- Can join
- Cannot join

Change leader

Communication mode

- Text only (synchronous)
- Multimedia session (synchronous)
- Message (synchronous or asynchronous)

Editing Mode

- Designated
- Baton mode
- First-come, first-served
- Free

Gesture on

Point

Write

Erase

Direct attention

List workgroup notes

Gesture off

Status information

Who is on-line

Topic

Group leader

Who is in control

Audio/Video on

Audio/Video off

Voice mute

Video mute

Merge Session

Split session

Save current snapshot

Retrieve a snapshot

Save session

Archive session

Encryption on

Encryption off

Import private view into workspace

File transfer

Send file or object

Receive file or object

Import file or object

Export file or object

Link to E-mail

Drawing/sketching

All session drawing objects have the following:

Send description

Coupling status

Owner ID

Acquisition status

Request change status

Do change status

Request change owner

Do change owner

Request to grab

Do grab

Where grabbed

End grab

Owned objects list

Text

Type text

Shared workspace text

- Request select text
- Do select text
- Change text

On-line decision making

Poll the group

Obtain group consensus

Obtain group vote

Task assignment

- Task
- Responsibility
- Deadline
- Send task

Access database

Writing and editing.

Baecker et al. (1993), Baydere et al. (1993), Denley et al. (1993), Knister and Prakash (1990), Neuwirth et al. (1990), Posner and Baecker (1992), Sharples (1993), Sharples et al. (1993), Shen and Dewan (1992), and Sobiesiak and Myopoulos (1991) provided source material for the writing and editing section of the workgroup computing taxonomy.

Writing and editing workgroup setup

Define writing/editing workgroup

- Document name
- User name

Session control

Roles

- Lead writer
- Co-writer
- Writer
- Consultant
- Editor
- Reviewer

Activities

- Brainstorm
- Research
- Initial plant
- Write
- Control changes
    - Edit document
    - Final edit
- Review

Writing strategies

- Single writer
- Scribe
- Separate writers
- Joint writing

Access control

Document control

- Centralized

- Relay
- Independent
- Shared

Permissions

- User ID
- Read/write
- Comment
- Read only
- Change access

Logon to writing system

- Enter "User ID"
- Enter "Password"

Logoff writing system

Request to join writing group

Approval to join conference

- Can join
- Cannot join

Sessions

- Create session
- Merge session
- Leave session

Document segmentation

- Separate document sections
- Join document sections

Writing rules

Style rules

Content rules

Editing rules

- Lead writer select rules
- Editor select rules
- Both select rules

Revision control

Small change

Many additions

Delete

Change history

Version control

- Versions allowed
  - Single
  - Multiple
- Parallel
- Sequential

Reciprocal

Automatic conflict extraction

Present to lead author

Present to all authors

Drawing/sketching

All session drawing objects have the following:

Send description

Coupling status

Owner ID

Acquisition status

Request change status

Do change status

Request change owner

Do change owner

Request to grab

Do grab

End grab

Owned objects list

Status monitoring

Revision status

Tools

Import text or objects

Export text or objects

Information filtering

Send E-mail message

Audio/video

Audio/video on

Audio/video off

Video channel mode

Multiple image mode

Single image mode

Preview

Audio select

All channels

Selected channel

Select

Selected video channel audio only

Audio mute

Video mute

Open multimedia conference

C. *Workgroup Computing Common Tasks and Primitives*

Figure 8 summarizes common workgroup computing tasks and primitives that are included in each workgroup computing functional area. Task and primitive groupings have been divided into workgroup computing management and workgroup computing support categories. Black rectangles show tasks and primitives that were identified in reference sources. White rectangles show where tasks and primitive groupings can be extended to a currently unsupported workgroup functional area. As previously mentioned, tasks and primitives for these functional areas were not identified in any reference source. Appendix C provides a detailed breakdown for each task/primitive group.

D. *Workgroup Computing Architecture*

Workgroup computing taxonomy tasks and primitives were sorted into workgroup support and management interface groupings. Appropriate tasks and primitives were extended to unsupported workgroup computing functions. This structure formed the basis for the workgroup computing architecture shown in Figure 9. The architecture includes all the workgroup management and workgroup support functions that were defined in the workgroup computing taxonomy. A windows-based operating system, external security, and networking functions were also included.

Workgroup management interfaces allow the workgroup leader and group members to manage workgroup sessions and tasks. The workgroup session leader can initiate a workgroup session and configure it by selecting from various communication modes. Workspaces can be displayed or arranged in tailored views. Sessions can be combined or

Workgroup Computing Functional Area / Workgroup Computing Task/Primitive Group	Computer-aided design	Drawing and presentations	Decision support	Electronic mail	Interactive communication	Meeting/conference support	Project management	Scheduling and calendaring	Shared databases	Workflow management	Writing and editing
Workgroup Management											
Session Setup	■	■	■	■	■	■	■	■	■	■	■
File/Session Management	■	■	■	■	■	■	■	■	■	■	■
Display/Arrange Views	■	■	❑	❑	❑	❑	❑	❑	❑		❑
Communication Mode	■	■	■	■	■	■	■	■	■	■	■
Other Tasks	■	■	❑	■	■	❑	❑	■	■	■	❑
Workgroup Support											
Decision Tools	❑	❑	■	■	■	■	■	❑	❑	■	❑
Drawing/Graphics	■	■			■	■	■	❑		■	■
Text	■	■	❑		■	■	❑			■	■
Writing/Editing	■	■	❑		❑	■	❑	■	■	■	■

Workgroup Computing Functional Area / **Workgroup Computing Task/Primitive Group**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Databases	❑	❑	■		■	■	❑		■	■	
Mail			❑	■	■	■	■	❑	❑	■	■
Calendar/Scheduling			❑		❑	■	■	■	❑	❑	❑
Ideas	❑	❑	■		❑	■	❑			❑	❑
Workflow	❑	❑	❑	❑	❑	❑	❑	❑	❑	■	❑

Figure 8. Workgroup computing functional areas showing the applicability of workgroup computing tasks and primitives and their extension to unsupported areas.

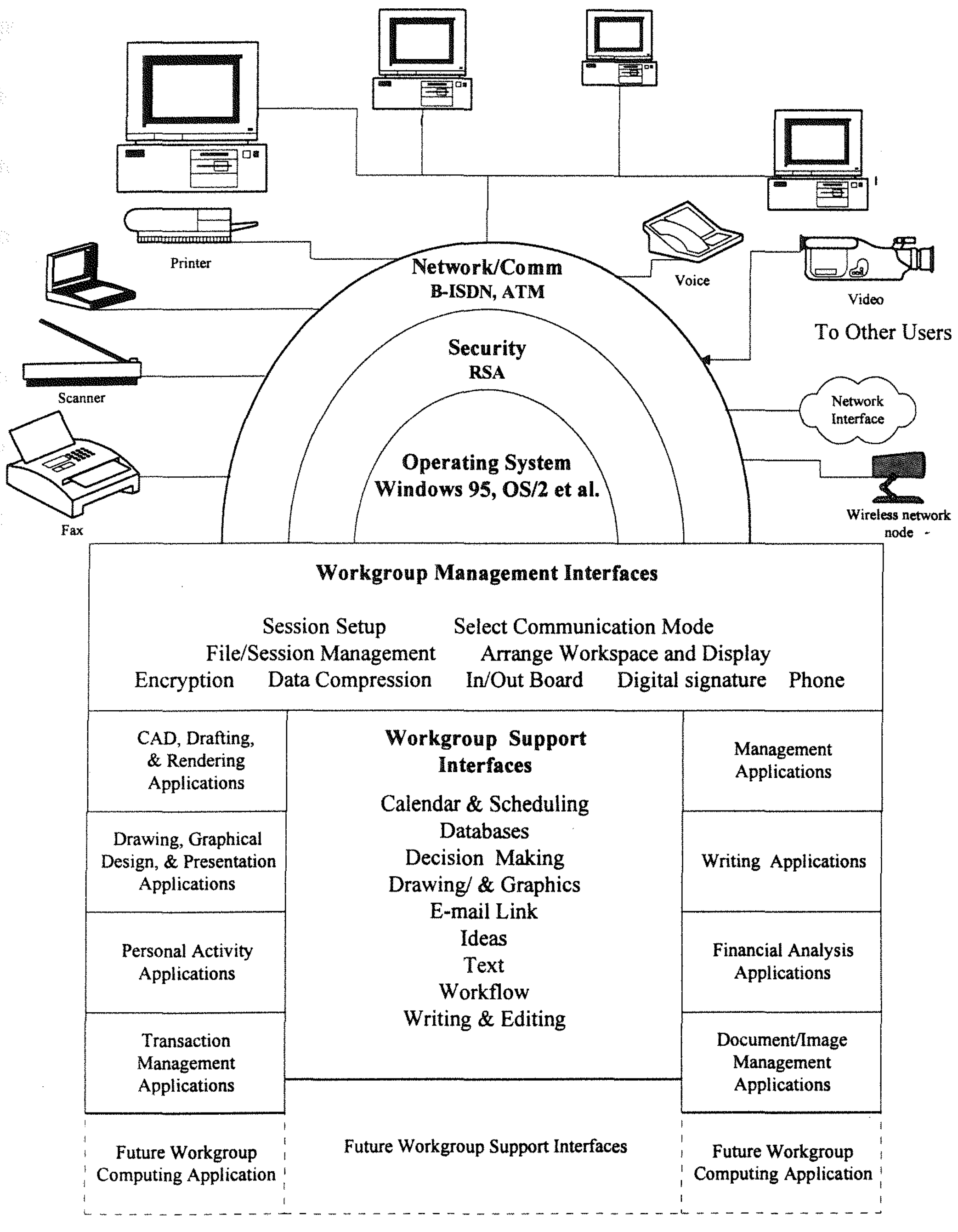

Figure 9. Workgroup computing architecture showing workgroup computing management and support interface functions.

split. Sessions can also be saved, archived, or replayed. In addition, encryption and data compression for files or sessions are available to the session leader or workgroup members. A digital signature can be added to a document or form. Telephone calls can be made from within the workgroup session and an In/Out board can provide status of the workgroup members.

Workgroup support interfaces allow the workgroup members to perform common workgroup production functions. They are (a) group calendar and scheduling functions, (b) database access and management, (c) decision making tools, (d) drawing and graphics support, (e) E-mail links, (f) idea generation and management, (g) minor text input and editing, (h) workflow management, and (i) full document writing and editing functions. The architecture allows additional functions to be added when they become available.

Workgroup applications functions form the lowest level of this architecture. These applications perform the work or activity production functions that have no common work-group link. Applications are (a) CAD, drafting, and rendering, (b) drawing, graphical design and presentation, (c) personal activity management, (d) transaction management, (e) management, (f) writing applications, (g) financial analysis, and (h) document and image management. If necessary, additional applications can be added to the architecture.

The operating system for the architecture is either Windows 3.1 or 95. Another windows-based operating system such as OS/2 can also be implemented. External and internal data security and digital authentication are provided by the RSA encryption algorithm. The networking interface is provided by B-ISDN and ATM. B-ISDN and ATM protocols were selected to provide the network speed and bandwidth required for video, multimedia, and parallel communication channels. A variety of input and output devices, such as facsimile, printer, video, telephone, and a scanner, are attached to the workgroup network. These devices permit text, graphics, audio, or video to be input or output in either electronic or hardcopy format.

E. *Application Programming Interface (API) Specifications*

Workgroup computing APIs are called by the Windows operating system when an application needs to execute a particular workgroup computing management or support function. One or more workgroup computing API specifications were written for each workgroup computing management or support interface function that was identified in the taxonomy. Appendix D is a complete listing of all workgroup computing API specifications for tasks and primitives identified in the taxonomy.

Workgroup computing APIs are divided into two groups: workgroup management and workgroup support. API groupings for workgroup management include (a) session setup, (b) file/session management, (c) display/arrange views, (d) communication mode, and (e) other tasks including encryption, data compression, In/Out board, and a digital signature. Workgroup support contains API groupings for (a) decision tools, (b) drawing/graphics, (c) text, (d) writing/editing, (e) databases, (f) E-mail, (g) calendar/scheduling, (h) ideas, and (i) workflow support.

Each API specification includes a purpose, description, uses, the value the API returns when it is executed by an application, and other related APIs. "Returns" defines a value or set of values that is returned to the program when the API function is executed. Returned values can be integers, a previous value, no returned value (a null or void), Boolean True if a value or configuration for a function is determined or successful, or Boolean False if an error or alternate configuration occurs. Additionally, an API may return a dialog box, a file handle, a window handle, or a handle to a pop-up window or menu (Conger, 1992).

Workgroup computing API examples.

As examples, the following are API specifications for parts of two workgroup computing task groupings: Session Setup and Writing and Editing.

SESSION SETUP

Define session files

CreateSessionFile

Purpose: Open or create a workgroup session file.

Description: This function opens an existing workgroup session file. If necessary, it will create a new file.

Uses: Normally used to open a workgroup file related to a specific workgroup session.

Returns: If valid, this function returns a handle to the session file. If an error occurs, it returns an invalid handle value.

See Also: *CloseSessionFile*

Start/connect session

WGSessionOpen

Purpose: Start a workgroup session.

Description: This function starts a workgroup session defined by workgroup leader.

Uses: Normally used to initiate a specific workgroup session.

Returns: If valid, this function returns a handle to the workgroup session. It returns a Null on error.

See Also: *WGSessionClose*

WGSessionClose

Purpose: End a workgroup session.

Description: This function ends a workgroup session.

Uses: Normally used to terminate a specific workgroup session.

Returns: If the workgroup session is closed, this function returns Boolean True. It returns Boolean False if the workgroup session does not close.

See Also: *WGSessionOpen*

Change access

ChangeAccess

Purpose: Change the workgroup session access list.

Description: Allows the workgroup leader to change access to the workgroup session by adding or deleting User IDs or E-mail addresses.

Uses: Used to change and control access to the workgroup session once the session has been setup.

Returns: This function returns a dialog box that allows the workgroup leader to add or delete specific user IDs or E-mail addresses.

See Also: *DefineAccessList*

Assign roles

AssignWGRoles

Purpose: Assign user roles for workgroup activities.

Description: Allows the workgroup leader to specify user roles for various workgroup activities. Normally, this functions permits the workgroup leader to specify various users different privileges or access to workgroup files.

Uses: Used to define user roles for writing, decision making, project management, or other workgroup functions.

Returns: This function returns a dialog box that allows the workgroup leader to specify user roles for each specific workgroup activity where a role is required.

See Also: *ChangeWGRoles*

ChangeWGRoles

Purpose: Change user roles for workgroup activities.

Description: Allows the workgroup leader to change user roles for various workgroup activities either based on his prerogative or a request from a user. Normally, this functions permits the workgroup leader to change user privileges or access to work-group files.

Uses: Used to change user roles for writing, decision making, project management, or other workgroup functions.

Returns: This function returns a dialog box that allows the workgroup leader to change user roles for any specified activity.

See Also: *AssignWGRoles*

Workgroup support API examples.

WRITING/EDITING

Activities

DefineWGWritingActivity

Purpose: Define workgroup writing activity sequence.

Description: Allows the lead writer to sequence or segment the writing activity into brainstorming, research of ideas and issues, drafting an outline, writing a document, or use of an open style without boundaries.

Uses: Normally used by the lead writer to structure writing tasks into discrete segments and sequences. Can also be used to define an open writing style that allows any type of input.

Returns: This function returns a dialog box that allows mode selection and an integer, the selected writing activity. It returns a Null on error.

See Also: *DefineWGWritingStrategy*

Strategies

DefineWGWritingStrategy

Purpose: Define workgroup writing strategy.

Description: Allows the lead writer to define how the writing will take place. Options include a single writer who writes most of a document while other writers assist, a scribe who writes down most of a group's thoughts and comments while other writers discuss the ideas that will be expressed in a document, separate writers where a document is broken down into sections that are authored separately, and joint writing where several group members compose the text together.

Uses: Normally used by the lead writer to define how a document will be written.

Returns: This function returns a dialog box that allows mode selection and an integer, the selected strategy. It returns a Null on error.

See Also: *DefineWGWritingActivity*, *DefineWGDocumentControl*

Document control

DefineWGDocumentControl

Purpose: Set up how the document can be accessed depending on the workgroup writing strategy.

Description: Determines how a document will be controlled depending on the selected writing strategy. Control can be centralized with only one writer, relayed between users, independent with document sections accessed by only one writer at a time, and shared simultaneous access.

Uses: Used to control access to a document or its sections based on the chosen writing strategy.

Returns: This function returns a dialog box that allows mode selection and an integer, the document control strategy. It returns a Null on error.

See Also: *DefineWGWritingActivity*

Permissions

DefineWGDocumentPermissions

Purpose: Set document access permissions.

Description: Allows the workgroup session leader or lead writer to designate document access levels by User ID. Access can be read/write, comment, or read only. Access permission can be changed.

Uses: Used to control access to a document or its sections based on user ID.

Returns: This function returns an integer, permission, for each affected user ID. During user logon, if a match is successful, this function returns Boolean True and appropriate access is granted. It returns Boolean False if a match does not occur for a User ID and no access is allowed.

See Also: *DefineWGWritingActivity*

Writing rules

DefineWGWritingRules

Purpose: Define workgroup writing rules.

Description: The lead writer defines workgroup style and content rules. Either the lead writer, editor, or both define workgroup editing rules.

Uses: Normally used to define style content and editing rules.

Returns: This function returns an integer for each style, content, or editing rule for each affected document.

See Also: *None*

Revision control

DefineWGDocumentRevision

Purpose: Control how document revisions are made.

Description: Describes the types of document revisions and how they are handled by the system. Examples are a small document change where the change is made without marking it, many additions where a change is marked so that it stands out from the other text and the responsible writer makes the changes, and text marked for deletion where the responsible writer approves the deletion. Document change history is also displayed.

Uses: Used to control who approves and makes document changes.

Returns: Below a certain specified error limit the document is automatically revised by the system. For more extensive changes, the system marks the document, matches the user ID for the portion of the document written by the user with the user ID of the document changer. If a match occurs, then the document is changed. If a match does not occur the responsible writer is notified and prompted to approve the change.

See Also: None

Version control

DefineWGVersionControl

Purpose: Select how many document versions are allowed.

Description: Allows the lead writer to define single version or multiple version documents and arbitrate writing conflicts with the lead author or all authors.

Uses: Used to define allowable document versions and resolve writing conflicts.

Returns: This function returns a dialog box allowing the lead author to select either single or multiple document versions. It returns a Null on error.

See Also: None

Post-meeting documents

CreateWGMeetingDocument

Purpose: Prepare post-meeting documents based on the meeting session.

Description: Allows the meeting/conference chairperson to select an output for the specified meeting. Selections include an electronic format meeting/conference transcript, complete hardcopy transcript, a meeting/conference snapshot report, a summary report, or a highlights report.

Uses: Used by the meeting/conference chairperson to document meeting and results.

Returns: This function returns a dialog box that allows the meeting/conference chairperson to select a format for the meeting/conference output. It returns a Null on error.

See Also: None

Video editing

EditWGVideo

Purpose: Edit workgroup video.

Description: Allows the workgroup session leader or designated users to interactively edit video clips or segments.

Uses: Used to edit workgroup video segments.

Returns: This function returns a dialog box that allows the workgroup session leader or designated users to select a video file and video editing tools. Returns a handle to the specified video file. It returns a Null on error.

See Also: None

F. *Summary of Results*

Based on a detailed analysis of workgroup computing applications, a comprehensive workgroup computing taxonomy was derived for 11 workgroup application areas. The analysis did not reveal any additional workgroup computing functional areas. Review and analysis of technical literature, product literature, and application manuals for each workgroup computing functional area, in conjunction with the workgroup applications review, provided sufficient source material to derive workgroup computing tasks and primitives. To provide clarity or continuity within a section, some non-workgroup tasks were included in the taxonomy.

Using the detailed taxonomy, an integrated workgroup computing architecture was developed based on common workgroup management and support functions. Workgroup computing taxonomy tasks and primitives were sorted into workgroup support and management interface groupings. Appropriate tasks and primitives were extended to un-supported workgroup computing functions.

Specifications were written for 86 workgroup computing APIs that will allow the integrated architecture to be implemented in a windows-like operating environment. Workgroup computing APIs are divided into two groups: workgroup management and workgroup support. Each API specification includes a purpose, description, uses, the value the API returns when it is executed by an application, and other related APIs.

Chapter VI. Discussion, Implications, and Recommendations

A. *Support for the Hypothesis*

The results of this study supported the hypothesis that the available workgroup computing literature and workgroup applications' documentation would provide sufficient information to develop a workgroup computing taxonomy. The developed taxonomy comprehensively covers 11 workgroup computing functional areas by providing details down to the workgroup task or primitive level.

By comparing workgroup tasks with workgroup computing functional areas, it was possible to derive a common set of workgroup computing management and support tasks that were based on the comprehensive workgroup computing taxonomy. These common workgroup management and support tasks formed the basis for an integrated workgroup computing architecture. In addition, it was possible to write new workgroup computing API function specifications for common workgroup computing management and support tasks.

B. *Workgroup Computing Taxonomy*

Sufficient information was available in the reference sources to develop a section of a workgroup computing taxonomy for each of the 11 workgroup computing functional areas. The analysis did not reveal any added workgroup computing functional areas and no areas had to be combined or divided. Each taxonomy section includes all the tasks and primitives that could be extracted from the available reference sources. A task was included in the taxonomy if it existed one time in any literature or application source for a functional area. The complete comprehensive taxonomy, included in Appendix A, was formed from the consolidation of the 11 taxonomy sections. Every one of the functional areas provided a contribution to the complete workgroup computing taxonomy.

In a number of instances, necessary tasks or primitives appeared to be overlooked by all sources in a functional area. In these instances, the missing task or primitive was not included in that particular section of the taxonomy. Missing elements were addressed in the common workgroup computing tasks and primitives analysis. The "send invitation" and "respond to invitation" primitives are examples. "Send invitation" existed in 5 of 11 functional areas; however, "respond to invitation" was only found in 2 out of the 5 areas that included "send invitation."

The analysis also confirmed the earlier statement that no single groupware product includes all workgroup computing functions. As discussed in the literature review, work-group researchers tend to build their applications to address a narrowly defined set of workgroup issues. Each application normally includes its own set of relevant functions that allow the application to perform a necessary activity such as meeting or conference support, scheduling, or project management. The taxonomy allowed identification of common workgroup tasks and primitives.

Current research-oriented workgroup applications for functions such as scheduling, writing, and drawing are aimed at satisfying the needs of only a single workgroup application area. Commercial applications such as Lotus Notes include several, but not all, workgroup functions. This study provides a detailed set of workgroup computing tasks and primitives that can help define totally integrated research or commercial workgroup computing products of the future. Implementation of the taxonomy and architecture will allow development of new workgroup applications such as distance voting, home shopping, hypermedia commenting, and intelligent optimization of group activities. There will also be no need to program workgroup computing support and management functions for every application instance.

C. *Workgroup Computing Common Tasks and Primitives*

Repetitive or missing workgroup tasks and primitives were identified by creating a matrix showing workgroup tasks and primitives versus workgroup computing functional areas. Results showed there was significant repetition of functions in the session setup and file management areas. There was moderate repetition of functions in the communication mode, mail, drawing/graphics, decision tools, and database areas. In contrast, there was little repetition of functions in display/arrange views, writing and editing, calendar and schedule, idea generation, and workflow management. In the past, this last set of functions has been developed to support a narrow range of applications. When placed in a common framework these functions can support other workgroup activities.

If a task existed in several functional areas or could be extended by inference to one or more additional functional areas then it became a common workgroup computing task function. Tasks groupings most suitable for other workgroup functional areas include display/arrange views, decision tools, databases, calendar and scheduling, ideas, and workflow management. This exercise resulted in sets of common workgroup computing tasks that could be included in a workgroup computing architecture. These common task groupings were sorted into two types: workgroup management and workgroup support. They formed the basis for the workgroup computing architecture and the API specifications.

D. *Workgroup Computing Architecture*

The developed workgroup computing architecture consists of four main elements. Figure 10 shows these elements and their relationship to one another. The architecture provides a common style of interface for all workgroup activities that permit a user to access shared functions. A user can adapt each workgroup session to a specific set of workgroup styles. Since all workgroup functions are defined and accessible by all users, complex, multiple-session workgroups can be supported.

To allow the workgroup to function efficiently, the architecture includes all necessary workgroup computing management and support elements. The workgroup environment can be configured by workgroup leaders or users to meet individual user needs or support organizational change. Users have the ability to select appropriate functions that

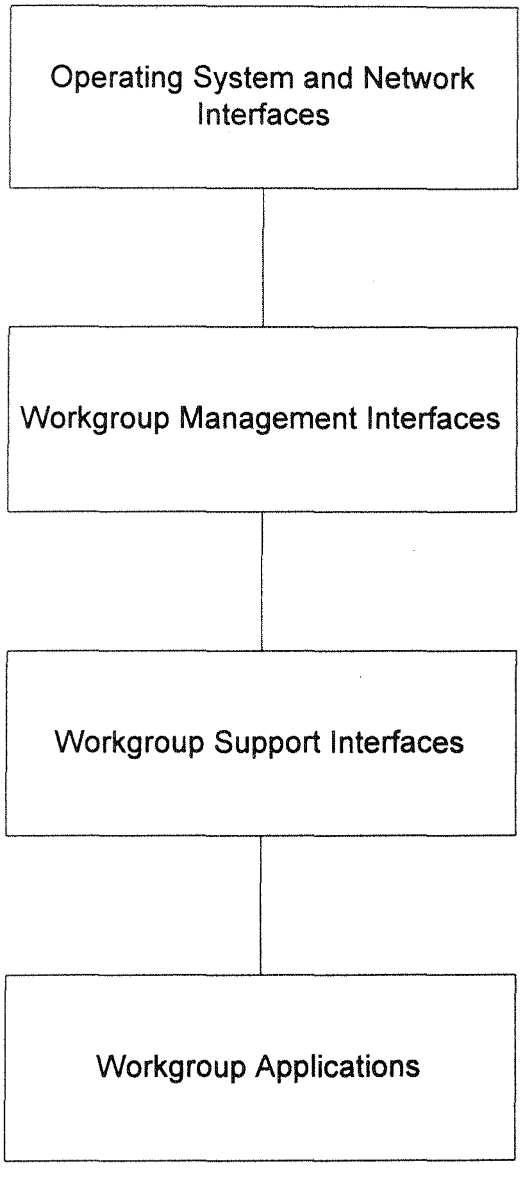

Figure 10. Workgroup computing architecture elements showing their relationship.

will tailor the system configuration to support any work task or activity. Workgroups and products can be integrated by combining workgroup sessions. Most workgroup session leaders designate user roles for some activities. It will be important for user roles to change as the workgroup consolidates or its goals change. In addition, user roles can change depending on how socially defined group roles develop. The architecture provides concurrent access to shared resources and it allows maximum use of these shared common functions. Various media are used including voice, text, graphics, gesturing, video, audio, and multimedia. Multiple media can be used on parallel channels to support and increase group interaction.

As user needs change, the architecture provides for both private and shared workspaces. Workgroup management allows the session leader or other designated users to configure and control the workgroup environment while allowing individual workgroup members flexibility in configuring their private workspace. Individuals can interact with or be isolated from the workgroup. A user need not be involved with the workgroup aspects of the architecture to perform private work. Should it be necessary to perform shared work the capability to request to join an existing workgroup is available in the architecture. In addition, this configuration allows the development of standard working practices and mechanisms that can be used by different workgroups.

Communication modes can be selected to provide a wide range of communications from text only to full multimedia connections. Communication modes including voice, video, audio, and multimedia provide an interface that is more like face-to-face meetings or work situations. These modes facilitate the exchange of information between individuals or groups by providing many parallel modes of communication. Users can add or remove communication functions to support workgroup goals. This gives the user maximum control over the workgroup environment.

Workgroup management functions provide file and session management. Files can be either saved, archived, or transmitted to or received from other users. The workgroup management interface also allows a user to make a phone call from on-line within the workgroup. For privacy, the user can be isolated from other workgroup activities while using the phone function. Users can select encryption and data compression for designated sessions and files. Finally, an In/Out board is available to track status of workgroup members.

The architecture supports both synchronous and asynchronous workgroup computing. A single user can be involved in a synchronous meeting session, leave the session to post a message on a bulletin board or send an E-mail message and then return to the original session or a new concurrent session.

Workgroup support provides programming interfaces for workgroup functions that are common to a large number of available applications. These functions need only be programmed once so they can be called by the Windows operating system whenever they are required by an application. Applications support the work tasks by providing necessary tools to complete functions.

The architecture can be seen as a broad set of functions that provide workgroup members with access to applications and support capabilities. The architecture's extensive resources are available to the workgroup so that it can accomplish workgroup tasks. If an application or interface has not been defined it can be easily added to the architecture when it is needed or available.

The workgroup computing structure does not have to be integrated into the operating system layer. Since the workgroup computing architecture is independent of the technology for operating systems and networking, either can be easily changed to support future workgroup requirements. Conversely, since the architecture is not integrated into the operating system it can also be changed with minimal impact. Any windows-based operating system can be implemented as the user interface. Should a new operating system be necessary, the workgroup computing APIs can be ported over to a new operating system with a minimum of programming effort.

B-ISDN and ATM were selected as the network interface to ensure that the network speed and bandwidth are available for video, multimedia, and parallel communication channels. New high-speed networking capabilities can be added when they are available. If needed, additional communication modes can be added to the communication API.

This system architecture is aligned with the controlled architecture proposed by University of Arizona Group Support Systems researchers rather than the open support systems model being researched by the University of Minnesota Group. In this architecture, session leaders must still control many of the workgroup activities and system definitions. There is adequate flexibility in the architecture to allow a moderate degree of choice in configuring the system to meet individual needs.

E. ***Application Programming Interface (API) Specifications***

Based on the developed workgroup computing tasks and primitives and the architecture, 95 workgroup computing APIs were identified in 10 functional areas. Of these, specifications were written for 86 new APIs that covered all identified common workgroup computing management and support functions. Existing APIs were found in the available API source literature for the other 9 functions. For example, workgroup mail APIs were not written since the emerging defacto standard for workgroup mail is the Microsoft Mail Application Programming Interface (MAPI) (Sheldon, 1994). The complete API specification listing is presented in Appendix D.

Based on the sorting of workgroup computing tasks and primitives, the API specifications were divided into two groups: workgroup management APIs and workgroup support APIs. Developed API specifications support the defined architecture and they allow maximum flexibility for configuring workgroup sessions.

Using workgroup management API functions, a workgroup leader or designated user can create and start a single or multiple workgroup sessions. Access authorizations can be defined or changed for each session allowing users to logon and logoff appropriate sessions. Invitations can be sent to other users or users can request to join a particular session. The session leader can give permission to join a group in progress. During a session, a full range of communication options is available to the session leader and to workgroup members. The interface can be text only to a full multimedia featuring audio and video muting by selected users. Floor, version, and revision control is provided for all synchronous or asynchronous workgroup sessions. Phone support is available.

Workgroup file management functions allow workgroup files to be opened or closed. A full range of session filing functions such as save session, recall session, archive session and playback session, are available. Objects or text can be imported into or exported from any session or file. Multiple display windows can be defined and views and private space can be tailored to the user's needs. As desired, the session leader can merge or split sessions.

Other workgroup session management functions include session and file encryption and data compression that can be selected by the workgroup leader or individual users. A digital signature can be added to any document or form. Finally, an In/Out board can indicate the status of workgroup members.

Workgroup support functions allow users to perform common workgroup tasks and primitives when they are attached to a workgroup session or if they desire to perform an asynchronous workgroup task. The calendar and scheduling function provides links to a workgroup calendar and scheduling activity. "To do" lists can be developed and maintained or shared with other users. Users can select a proxy and provide permission for the proxy to represent the user in various workgroup activities and decision making processes.

Databases provide resources for filing, data collection, and group memory. Functions are available that allow group access to selected databases so that they can be browsed, updated, filtered, and sorted. Database functions can support all workgroup activities. They also provide workgroup memory that is important for future workgroup sessions and decision making.

Decision tools allow a user to select a preferred decision style. This can range from only obtaining information from the group members to make a decision to full involvement of all session members in defining and selecting decision alternatives. A decision model can be selected that allows consensus building, polling, and voting on issues and alternatives. Individual or group tasks can be assigned by authorized users. Users can accept or reject these tasks or request clarification. Users can make proposals relating to alternatives and ideas. If necessary, declarative decision rules can be developed in a private or collaborative environment. Idea functions allow users to generate and organize group ideas. Idea lists can be prioritized or provided to the group for consensus, polling, or a vote.

A full range of group drawing functions allow users controlled access to drawing objects. Users can access private views or move to an empty portion of the workgroup workspace to perform individual work. This work can be integrated into the workgroup workspace at a later time. When desired, a user toggles back to the workgroup view. In addition, the system can show the virtual position of all users in the drawing or design environment. This function is especially useful when working and editing a graphic or while drawing or designing in the 3D mode.

E-mail links to all workgroup users allow users to send and receive multimedia E-mail messages and attachments either privately or while they are logged on to a work-group session. Using inputs form workgroup members, a group address book is kept current by the system. Return mail receipts can be requested and access permission can be provided to access another user's mailbox. Mail folders can also be shared based on agreements between individual group members.

Two closely related but different sets of functions are the text input function and the document writing and editing function. The text function allows a user to insert and edit text in a small section of a workspace or document. Writing and editing functions provide full configuration, revision, and version control of the writing workspace. Writing strategies and rules can be selected, documents can be controlled, and access and editing permissions designated. Document editing is the same with either text or writing functions; however, formal workspace control functions exist in the writing and editing mode. There is also an API function that performs group video editing.

The last set of API functions handles workflow definition and control. A complete range of workflow rules and criteria can be defined. Routing of the work package can be defined and work actions assigned. This API function can be integrated with all work-group applications.

F. *Implications for the Field of Workgroup Computing*

This study is important for the collaborative computing field because it provides a detailed listing of workgroup tasks and primitives ordered as a taxonomy. It also provides an integrated workgroup computing architecture that includes common workgroup support and management tasks. API specifications were developed for these common workgroup computing tasks.

The comprehensive taxonomy provides the foundation for developing integrated workgroup computing products that meet the needs of diverse workgroup members. Where relevant, workgroup computing tasks and primitives were combined into common workgroup management and support functions. These common workgroup tasks and primitives were extended to functional areas that were not covered in the taxonomy. These common functions can be implemented by application developers and researchers across a defined set of workgroup applications included in an integrated architecture.

The workgroup computing architecture that was defined includes these common workgroup management and workgroup support functions. The study may help workgroup computing researchers to focus and expand their research into providing broader and more integrated workgroup computing applications. The architecture includes communication modes that support multimedia and parallel communication channels. This architecture may lead to new classes of workgroup applications such as virtual workgroup applications that use high-resolution and 3D imagery. For example, these new workgroup applications may be used to allow groups to safely learn manufacturing and assembly operations for critical or dangerous tasks without being exposed to the real environment. Surgical teams may realistically practice complex operations or delicate procedures to reduce risk to a patient.

This next generation of workgroup systems may also allow disabled individuals to take full advantage of their capabilities and change their interaction styles by using all their senses in a multimedia workgroup setting. The system can simulate the performance of an individual or group function that would not be possible due to the individual's handicap.

A set of workgroup computing API specifications was generated that can be used to implement the new architecture. These API specifications can form the basis for implementation of workgroup applications that are ubiquitous, fully integrated, and extendable to every work or leisure group activity. A workgroup user can fully connect to the workgroup sessions they choose allowing daily work or leisure activities to be conducted via networked computers from the comfort of a user's office or home.

G. *Recommendations*

Recommendations for the field of workgroup computing.

This study can be used by researchers and workgroup application developers to determine which common workgroup computing functions should or should not be included in new workgroup computing products. The results of this study can be used to develop new integrated workgroup applications or future versions of existing workgroup applications such as DEC LinkWorks, Lotus Notes, and Microsoft's and IBM's operating system workgroup products. Including the results of this study in future workgroup computing systems will lead to more flexible and integrated systems that are easier to use and more transparent to the user. Also, workgroup computing researchers can review this study to help identify all necessary workgroup computing functions that should be included in their research areas.

Recommendations for future research.

This study provides the foundation for additional research in several interesting workgroup areas that can expand the field of workgroup computing. The following are a few areas that may merit further study.

Intelligent workgroup computing architecture

The workgroup computing taxonomy provides a detailed structure to help identify where intelligent agents can be incorporated into the workgroup computing system. Existing workgroup architectures, including the one developed in this study, depend on the user or users to select the system configuration that best suits the defined work activity. There may be a way to automate or make this process more intelligent. Develop a smart workgroup architecture and produce an integrated and intelligent workgroup model. Define the impact on the workgroup and the workgroup user.

Implement and test the APIs on a network

This study provides a workgroup architecture and implementing API specifications. Program the APIs for Windows 95 using C or C++ and implement it with several applications on a networked system. Test and refine the architectural model. Identify issues and problems and show how they can be can be rectified. Discuss the impact on the workgroup architecture.

Extend results to a new class of workgroup computing applications

Using existing workgroup research and groupware applications, this study developed the workgroup computing taxonomy, architecture, and APIs. Therefore, results are limited to what presently exists. Extend the results and identify and apply them to new classes of workgroup computing applications. Define these new applications classes. Identify the gaps and determine what new APIs are required.

Develop an object-oriented workgroup computing model

This study is based on procedural programming concepts and models that are familiar to the author. As discussed in the introduction to this thesis, the future is in object-oriented modeling and programming (Booch, 1994; Taylor, 1990). Develop an object-oriented workgroup computing model and architecture. Compare and contrast the object-oriented model to the model developed in this study.

Assessment of unpredictability of user needs versus provided workgroup computing services

The human element of workgroup systems is unpredictable (Bair, 1993). This may impact workgroup systems (Neuwirth et al., 1990). Develop the relationship between user needs and the services provided by the architecture. Show the impact of unpredictable behavior and how defining roles may inhibit motivation. Using the predicted flow define activities that have no supporting tools. Identify gaps in the support structure and recommend changes required to improve the system response to unpredictability.

Distributed workgroup multimedia storage systems

Storage systems for workgroup session information and databases will require megabyte, gigabyte, and eventually terabyte storage technology (Stallings & Van Slyke, 1994). Distributed group requirements will make filing, manipulation, sorting, and extraction of necessary data extremely difficult (Stallings & Van Slyke, 1994). Define how these data files should be structured. Determine the optimum access speed for different workgroup situations. Identify the best storage medium and extend it to future system configurations. Determine if workgroups should be allowed to structure their own file management system. Define how this might take place.

Future computer system and network support configurations

Current computer systems and networks were not been designed with workgroup support in mind. Compromises have had to be made to make workgroup computing systems function. The systems are the limiting factor in how far workgroup computing technology can be integrated into daily work and life processes (Koshafian et al., 1992). If user collaboration is a basic computer system requirement, define what the networking and computer systems of the future should look like.

Customizable workgroup computing applications

Each workgroup situation requires specific software to meet organizational needs. Existing software must be modified or new software must be produced to meet organizational workgroup computing requirements. The trend is to move away from software development within companies (Opper & Fersko-Weiss, 1992). If this is true, develop the building blocks for modular workgroup computing software. Show how it can be tailored to fit group and individual needs while meeting organizational goals. Show how existing applications such as WordPerfect and Excel can be seamlessly integrated into this environment.

Virtual reality conferences and interactive training

Virtual reality applications will allow users to experience life-like situations in real-time. Develop a virtual reality conference or interactive training module that integrates the activities of group members in remote locations. Define the impact on the user interface and develop the workgroup functions that are required. Investigate enabling technologies required to implement this concept.

Interactive group home shopping network

Interactive group shopping will allow friends or groups to shop together without leaving their homes. Define the workgroup system that can make this happen. Investigate the supporting technologies, such as networking improvements and virtual reality, that are required for implementation. Discuss the impact on the user interface.

True democratic voting

Using advanced workgroup computing technology, political issues and information can be presented and discussed by large democratic groups to allow consensus building, polling, and voting. Define a group decision support system that extends this concept to the American political process as it is currently defined. Address the impact of this type of group decision support system on the American political system and culture.

H. *Summary*

After thorough research, this study allowed a comprehensive workgroup computing taxonomy, an integrated workgroup computing architecture, and new common workgroup computing API functions to be developed. The results of this study supported the hypothesis that the available workgroup computing literature and workgroup applications' documentation would provide sufficient information to develop a workgroup computing taxonomy.

The developed taxonomy comprehensively covers 11 workgroup computing functional areas by providing details down to the workgroup task or primitive level. These results will help the workgroup computing community to develop more integrated workgroup computing systems and applications that support the diverse needs of workgroups and workgroup members.

The developed workgroup computing architecture consists of four main elements: workgroup applications, workgroup support interfaces, workgroup management interfaces, and the operating system and network interfaces. To allow the workgroup to function efficiently, the architecture includes all necessary workgroup computing management and support elements. The architecture provides a common style of interface for all workgroup activities that permits a user to access shared functions. A user can adapt each workgroup session to a specific set of workgroup styles. Since all workgroup functions are defined and accessible by all users, complex, multiple-session workgroups can be supported.

Based on the developed workgroup computing tasks and primitives and the architecture, 95 workgroup computing APIs were identified in 10 functional areas. Of these, 86 were new APIs that covered all identified common workgroup computing management and support functions. Existing APIs were found in the available API source literature for the other 9 functions.

This study is important for the collaborative computing field because it provides a detailed listing of workgroup tasks and primitives ordered as a taxonomy. It also provides an integrated workgroup computing architecture that includes common workgroup support and management tasks. API specifications were developed for these common workgroup computing tasks. Researchers can use the results of this study to improve research focus by ensuring that appropriate workgroup computing functions are included in research projects. Application developers can improve the functionality and scope of workgroup applications by including a wider range of workgroup computing functions in developed products.

Finally, study results can be used as a basis for future workgroup computing research that will expand and clarify the workgroup computing field. Recommendations for the field of workgroup computing and future research were discussed.

Bibliography with Annotations

This bibliography includes articles that were cited in the study and recommended readings. Annotations are included. Recommended readings provide additional workgroup computing information that may be useful for future research or work. Recommended readings have an "(rr)" after the citation. The reference list includes articles from the following areas: collaborative computing, communications, computer supported cooperative work, distributed systems, groupwork, groupware, hypermedia, multimedia, networking, object-oriented programming, workgroup computing, and workgroup structures. Many articles in this reference list were obtained from a few major compiled sources including CSCW, Groupware, or Computer-Human Interaction (CHI) conference proceedings. Important workgroup computing articles were obtained from CSCW '90, CSCW '92, Groupware '92, Groupware '93, Groupware '94, and CHI '91.

Ahuja, S. R. (1992). Multimedia communications: An infrastructure for remote collaborations. In David D. Coleman (Ed.), *Groupware 92* (pp. 413-414). San Mateo, CA: Morgan Kaufmann. (rr)

Ahuja, from AT&T Bell Laboratories, proposes that both asynchronous and synchronous multimedia communications are necessary to support collaborative computing at a distance. The media must be transparent, interruptable, and sharing for a successful implementation. The authors believe that remote collaboration requires integrated multimedia communications and not just an extension of a single media.

Allison, D. (1992). Groupware: Systems support and glueware. In David D. Coleman (Ed.), *Groupware 92* (pp. 231-233). San Mateo, CA: Morgan Kaufmann. (rr)

Groupware is the software that facilitates group efforts. It can be used for decision making or the collaborative development of complex systems. The author stresses that groupware must be tailored to the needs of the task as well as the organization using it. The underlying support systems including networks, databases, and file transfer are discussed. When using a number of different applications programs the software that holds together the underlying components is called glueware.

Anderson, W. L. (1991). Group relations psychology and computer supported work: Some new directions for research and development. In P. de Jong (Ed.), *Conference on Organizational Computing Systems* (pp. 117-122). Atlanta, GA: ACM Press. (rr)

Anderson looks at the nature of Groupwork so it can be better understood. Anderson discusses his "Psychodynamic Model" (p. 117) of group relations which provide insight into the day-to-day activities of work groups and larger organizations. It also gives the researcher information on how to develop richer models.

Anderson recognizes that computer systems affect the social conditions of work groups. In order for CSCW methods to be effective they must incorporate the social sciences during design and implementation. Anderson ran several experiments on computer-mediated communications which are reported on in this article. Results of these experiments provide insight into implementing CSCW techniques in organizations.

Baecker, R. M., Nastos, D., Posner, I. R., & Mawby, K. L. (1993). The user-centered iterative design of collaborative writing software. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, and T. White (Eds.), *InterCHI '93: Conference Proceedings on Human Factors in Computing Systems* (pp. 399-405). Amsterdam, The Netherlands: ACM Press.

The authors present the design of user-centered collaborative writing software. A taxonomy is presented along with a set of design requirements. They also present lessons learned from several tests.

Bair, J. (1993). Contrasting workflow models: Getting to the roots of three vendors. In David D. Coleman (Ed.), *Groupware 92* (pp. 229-237). San Mateo, CA: Morgan Kaufmann.

This paper reviews several workflow software models and discusses fundamental differences. Discusses workflow elements including forms, documents, activities, users, access, and graphics.

Bannon, L., Bjorn-Anderson, N., & Due-Thomsen, B. (1988). Computer support for cooperative work: An appraisal and critique. In H. J. Bullinger (Ed.), *Eurinfo '88: Information Systems for Organizational Effectiveness.* Amsterdam: North-Holland.

This paper provides background and definitions for CSCW. An appraisal of the field indicates that it is growing and that there are problems insuring the effectiveness of groupwork implementations.

Bannon, L. J., & Schmidt, K. (1991). CSCW: Four characters in search of a context. In J. M. Bowers & S. D. Benford (Eds.), *Proceedings of the First European Conference on Computer Supported Cooperative Work* (pp. 3-16). North Holland: Elsevier Science.

The authors examine the assumptions in the name "Computer Supported Cooperative Work (CSCW)". Core issues of articulating cooperative work, sharing an information space, adapting the technology to the organization and vice versa must all be addressed for this area to be successful. The authors conclude that CSCW is a valid representation of the research area and that there is much work to be done to address CSCW issues.

Baydere, S., Casey, T., Chuang, S., Handley, M., Ismail, N., & Sasse, H. (1993). Multi-media conferencing as a tool for collaborative writing: A case study. In M. Sharples (Ed.), *Computer-supported collaborative writing* (pp. 113-135). London: Springer-Verlag.

The authors investigated how multimedia conferencing systems support collaborative work. A case study is used to show how this chapter in the book was produced by a team effort.

Beaudouin-Lafon, M., & Karsenty, A. (1992). Transparency and awareness in a real-time groupware system. *Proceedings of the ACM Symposium on the User Interface Software and Technology* (pp. 171-180). Monterey, CA, New York: ACM Press. (rr)

This article explores real-time groupware systems from both designer and user points of view. The exploration is carried out using GroupDesign, a real-time, multi-user drawing tool that was developed by the authors. Beaudouin-Lafon and Karsenty advocate "transparent groupware systems" (p. 171) that do not require special settings or hardware and that integrate smoothly with the way computers are used. Group awareness is also an important factor. Preliminary results show that the features proposed by the authors were easily understood by the users.

Benest, I. D., & Dukic, D. (1993). Computer supported teamwork. In D. Diaper and C. Sanger (Eds.), *CSCW in practice: An introduction and case studies* (Chap. 9, pp. 127-150). New York: Springer-Verlag.

This chapter describes the Automated Office Metaphor. Emphasis is placed on the ability to share captured and computed information. This chapter emphasizes that it is the electronic equivalent of office forms that turns information sharing into a conferencing

system. A number of these instruments are described and implementation problems are discussed.

Benford, S., Mariani, J., Navarro, L., Prinz, W., & Rodden, T. (1993). MOCCA: An environment for CSCW applications. In S. Kaplan (Ed.), *Conference on Organizational Computing Systems* (pp. 172-177). Milpitas, CA: ACM Press.

The authors describe a CSCW environment called MOCCA that may facilitate interaction between different CSCW applications. Models are described and discussed to show how an open distributed architecture can be implemented.

Bentley, R., Rodden, T., Sawyer, P., & Sommerville, I. (1994, May). Architectural support for cooperative multiuser interfaces. *IEEE Computer*, pp. 37-46.

An air traffic control workgroup architecture was developed that allowed the researchers to investigate a wide range of user interfaces. Office information systems often inhibit cooperation because users are not aware of other users on the system. Interfaces developed for this project allow the real-time presentation and manipulation of shared information.

Benton, P. M., & Devlin, J. (1991). Workgroup automation and hypertext. In E. Berk & J. Devlin (Eds.), *Hypertext/hypermedia handbook* (pp. 415-434). New York: McGraw-Hill. (rr)

This chapter from the Hypertext Handbook reviews the evolution of work group automation. It also discusses problems and some examples of how systems can be implemented in a Hypertext environment. Benton et al. also discuss what other tools are required for success.

Benton et al. say that "Most of these systems will fail" (p. 416). The primary reason for failure is that the programs do not consider problems that are unique to working in the networked environment. Better cognitive models are needed which include social structure, politics, and group dynamics.

Berlin, L. M., & Jeffries, R. (1992). Consultants and apprentices: Observations about learning and collaborative problem solving. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 130-137). Toronto, Canada: ACM Press. (rr)

The interaction between experts and apprentices is a common, but little studied, collaboration in workgroups. Berlin and Jefferies conducted empirical studies of this relationship. Results provided insights into the nature of this collaborative work process. The usefulness of the computer as a collaborative tool is validated.

Bikson, T. K., & Eveland, J. D. (1990). *Interplay of work group structures and computer support (RAND Report No. N-3429-MF).* Santa Monica, CA: RAND Corporation. (NTIS No. AD-A256 803/8/XAB)

When group members communicate using computers instead of traditional means, much about the group can change. Group structure, intensity of communication, interaction across physical barriers, and the work process are all explored in a field experiment among active workers and retirees planning a company's retirement policy. The study shows the effect of computer communication and how it can help reduce barriers to social interaction in distributed groups.

Bly, S. A., Harrison, S. R., & Irwin, S. (1993). Media spaces: Bringing people together in a video, audio, and computing environment. *Communications of the ACM, 36*(1), 29-47. (rr)

Productive workgroups must smoothly integrate both casual and task-specific interactions. Most groupwork tools address the computaional aspects of the collaborative process. The authors discuss the need for media spaces which address the social and interactive aspects of the collaborative work environment. Informal interactions, spontaneous conversations, and awareness of people and events are discussed in this comprehensive article that ties together video, audio, and computing.

Bly, S. A. & Minneman, S. L. (1990). Commune: A shared drawing surface. In F. H. Lochovsky and R. B. Allen (Eds.), *Conference on Office Information Systems* (pp. 184-192). Cambridge, MA: ACM Press.

Drawing processes are as important to the design process as drawings themselves. Commune is a shared drawing system that allows designers remote access to shared drawing spaces. Interaction is possible and support is similar to face-to-face sessions.

Bock, G. (1992). Workflow a groupware: A case for group language? In David D. Coleman (Ed.), *Groupware 92* (pp. 168-170). San Mateo, CA: Morgan Kaufmann.

The authors contend that workflow concepts may be part of groupware. Groupware applications can enhance organizational effectiveness. The author makes the case for enabling a groupware language to build "coordinated environments" (p.168).

Bodker, S., Knudsen, J. L., Kyng, M., Ehn, P., & Madsen, K. H. (1988). Computer support for cooperative design. In D. Marca and G. Bock (Eds.), *Groupware: Software for computer-supported cooperative work* (pp. 82-99). Los Alamitos, CA: IEEE Computer Society Press. (rr)

The authors outline their theoretical perspective on design as cooperative work and they demonstrate their approach using a system called APLEX. Facilities that support experiments and future situations are presented along with background and reflections on the rationality of computer use for supporting cooperative work.

Booch, G. (1994). *Object-oriented analysis and design: With applications* (Second Edition). Redwood City, CA: Benjamin/Cummings.

The author provides fundamental concepts for the object-oriented model, notation process, and object-oriented analysis and design.

Borenstein, N. S., & Thyberg, C. A. (1991). Power, ease of use and cooperative work in a practical multimedia message system. *International Journal of Man-Machine Studies 34*(2), 229-259.

Describes the operation of a successful multimedia mail and bulletin board called Messages. The "Messages" program is an interface to the Andrew Message System (AMS). It is easy to learn; however, it is extremely powerful and it satisfies both novices and experts.The system which is part of the Andrew Advisor system supports cooperative work that is not possible with other mail systems. The system is used weekly by about 5300 people at Carnegie Mellon University to read bulletin boards. This successful user interface is described in this paper.

Borland, R., Lorenz, L. L., & O'Mara, R. M. (1993). *Windows for Workgroups companion*. Redmond, WA: Microsoft Press.

This is an application manual for Windows for Workgroups.

Bostrom, R. P., Watson, R. T., & Kinney, S. T. (1992). *Computer augmented teamwork: A guided tour*. New York: Van Nostrand Reinhold. (rr)

A guide to group systems technology, electronic meeting technologies, work group supports. Provides extensive information on tailoring computer support to meet organizational needs.

Bowers, J., & Rodden, T. (1993). Exploding the interface: Experiences of a cscw network. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (Eds.), *INTERCHI '93: Conference Proceedings on Human Factors in Computing Systems* (p. 255-262). Amsterdam, The Netherlands: ACM Press.

Bowers and Rodden studied different views of the computer interface in the context of workgroups. Traditional concepts of the interface were modified to provide richer conceptualizations of the relationships between users and computer systems. These alternative concepts were examined within the framework of actual organizations to determine applicability and usefulness. This work will be useful in defining the CSCW interfaces in various workgroup settings.

Brittan, D. (1992). Being there: The promise of multimedia communications. *MIT Technology Review 95*(4), 42-50.

Multimedia experiments combining voice and data with video can revolutionize desktop computer communications. They can also redefine the office structure and allow individuals to work at home. Several experimental systems providing elements of this advanced communications capability are described.

Bullin, C. V., & Bennett, J. L. (1991). Groupware in practice: An interpretation of work experiences. In C. Dunlop and R. King (Eds.), *Computerization and controversy: Value conflicts and social choices*. New York: Academic Press.

The authors interviewed 223 people in 25 organizations to determine how groupware systems were being utilized. They defined groupware as "computer-based tools that can be used by workgroups to facilitate the exchange and sharing of information" (p. 71). Bullen, et al. were attempting to understand the value that technology brings to the office environment.

Butterfield, J., Rathnam, S., & Whinston, A. (1992). Groupware: A survey of perceptions and practice. *SIGOIS Bulletin*, pp. 6-7. (rr)

The authors conducted a survey to "measure people's perceptions of the field" (p. 6) of groupware. The survey was administered by electronic mail and asked simple questions concerning user perceptions. The results may suggest directions for future research.

Campbell, D. S. (1992). Calendaring and group scheduling. In David D. Coleman (Ed.), *Groupware 92* (pp. 388-390). San Mateo, CA: Morgan Kaufmann.

Discusses the problems with and a successful solution, OnTime, to group scheduling and calendaring.

Carbom, I., Hsu, W., Klinker, G., Szeliski, R., Waters, K., Doyle, M., Gettys, J., Harris, K. M., Levergood, T. M., Palmer, R., Palmer, L., Picart, M., Tonneson, D., Vannier, M., & Wallace, G. (1992). Modeling and analysis of empirical data in collaborative environments. *Communications of the ACM, 35*(6), 75-84. (rr)

The authors have a vision of a "collaborative scientific visualization environment" (p. 75) where various professionals can work together on analysis of empirical data. An integrated set of tools and techniques from computer graphics, computer vision, and image processing will be used. The environment will also take advantage of interaction techniques such as sound, speech, and gestures. Developed techniques are presented in a series of scenarios and demonstrations. The authors indicate that their techniques will have a profound effect on collaborative work.

Celentano, A., Fugini, M. G., & Pozzi, S. (1991). Classification and retrieval of documents using office organization knowledge. In P. de Jong (Ed.), *Conference on Organizational Computing Systems* (pp. 159-164). Atlanta, GA: ACM Press.

This paper provides an approach to document classification, filing, and retrieval uisng semantic model documents. It covers a classification scheme and retrieval system based on document roles. It includes a discussion of the Kabiria project that used knowledge techniques for document management.

Cerf, V. G. (1991). Networks. *Scientific American,* pp. 72-81. (rr)

The increasing number of computer applications requires an increase in the flow of data between machines. As the demands on networks increase, the flow will be handled by wider bandwidth and more stable networks. Cerf reviews the various technologies, such as ISDN-B, that can make this happen.

Chalstrom, B. (1993). Lotus Notes, release 3 offers something good for everyone. *Infoworld*, pp. 68-76.

Provides an up-to-date review of the latest features included in Lotus Notes, Release 3.

Cini, A. (1993). The object-oriented enterprise. *DEC Professional*, pp. 48-53.

Describes DEC PathWorks, a general-purpose PC integration product suited for ad hoc, informal workgroups that exist in most organizations.

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnik, J. M. Levine & S. D. Teasley (Eds.), *Perspectives on socially shared cognition.* (pp. 127-149), Washington, DC: American Psychological Association. (rr)

The principle of grounding is basic to communications and all collective actions. Communication requires the coordination of both content and process. Different mediums of communication require different techniques. All collective actions are built on common

ground and its accumulation. Clark and Brennan discuss communication grounding concepts and techniques that may also be applied to workgroup computing.

Clement, A. (1990). Cooperative support for computer work: A social perspective on the empowering of end users. In F. Halasz (Ed.), *CSCW 90: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 223-236). Los Angeles, CA: ACM Press. (rr)

Clement addresses the question of empowerment through computerization. He looks at common problems found in an office administrative environment. Based on the study findings, proposals were formulated for design and implementation of CSCW applications that could enhance the power of office workers. Research results showed that secretaries in the study became more empowered by forming collaborative networks and establishing training courses. This result had been relatively invisible in previous CSCW studies.

Coleman, D. E. (1992). Integrating groupware into corporate cultures. In David D. Coleman (Ed.), *Groupware 92* (pp. 113-116). San Mateo, CA: Morgan Kaufmann. (rr)

Studies within companies that have implemented groupwork systems have seen a culture shift within the organization. In order to understand this phenomena Coleman discusses the types of groupware that are available, the types of organizations being effected, and the types of collaborative processes being utilized. A key point is that collaborative or interactive groupware allows individuals to act anonymously, increase participation, achieve consensus, and generally improve the quality and quantity of workgroup output.

Conger, J. (1992). *Windows API bible: The definitive programmer's reference.* Corte Madera, CA: Waite Group Press.

Provides Windows' application programming interface (API) specifications for over 800 functions organized by API category.

Cook, S., Birch, G., Murphy, A., & Woolsey, J. (1991). Modelling groupware in the electronic office. In S. Greenberg (Ed.), *Computer-supported cooperative work and groupware* (pp. 243-267). London: Harcourt Brace Jovanovich.

This project explored ideas on how a distributed system could enhance the effectiveness of communications including support for planning and completing cooperative tasks. It includes a discussion of architectural issues and future workgroup products. The approach created software models that were used in group experiments.

Cool, C., Fish, R. S., Kraul, R. E., & Lowery, C. M. (1992). Iterative design of video communication systems. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 25-32). Toronto, Canada: ACM Press. (rr)

The authors review a case study on the design and implementation of a video telephone at Bellcore. System capabilities depend jointly on what the developers implement and how the users behave toward the system. Four dilemmas for iterative design due to the social nature of the communication system are discussed. The authors conclude with design recommendations to supplement traditional communication system design principles.

Cortese, A. (1993, November). IBM, Apple plan for Taligent. *PC Week*, pp. 61-65.

Taligent will improve how users can display and use workgroup information. Taligent will be the first completely object-oriented OS, providing developers with pre-programmed collections of code objects called frameworks. This new OS will improve upon existing graphical user interfaces and dramatically speed up data flow between workgroup users.

Craighill E. (1992). Commercial multimedia software. In David D. Coleman (Ed.), *Groupware 92* (pp. 408-412). San Mateo, CA: Morgan Kaufmann.

This paper discusses interaction in shared environments and products that share and manipulate multimedia information at remote locations. It addresses the shared work-space concept.

Crowley, T., Millazzo, P., Baker, E., Forsdick, H., & Tomlinson, R. (1990). MMConf: An infrastructure for building shared multimedia applications. In F. Halasz (Ed.), *CSCW*

90: Proceedings of the Conference on Computer-Supported Cooperative Work (pp. 329-342). Los Angeles, CA: ACM Press.

Computers have traditionally allowed asynchronous group work through shared file systems. MMConf explored how computers can support real-time, distributed, group collaborations. The architecture has been implemented on UNIX systems as both a toolkit and conference manager. Applications that support real- time cooperative work have been produced and tested.

Cutkosky, M. R., Engelmore, R. S., Fikes, R. E., Genesereth, M. R., Gruber, T. R., Mark, W. S., Tenenbaum, J. M., & Weber, J. C. (1993). Pact: An experiment in integrating concurrent engineering systems. *Computer*, pp. 28-37. (rr)

The Palo Alto Collaborative Testbed (PACT) is being developed by several research groups. The approach has been to integrate existing multi-tool systems encompassing multiple sites, subsystems, and disciplines. A goal has been to minimize the impact to engineering groups that have developed their own tools. Using this testbed, four teams developed a robotic device simulation and synchronized their efforts on a design modification.

Dale, T. (1993). Groupware, the Macintosh, and collaborative environments. In D. Coleman (Ed.), *Groupware 93* (pp. 316-332). San Mateo, CA: Morgan Kaufmann.

This paper discusses a study of groupware running on a Macintosh network. It includes discussion of issues for both technical and human-computer interaction. The paper concludes with benefits and problems of building this type of system. It also compares commonly expressed views of groupware with real-world experiences.

Denley, I., Whitefield, A., & May, J. (1993). A case study in task analysis for the design of a collaborative document production system. In M. Sharples (Ed.), *Computer supported collaborative writing* (pp. 161-184). London: Springer-Verlag.

Presents issues related to the design of a multimedia collaborative writing system. Progesses from basic concepts to the final design and its utility.

Dewan, P., & Choudhary, R. (1991). Flexible user interface coupling in a collaborative system. In S. P. Robertson, G. M. Olson and J. S. Olson (Eds.), *CHI '91 Conference Proceedings* (pp. 41-48). New Orleans, LA: ACM Press. (rr)

The kind of sharing or coupling between various different windows is an important issue in the design of multi-user applications. Dewan and Choudhary address the notion of "flexible coupling" (p. 41) and its associated issues. The authors designed a coupling approach for a collaborative application called Suite. Most aspects of the Suite approach can be applied to other workgroup applications that support an interactive editing model.

Dewan, P., & Reidl, J. (1993). Toward computer-supported concurrent software engineering. *Computer*, pp. 17-27. (rr)

Flexible environment for Collaborative Software Engineering (Flecse) is a new multimedia environment that supports both synchronous and asynchronous collaboration. It features tools designed to overcome collaboration problems between software engineers. The authors discuss tools, concepts, life cycle, integration, and sharing. Solutions identified in the software engineering area can be applied to other collaborative work areas.

Digital Equipment Corporation (1994a). *Software product description: LinkWorks Client for Microsoft Windows, Version 2.1.* (SPD 48.55.00). Nashua, NH: Author.

This is a software product description for DEC Linkworks for Microsoft Windows. It provides details of LinkWorks workgroup software functions.

Digital Equipment Corporation (1994b). *Software product description: TeamLinks Information Manager for Microsoft Windows, Version 2.0.* (SPD 37.36.05). Nashua, NH: Author.

This is a software product description for DEC TeamLinks for Microsoft Windows. It provides details of TeamLinks workgroup software functions.

Digital Tools (1993). *User's Guide: AutoPLAN II.* Cupertino, CA: Author.

User's guide for AutoPLAN II. The program performs project management for workgroups.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1993). *Human-computer interaction.* New York: Prentice-Hall.

Chapter 13 provides extensive information on groupware frameworks and implementation. Chapter 14 discusses CSCW issues and theory.

Dourish, P., & Bellotti, V. (1992). Awareness and coordination in shared workspaces. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 107-114). Toronto, Canada: ACM Press.

Successful collaboration requires awareness of both individual and group activities. CSCW systems support this with information generation systems separate from the shared workspace. Dourish and Bellotti discuss a study of passive awareness mechanisms that allow users to be aware of and exploit information through the shared environment while avoiding the problems with active approaches. Users can move between loose and close collaboration and coordinate work dynamically.

Dressler, F. R. S. (1992). Delivering groupware in the 90s. In David D. Coleman (Ed.), *Groupware 92* (pp. 375-377). San Mateo, CA: Morgan Kaufmann. (rr)

Groupware is extremely difficult to pin down. It means many things to different people. Dressler discusses the need to establish "complementary and richly-developed" (p. 375) delivery channels for groupware. Dressler uses Novell as an example of how this worked and fostered the LAN industry.

Dubs, S., & Hayne, S. C. (1992). Distributed facilitization: A concept whose time has come? In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 314-321). Toronto, Canada: ACM Press.

When a workgroup is geographically separated, group facilitization becomes difficult. The authors are designing and implementing a prototype facilitization system that will support remote collaboration.

Dyson, E. (1992). A framework for groupware. In D. Coleman (Ed.), *Groupware '92* (pp. 10-20). San Mateo, CA: Morgan Kaufmann. (rr)

The author sees groupware as "a tool for change" (p. 10) in an organization. He provides a new framework for structuring groupware. Dyson classifies groupware as information or workflow oriented with user-centered, work-centered, or process-centered categories.

Easterbrook, S. M., Beck, E. E., Goodlet, J. S., Plowman, L., Sharples, M., & Wood, C. C. (1993). A survey of empirical studies of conflict. In S. Easterbrook (Ed.), *CSCW: Cooperation or conflict* (pp. 1-68). London: Springer-Verlag. (rr)

Chapter 1 discusses conflict and its relation to CSCW. Conflict is defined according to contemporary literature relevant to CSCW. Assertion are related to work on current CSCW systems. This chapter also provides a top level structure for CSCW applications.

Egan, G. A. (1993). Groupware: It's for all companies. *Inside DPMA*, pp. 7-20.

Egan, defines groupware and he divides it into several groupings. These include decision making, workflow, and work management.

Ellis, C. A., Gibbs, S. J., & Rein, G. L. (1991). Groupware: Some issues and experiences. *Communications of the ACM, 34*(1), 39-58.

This article provides a broad view of Groupware. Ellis, Gibbs, and Rein consider Groupware as a merge of computers, large information databases, and communication technology. They consider Groupware to be ". . .computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment" (p. 40).

Ellis et al. looked at several design issues, one of which included the ability to communicate using selected software packages. In the area of communications, Ellis et al. saw the goal of groupware as an interface to a shared environment which would help the groups to communicate and do other comparable functions.

The challenge from a communications perspective is how to make distributed interactions as effective as face-to-face communications. The authors suggest that groupware would be useful in both a face-to-face situation and also over longer distances. Their assessment was that "While this will not replace face-to-face communication, it may actually be preferable in some situations for some groups because certain difficulties, inconveniences, and breakdowns can be eliminated or minimized" (p. 40).

Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janesen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J., & Welch, B. (1992). Liveboard: A large interactive display supporting group meetings, presentations, and remote collaboration. *Communications of the ACM, 34*(7), 40-61. (rr)

The authors developed Liveboard, a large interactive display. It provides a research tool for experiments on group meetings, presentations, and remote collaboration. Survey results with Liveboard users are presented along with recommended improvements. The authors also present general observations on the use of large interactive displays.

Englebart, D. (1992). Toward high-performance organizations: A strategic role for groupware. In David D. Coleman (Ed.), *Groupware 92* (pp. 77-100). San Mateo, CA: Morgan Kaufmann.

Englebart considers groupware a means to an important end: ". . .creating truly high performance human organizations." (p. 77). Englebart discusses the complexity of implementing groupware changes that will play an evolutionary strategy in organizational development. Four groupware architectural requirements proposed by Englebart are global and individual vocabulary control, multiple look and feel interfaces, shared-window teleconferencing and linkage between hyperdocuments and other data systems. A groupwork system called CODIAK that supports these requirements is discussed in detail.

Englebart, D., & Lehtman, H. (1988, December). Working together. *Byte*, pp. 245-252. (rr)

CSCW as defined by Englebart and Lehtman "deals with the study and development of systems encouraging organizational collaboration" (p. 246) of which communication is a vital part. The emergence of CSCW, also known as Groupwork or workgroup computing, into the workplace, will inject computer processes into the middle of the communication process. To do this effectively, users will require tools for composing messages and performing other communication functions. This process must be automatic, have full cataloging, and be accessible on-line.

Using computers as a communication medium and facilitator will provide a cross between oral and written communication methods. This hybrid communication system may have great impact on the workplace and effectiveness of the communication process. One question which should be answered is how CSCW will change the dimensions of the communications' process. Englebart et al. state that "The optimum design for either a tool system or a human system is dependent on the match it must make with the other" (p. 252). Both must evolve in a balanced way.

Eveland, J. D., & Bikson, T. K. (1988). Work group structures and computer support: A field experiment. *ACM Transactions on Office Information Systems*, *6*(4), 354-379. (rr)

Eveland and Bikson performed a set of field experiments that tested the hypothesis that work groups who use computer technologies develop different structures and processes than groups that rely on more conventional techniques. One of the more significant results showed that members of electronic groups had more involvement in the work of the group and were more satisfied with the outcome.

Galegher, J., & Kraut, R. E. (1990). Computer-mediated communication for intellectual teamwork: A field experiment in group writing. In F. Halasz (Ed.), *CSCW 90: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 65-78). Los Angeles, CA: ACM Press. (rr)

To work on shared projects, people must agree on a set of goals, coordinate actions, and form the components into a unifies whole. These activities form the basis for intellectual teamwork. Variations in the process over its lifetime indicate that different communication modalities may be useful at different times. The authors have investigated these different communication types and results confirm their expectations. In general, computer-mediated groups work harder and longer than their counterparts who meet face-to-face; however, new projects were harder to start using computer-mediated methods.

Galegher, J., & Kraut, R. E. (1992). Computer-mediated communication and collaborative writing: Media influence and adaptation to communication constraints. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 155-162). Toronto, Canada: ACM Press. (rr)

The authors studied a collaborative writing task to determine how individuals adapt their behavior when faced with obstacles in the communication environment. Results support the validity of contingency theory and points out the human potential for adaptation when faced with extreme limitations on interactivity.

Ganssele, J. G. (1994, February). Data compression. *Embedded Systems Programming*, pp. 85-87. (rr)

Discusses the pros and cons of various data compression techniques. Lossless and lossy compression are included.

Garry, G. (1993). Groupware, OSes in software overhaul. *Digital News & Review*, pp. 1, 60.

DEC proposes to utilize LinkWorks as a "framework" (p. 1) for groupware. The author describes functions included in DEC Linkworks for Microsoft Windows.

Gaver, W. W. (1992). The affordances of media spaces for collaboration. In J. Turner & R. Kraut (Eds.), *Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 17-24). Toronto, Canada: ACM Press. (rr)

Gaver addresses and contrasts how media spaces are supplied for collaboration. The implications for perception and social interaction are also analyzed. How a media space environment offers the actions and interactions to those within it is important and may suggest possibilities for design of collaborative systems.

Gaver, W. W., Sellen, A., Heath, C., & Luff, P. (1993). One is not enough: Multiple views in a media space. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (Eds.), *INTERCHI '93: Conference Proceedings on Human Factors in Computing Systems* (pp. 335-341). Amsterdam, The Netherlands: ACM Press. (rr)

Although media spaces support collaborative work, limited access for remote users can compromise their usefulness. The authors ran highly visual task experiments between two remote offices. They allowed the users to see each other on two different video camera views. The authors contended that these views would increase the ability to collaborate compared to just head and shoulder views. Results showed that the users successfully collaborated on the tasks using the multiple cameras. Strengths and weaknesses of this strategy are discussed along with recommendations for expanding media spaces and increasing access to remote sites.

Gershman, A., & Sato, S. (1994). Multimedia and groupware: A proposed situational approach to selecting communication systems for collaboration. In David D. Coleman (Ed.), *Groupware 94* (pp. 599-604). Scottsdale, AZ: The Conference Group.

This paper proposes a framework that maps communication systems against group business interactions. The authors discuss a framework to help define what group communication products are most useful and productive.

Gerwirtz, D. (1994). *Lotus Notes 3: Revealed.* Rocklin, CA: Prima.

This book describes Lotus Notes 3. It is designed to help information managers and network administrators determine how Lotus Notes fits into their organization.

Gilbert, G. N. (1993). CSCW for real: Reflections on experience. In D. Diaper and C. Sanger (Eds.), *CSCW in practice: An introduction and case studies* (Chap. 4, pp. 39-50). New York: Springer-Verlag. (rr)

This chapter discusses CSCW and shows that it is now possible using existing facilities. The author also describes case studies of good and bad groupware experiences. The author is surprised that CSCW has been possible for over the past ten years; however, it is rarely practiced today. He investigates why this is so.

Gery, G. J. (1991). *Electronic performance support systems: How and why to remake the workplace through the strategic application of technology.* Boston, MA: Weingarten.

The author shows how currently available technology can be used to improve productivity in an organization.

Glickman, J., & Kumar, V. (1993). A SHARED collaborative environment for mechanical engineers. In D. Coleman (Ed.), *Groupware '93* (pp. 335-347). San Mateo, CA: Morgan Kaufmann.

A mechanical engineering groupware project called SHARE is providing tools for computer-based collaboration. These tools will assist the engineering process by capturing and manipulating information that is now processed in other ways.

Goetsch, D. L. (1986). *CADD: Understanding computer-aided design and drafting.* Tulsa, OK: Pennwell.

Explains the evolution, structure, operation, and future development of computer-aided design and drafting systems.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35*(12), 61-70. (rr)

The motivation for Tapestry, an experimental mail system, comes from the increasing use of electronic mail that results in users being inundated with information. Goldberg, Nichols, Oki, and Terry, of the Xerox Palo Alto Research center, have developed a series of filters that allow list scanning for document selection. This process supports both con-

tent and collaborative filtering and allows individuals to tag their reactions to documents. These reactions called annotations can be accessed by other filters and would be useful in a workgroup context. Goldberg et al. discuss the flow of information through the Tapestry system as well as typical user scenarios.

Goldstein, A., & Healey, D. L. (1990). *Peterson's business and management jobs* (Sixth edition). Princeton, NJ: Peterson's Guides.

Used as a source for workgroup work area designations.

Greenberg, S. (1991). Computer-supported cooperative work and groupware. In S. Greenberg (Ed.), *Computer-Supported Cooperative Work and Groupware* (pp. 1-7). London: Harcourt Brace Jovanovich.

Defines workgroup computing. Includes a compilation of writings that explore CSCW and groupware. Includes a detailed annotated bibliography listing CSCW research to 1991.

Greenberg, S., Roseman, M., Webster, D. & Bohnet, R. (1992). Issues and experiences designing and implementing two group drawing tools. *Proceedings of the 25th Annual Hawaii International Conference on the System Sciences* (pp. 139-150). Hawaii: IEEE Computer Society Press.

The authors describe issues and hurdles that surfaced during the design of two real-time, multi-user drawing systems, Groupsketch and Groupdraw. Differences between the single and multiple user applications were detailed. Based on design and implementation results there was no compelling reason to use a centralized versus replicated architecture. In addition, the use of moderate speed local area networks was not a problem.

Greenwood, E. (1992). Personal calendaring & group scheduling: An extension of the electronic mail system. In David D. Coleman (Ed.), *Groupware 92* (pp. 385-387). San Mateo, CA: Morgan Kaufmann.

This article describes personal calendaring and group scheduling functions. It explains why they are really an extension of electronic mail.

Greif, I. (1992). Designing group-enabled applications: A spreadsheet example. In David D. Coleman (Ed.), *Groupware 92* (pp. 515-525). San Mateo, CA: Morgan Kaufmann.

The author believes that the next generation of workgroup products will come from the union of communication and data sharing capabilities with desktop tools. Both groupware tools and application software must change for this to happen. This paper describes a case study for the design of a spreadsheet example that responds to the need for an integrated product.

Greif, I. (1988). Overview. In I. Greif (Ed.), *Computer-supported cooperative work: A book of readings* (pp. 5-12). San Mateo, CA: Morgan Kaufmann.

The author provides background on CSCW and the field research being conducted.

Greif, I., & Sarin, S. (1988). Data sharing in groupwork. In I. Greif (Ed.), *Computer-supported cooperative work: A book of readings* (pp. 477-508). San Mateo, CA: Morgan Kaufmann.

A major concern in CSCW is how shared data can be accessed. This paper defines the scope of data sharing requirements for CSCW systems. three cooperative work systems are discussed along with limitations of implementation techniques. Features for modifying roles and working relationships are discussed. The authors conclude that due to the diversity of programs investigated their requirements are representative of a wide range of workgroup applications.

Gronbeck, K., Kyng, M., & Mogensen, P. (1992). CSCW challenges on large-scale technical projects--A case study. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 338-345). Toronto, Canada: ACM Press. (rr)

The authors investigate CSCW aspects of large-scale technical projects in a Danish engineering company. Problems and bottlenecks in daily work and collaboration were uncovered. Examples are sharing materials, issuing tasks, and keeping track of task status. Workshops uncovered several challenges for implementation of collaborative organizational structure. Integration of massive amounts of diverse information into a multimedia database, quickly changing plans, and task/material coordination are all discussed in detail in this case study.

Gronbeck, K., Kyng, M., & Mogensen, P. (1993). CSCW challenges: Cooperative design in engineering projects. *Communications of the ACM, 36*(4), 67-77. (rr)

The authors continue their investigations of CSCW aspects of large scale technical projects. The cooperative design process using a Coordination Tool and a Cooperative Hypermedia Tool is presented. Challenges to this process include hypermedia support for shared materials and coordination support. The authors believe that requirements for hypermedia and coordination tools can be generalized to most CSCW systems.

Grudin, J. (1988, December). Perils and pitfalls. *Byte*, pp. 261-264. (rr)

Grudin discusses the perils and pitfalls of groupware and he shows that group interaction, including communication, is extremely complex. Grudin explains that there are many hurdles that must be cleared before you can succeed with Groupware. Implementation can fail if proper attention is not paid to the subconscious things that help to implement a CSCW system. Grudin goes on to discuss various perils and pitfalls in the Groupwork implementation process. Some of Groupware's successes are discussed along with requirements for evaluating Groupware projects.

Grudin, J. (1991a). CSCW: The convergence of two development centers. In S. P. Robertson, G. M. Olson and J. S. Olson (Eds.), *CHI '91 Conference Proceedings* (pp. 91-97). New Orleans, LA: ACM Press.

Research into CSCW and development of groupware represent separate and converging interests in the collaborative computing area. Group dynamics issues and organizational impact have been explored during the development of systems for organizations. These issues have not been adequately researched for small groups. Grudin contends that in order to effectively integrate these experiences, interests, and approaches ". . .we have to go beyond what is shared and explore the differences." (p. 91). As these development areas converge they are creating a common language that will allow developers to gauge their position in a rapidly changing systems world.

Grudin, J. (1991b). CSCW introduction. *Communications of the ACM, 34*(12), 31-34.

Grudin's emphasis in this more recent paper is on the collaboration of groups. Grudin observed that people interact continually without much effort. Grudin comments that "Computer collaboration is difficult and requires a much better understanding of the way groups and organizations function" (p. 32). There must be a better understanding of group processes for workgroup computing systems to be successfully implemented. Grudin's views can help to pinpoint areas of group and organizational interaction that should be considered when developing workgroup computing systems.

Grudin, J. (1994). Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM, 37*(1) 90-106.

Grudin discusses differences between groupware and traditional computer support. He outlines the origins of groupware, discusses eight problem areas, and examines groupware successes.

Gutag, K., Gove, R. J., & Van Aken, J. R. (1992, November). A single-chip multiprocessor for multimedia: The MVP. *IEEE Computer Graphics & Applications*, pp. 5-64. (rr)

Today's user interfaces represent the merging of the separate text and graphics systems of the 1970s and 1980s. New approaches are required to support graphics intensive multimedia applications. This new computer chip architecture will support parallel processing techniques and data throughput of up to 2 billion operations per second. The architecture focuses on document image processing, image generation, and compression techniques. The authors expect this technology to transform not only the desktop environment but also digital copiers, video-conferencing systems, transaction processing, and security systems.

Harrison, B. T. (1994). Layer upon layer: The world of distributed APIs. *DEC Professional*, pp. 30-36.

This article discusses and architecture for networking application programming interfaces (APIs). These APIs allow local and wide area networks to interface seamlessly. Various API layers perform different functions.

Harrison, W. H., Ossher, H., & Sweeney, P. F. (1990). Coordinating concurrent development. In F. Halasz (Ed.), *CSCW 90: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 157-168). Los Angeles, CA: ACM Press. (rr)

Large systems development requires the coordination of many developers. These activities can occur concurrently. The goal of coordination is to enhance developer productivity while insuring concurrent development activities run smoothly. The authors present a formal model of concurrent development that includes modification and merge activities. Protocols for concurrent development are presented.

Heath, C., & Luff, P. (1991). Disembodied conduct: Communication through video in a multi-media office environment. In S. P. Robertson, G. M. Olson and J. S. Olson (Eds.), *CHI '91 Conference Proceedings* (pp. 99-103). New Orleans, LA: ACM Press. (rr)

This paper discusses research concerning video-mediated group work in an office environment. Heath and Luff video-recorded group interactions and then analyzed social interactions including verbal and non-verbal conduct. How technology transforms these processes is also discussed. The authors' goal is to invigorate aspects of visual conduct normally missing from a non-visual groupwork environment.

Held, J. J. (1992). Groupware in investment banking: Improving revenue and deal flow. In David D. Coleman (Ed.), *Groupware 92* (pp. 461- 464). San Mateo, CA: Morgan Kaufmann. (rr)

Lotus Notes, with several add-on products, was used in this large banking organization to resolve group collaboration problems. The system has been successfully expanded to over 100 users at multiple locations. Even senior management who did not have computers before this system was implemented are using the system.

Heldman, R. K. (1993). *Future telecommunications: Information applications, services, and infrastructure*. New York: Mcgraw-Hill.

Provides excellent background information on telecommunication trends. Covers networking, services, and infrastructure.

Higgins, S. (1992). Groupware: Getting a grip on work-group computing. *PCWeek Special Report*, pp. 1, 14.

The term "groupware" is difficult to define and has been used to describe simple e-mail programs to complicated work-flow-automation software. Many experts do not think groupware should be considered a distinct category of software. Higgins explains that industry analysts recognize applications that deserve to fall under the groupware category. He discussed the six categories of groupware including messaging systems, database-enabled applications, message filtering applications, work group application development software, calendaring and scheduling products, and document management and imaging systems.

Hiltz, S. R., & Turoff, M. (1985). Structuring computer-mediated communication systems to avoid overload. In D. Marca and G. Bock (Eds.), *Groupware: software for computer-supported cooperative work* (pp. 384-393). Los Alamitos, CA: IEEE Computer Society Press. (rr)

Computer-mediated communications use computers and networks to compose, store, deliver, and process communication. Unless these systems are structured correctly they will be overloaded by users. Structure should be imposed by individuals and user groups according to their needs and abilities, rather than through general software features.

Holsapple, C. W., Rathnam, S., & Whinston, A B. (1993). Groupware. In A. Ralston. and E. D. Reilly (Eds.), *Encyclopedia of computer science* (Third Edition), (pp. 588-589). New York: Van Nostrand Reinhold. (rr)

Provides a concise description of groupware and possible applications.

Holtham, C. (1993). Group communications, management processes and groupware tools. In D. Coleman (Ed.), *Groupware '93* (pp. 292-303). San Mateo, CA: Morgan Kaufmann.

The author identifies five workgroup drivers: change, coordination, collaboration, control, and connectivity. A new grid provides a way of classifying groupware tools. The author argues that "bundles" (p. 292) of different groupware tools are required to support different business situations.

Hsu, J., & Lockwood, T. (1993, March). Collaborative computing: Computer aided teamwork will change your office culture forever. *Byte*, pp. 113-120.

Collaborative computing systems are environments that allow people to share information. Hsu and Lockwood discuss the three fundamental aspects of collaborative computing systems: common task, shared environment, and time/space. Problems with groupware are discussed. These systems can meet resistance when they are introduced because they challenge the existing organizational culture. Organizations must change the way they operate for collaborative systems to be successful.

Huber, G. (1990). A theory of the effects of advanced information technologies on organizational design, intelligence, and decision making. *Academy of Management Review, 15* (1), 47-71.

Huber directly addresses the forms of communication and the effects of these technologies can have on change in the workplace. He explains the nature of advanced information technologies in terms of several basic characteristics, one of which is communication. Precision, rapidity, and access control are also factors which Huber says must be considered. Huber reasons that use of these advanced technologies will not usurp traditional technologies unless organizational effectiveness can be increased. Organizations will adopt only those technologies which achieve their purposes.

Huber discusses the effects of computer-assisted communication on decision-making. Although there will be less face-to-face communication, managers and professionals can choose the medium which best fits the task and required decisions. It is also possible that the decision-making level will shift to a lower organizational level, since personnel will have the data necessary to make the appropriate decision.

Ibbs, W. C. (1989). Relational decision making and communications for project management . In R. L. Kimmons & J. H. Loweree (Eds.), *Project management: A reference for professionals* (chap. XIV, 6, pp. 1059-1075). New York: Marcel Dekker. (rr)

Recent advances in computer and electronic communication are reviewed. Ibbs looks at how advanced communication, software tools, and management information systems might be applied to project management; however, they can be extrapolated to other technical and program functions. Ibbs concludes that the ability to transmit information will be an important factor in the success of a project.

Jay, A. (1976). How to run a meeting. *Harvard Business Review, 54*(2), 43-57.

This article, written before desktop personal computers, furnishes essential background on how a meeting can be structured and run. It discusses meeting functions and objectives.

Jessup, L. M., & Valacich, J. S. (1993). *Group support systems: New perspectives.* New York: Macmillan Publishing Company.

A compilation of current research on Group Support Systems (GSS). Discusses work at nine research sites: the University of Arizona; the University of Minnesota; the University of Georgia, Indiana University; New Jersey Institute of Technology; Queen's University; the University of Michigan; the Claremont Graduate School; and the University of Hohenheim.

Johansen, R., Sibbet, D. Benson, S., Martin, A., Mittman, R., & Saffo, P. (1991). *Leading Business Teams*. New York: Addison-Wesley.

This book explores the emerging area of groupware. The authors chart the terrain, discuss building blocks, and provide a perspective on how to turn groups into teams. Examples show how groupware can be applied in actual business situations.

Johnson, B. (1992). Introducing workgroup technology: "How to build momentum and success". In David D. Coleman (Ed.), *Groupware 92* (pp. 117-129). San Mateo, CA: Morgan Kaufmann. (rr)

This paper discusses the issues that must be addressed when introducing groupwork technology to an organization. The authors focus on how corporate proponents and software vendors can enhance product acceptance and use. There is a need to build group-oriented issues to build positive awareness, alleviate concerns, focus interest, and optimize group product evaluations.

Johnson-Lenz, P., & Johnson-Lenz, T. (1991). Post-mechanistic groupware primitives: Rhythms, boundaries and containers. In S. Greenberg (Ed.), *Computer-supported* *cooperative work and groupware* (Chap. 2, pp. 11-28). London: Harcourt Brace Jovanovich. (rr)

The authors are exploring several different approaches to groupware in order to find a middle path. The first approach is to make groups work through the use of explicit forms and procedures. The second approach is to allow groups to self organize. Groupware must be capable of being tailored for changing needs and evolving purposes. Purpose centered groupware has the potential to be applied to any organizational group and it is necessary for transition to a sustainable culture.

Johnston, S. (1993, October). Open APIs in Windows 4.0 will integrate applications. *Infoworld*, p. 10.

Discusses the open APIs that will integrate applications into the Windows 4.0 environment.

Kaplan, A., Lauriston, R., & Fox, S. (1992, March). Groupware. *PC World*, pp. 209-214. (rr)

Kaplan, Lauriston, and Fox reviewed twenty-eight different Groupware products. Their review included newer versions of some of the older products discussed by Opper (1988). Kaplan et al. also provide guidelines concerning what to look for in personal computer groupware products.

Kaplan, S. M., Carroll, A. M., & MacGregor, K. J. (1991). Supporting collaborative processes with ConversationBuilder. In P. de Jong (Ed.), *Conference on Organizational Computing Systems*, (pp. 69-79). Atlanta , GA: ACM Press. (rr)

The authors developed a groupware program called ConversationBuilder. The product currently has an installed base at the University of Illinois on their UNIX system. ConversationBuilder is a collaborative open system that can be tailored to support various group activities. This paper discusses the collaborative process and the basis for ConversationBuilder. Research done by Kaplan et al. is significant because it looks at group needs. The developed program allows more than one way to complete a task and users can specify the level of collaboration.

Kaplan, S. M., Tolone, W. J., Bognia, D. P., & Bignoli, C. (1992). Flexible, active support for collaborative work with ConversationBuilder. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 378-385). Toronto, Canada: ACM Press. (rr)

This article provides an update on progress in developing, ConversationBuilder, a flexible, active product to support workgroup activities. Unlike other active support tools, such as the Communicator, ConversationBuilder allows the specification of multiple protocols and allows users the ability to work with these protocols depending on the situation at hand. The type of work supported by ConversationBuilder are discussed and illustrated through the use of an example. The product is used daily at the University of Illinois and is extremely stable.

Kaplan, S. M., Tolone, W. J., Bognia, D. P., & Phelps, T. (1993). Flexible, active support for collaborative work with ConversationBuilder. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (Eds.), *INTERCHI '93: Conference Proceedings on Human Factors in Computing Systems* (p. 248). Amsterdam, The Netherlands: ACM Press. (rr)

This was an update and demonstration of the ConversationBuilder program at the INTERCHI conference. The system now has shared hypertext with hyperlinks. In addition, several new tools including construction of dynamic user interfaces and a message bus for tool interconnection have been added to the program.

Kawell, L. Jr., Beckhardt, S. Halvorsen, T., & Ozzie, R. (1988). Replicated document management in a group communication system. In D. Marca and G. Bock (Eds.), *Groupware: Software for computer-supported cooperative work* (pp. 226-235). Los Alamitos, CA: IEEE Computer Society Press. (rr)

This paper discussed the design and implementation of a replicated database to support the Notes group communication system. The system supports groups of people working on

shared documents and is intended for use in a networked environment. The authors characterization of this class of applications suggests that this technique can be applied to other groupwork systems such as computer conferencing and bulletin board systems.

Kedzierski, B. I. (1988). Communication and management support in system development environments. In I. Greif (Ed.), *Computer-supported cooperative work: A book of readings* (pp. 253-268). San Mateo, CA: Morgan Kaufmann.

The author proposes an environment that supports integrated capabilities for project management, system eveluation, documentation/help, and intelligent communiction between design users. The work is based on a theory of "communication acts" (p. 253) such as questioning, griping, planning, and requesting.

Kerr, S., & Hurwicz, M. (1993). Broadband decisions: ATM: Ultimate network or ultimate hype & FDDI: Not fastest but still fit. *Datamation*, pp. 30-36. (rr)

The authors discuss the two primary contenders for high-speed network transmission during the 1990's and beyond. ATM promises to be a high-bandwidth, relatively simple technology that may oust competitve networking approaches. FDDI appears to have two critical advantages over ATM: lower prices and tested interoperabiltiy. These companion articles compare the two approaches head-to-head and allow the reader to reach his own conclusions about the future of these competing technologies.

Kerzner, H. (1992). *Project management: A systems approach to planning, scheduling, and controlling* (Fourth Edition). New York: Van Nostrand Reinhold.

In chapter 17 of this book, Kerzner discusses computerized aids for project managers. Project management software features include planning, tracking, monitoring, report generation, project calendars, what-if analysis, and multi-project analysis. Kerzner addresses workgroup applications.

Kling, R. (1991). Cooperation, coordination, and control in computer supported cooperative work. *Communications of the ACM, 34*(12), 83-84. (rr)

Kling states that "CSCW may be seen as a conjunction of certain kinds of technologies, . . . certain kinds of users. . ., and a worldview that emphasizes convivial work relations" (p. 83). He differentiates CSCW from other forms of computerization and he feels that CSCW is a "computer-based social movement rather than a family of technologies" (p. 84).

CSCW implementation progress during the last five years has been slow. There have been a few CSCW successes, notably E-mail and computer conferencing. Many prototype systems have been produced; however, there have been few commercial ones. Kling's assessment is that it may be difficult to implement CSCW because the dynamics of group social processes are not well understood.

Knister, M. J., & Prakash, A. (1990). DistEdit: A distributed toolkit for supporting multiple group editors. In F. Halasz (Ed.), *CSCW 90: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 343-355). Los Angeles, CA: ACM Press.

The authors developed a toolkit, called DistEdit, that allows building of interactive group editors for distributed environments. Discussions include toolkit evaluation and future work.

Khoshafian, S., Baker, B. A., Abnous, R., & Shepherd, K. (1992). *Intelligent offices: Object-oriented information management in client/server architectures.* New York: John Wiley & Sons, Inc.

Provides a comprehensive description of the computing components necessary to create the next generation of "intelligent offices." Combines digital imaging, database management, data storage, and networking for information accessibility.

Krantz, L. (1992). *The job rater almanac* (2nd edition). New York: World Almanac.

Used as a source for workgroup work area designations.

Kreifelts, T., & Prinz, W. (1993). ASCW: An assistant for cooperative work. In S. Kaplan (Ed.), *Conference on Organizational Computing Systems* (pp. 269-278). Milpitas, CA: ACM Press. (rr)

Knister, M. J., & Prakash, A. (1990). DistEdit: A distributed toolkit for supporting multiple group editors. In F. Halasz (Ed.), *CSCW 90: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 343-355). Los Angeles, CA: ACM Press.

The authors developed a toolkit, called DistEdit, that allows building of interactive group editors for distributed environments. Discussions include toolkit evaluation and future work.

Khoshafian, S., Baker, B. A., Abnous, R., & Shepherd, K. (1992). *Intelligent offices: Object-oriented information management in client/server architectures*. New York: John Wiley & Sons, Inc.

Provides a comprehensive description of the computing components necessary to create the next generation of "intelligent offices." Combines digital imaging, database management, data storage, and networking for information accessibility.

Krantz, L. (1992). *The job rater almanac* (2nd edition). New York: World Almanac.

Used as a source for workgroup work area designations.

Kreifelts, T., & Prinz, W. (1993). ASCW: An assistant for cooperative work. In S. Kaplan (Ed.), *Conference on Organizational Computing Systems* (pp. 269-278). Milpitas, CA: ACM Press. (rr)

The Assistant for Cooperative Work is a system for managing cooperative work. It includes a task manager, information system, and video conference tool. The model is flexible and task oriented. Synchronous communication provides access using standard message handling standards.

Krill, P. (1993). Software emphasis is on groupware and mail. *Open Systems Today*, pp. 89-90.

Digital Equipment Corporation (DEC) has introduced groupware software called LinkWorks Client for Microsoft Windows, Version 2.1. This article provides a description of Linkworks including information administration and management capabilities, document sharing, access control, electronic mail, and automation of workflow activities. LinkWorks is suited to tightly integrated workgroups.

Kundargi, K., & Subramaniam, R. (1992). Practical aspects of implementing computer supported collaboration (csc) vertical solution: Residential real estate transactions. In David D. Coleman (Ed.), *Groupware 92* (pp. 533-536). San Mateo, CA: Morgan Kaufmann. (rr)

The authors describe experiences designing and implementing a computer based environment to improve business collaborations in residential real estate transactions. The resulting system is server-based network of personal computers. The authors chose to use a graphical user interface specifically designed for real estate transactions instead of off-the-shelf groupware products.

Kuutti, K., & Arvonen, T. (1992). Identifying potential cscw applications by means of activity theory concepts: A case example. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 233-240). Toronto, Canada: ACM Press. (rr)

Kuutti and Arvonen use activity theory to identify potential CSCW applications. A 3x6 support type classification is formed and studied using a real work situation. Possible areas of support are defined and problems identified. The "shared material" (p. 240) metaphor used by Kuutti et al. may be too broad because it can be used to represent several distinct purposes.

Kyng, M. (1991). Designing for cooperation: Cooperating in design. *Communications of the ACM, 34*(12), 65-73.

Kyng pays close attention to the collaborative aspects of work in the computerized setting. Kyng discusses cooperation as a factor that must be integrated into computer support efforts. Kyng also perceives that some of these aspects of cooperation have been ignored by the more traditional computer support areas. This lack of cooperation has hindered the introduction of Groupware into organizations.

The importance of information exchange between workers must not be underestimated. Systems must be designed so that they apply both to users and designers. System use must stimulate the flow of knowledge while allowing personnel to apply this knowledge to the design situation. Proper tools must be developed that support active participation in the design process.

Kyng's assessment is that system designers have "a long way to go" (p. 72) before users work with design tools on the computer the way they use pens and other implements. Kyng's work is important for understanding what is required to successfully apply CSCW methods.

Lai, K., & Malone, T. W. (1991). Object lens: Letting end-users create cooperative work applications. In S. P. Robertson, G. M. Olson and J. S. Olson (Eds.), *CHI '91 Conference Proceedings* (pp. 425-426). New Orleans, LA: ACM Press. (rr)

Object Lens is tool that supports groupwork and information management applications. The system provides objects, folders, and agents to coordinate and manipulate processes and actions. Groups can explore and capture the elements of decision making. Argument networks can display group results.

Lakin, F. (1988). A performing medium for workgroup graphics. In I. Greif (Ed.), *Computer-supported cooperative work: A book of readings* (pp. 367-396). San Mateo, CA: Morgan Kaufmann.

Writing and drawing on a common display assists workgroups. There must be a computerized system to more efficiently manipulate text and graphics The system must also refine work and record it for future reference. This paper presents an approach for the design of an appropriate working medium. It discusses new computer examples, key features of text and graphic manipulation, and a proposed architecture.

Lange, B. M. (1992). Electronic group calendaring: Experiences and expectations. In David D. Coleman (Ed.), *Groupware 92* (pp. 428-432). San Mateo, CA: Morgan Kaufmann.

Lange discusses the essential elements for the successful introduction of a groupware product in an organization. The implementers found that there were two highly related sets of factors for the use of groupware. First, the product, electronic calendaring, had to have well defined expected uses and clear guidelines for usage. Second, the organization had to revise procedures as the product was introduced to encourage and stimulate usage.

Laplante, P. A. (1992). *Real-time systems design and analysis*. Piscataway, NJ: IEEE Press.

A reference tool for software engineers. This guide to real-time software covers computer architecture, operating systems, programming languages, software engineering and systems integration.

Larson, R. W., & Zimney, D. J. (1990). *The white collar shuffle: Who does what in today's computerized workplace*. New York: AMACOM. (rr)

The transfer of information can be a motivating force in the business environment. Larson and Zimney discuss this subject and other factors in their book, *The White Collar Shuffle*. Larson et al. also look at the negative effects of computerization which include loss of face-to-face communication, and depersonalization. Segments of this book are important because they show how technology can be intimidating to workers and how there must be an appropriate distribution of work to encourage success.

Lauwers, J. C., & Lantz, K. A. (1990). Collaboration awareness in support of collaboration transparency: Requirements for the next generation of shared window systems. *Proceedings of CHI '90* (pp. 303-311). Seattle, WA: ACM Press. (rr)

The development of shared window systems allows existing applications to be shared. Several limited group implementations have shown the merits of this approach. There are areas such as spontaneous interactions, shared workspace management, annotation, and telepointing that have not been adequately addressed. Lauwers and Lantz define user requirements in these areas. The goal is to provide "collaboration transparent" (p. 303) applications that are "collaboration aware" (p. 303).

Lepick Kling, J. (1992). But i don't talk TCP/IP: Solving the interoperability problem. In David D. Coleman (Ed.), *Groupware 92* (pp. 254-255). San Mateo, CA: Morgan Kaufmann.

The lack of a consistent set of data services across corporate platforms is inhibiting the expansion of groupware. A homogeneous environment such as TCP/IP or a product such as PIPES Platform, an object oriented distributed operating system, might help to mitigate this problem. Lepick Kling discusses interoperability issues associated with implementation of system wide communication networks. She also provides insight into architectural requirements for enterprise wide groupware developments.

Little, T. D. C., & Venkatesh, D. (1994). Prospects for interactive video on demand. *IEEE Multimedia*, *1*(3), 14-23.

Multimedia is a rapidly evolving technology. Little and Venkatesh survey the technological considerations for designing large-scale, interactive, distributed systems. Problems and implementation issues are discussed.

Lyles, B. (1993). Media spaces and broadband isdn. *Communications of the ACM*, *36*(1), 46-47. (rr)

The deployment of high-speed networks will open a new era for media spaces by improving video and audio quality. The new ISDN-B standard has the potential to provide universal access and scalable bandwidths. Asynchronous Transfer Mode (ATM) will provide features such as ". . .global connectivity. . .", ". . .tens of Mb per second. . .", and ". . .aggregate bandwidth measured in tens, hundreds or thousands of Gb per second." (p. 47).

Madsen, C. M. (1989). Approaching group communication by means of an office building metaphor. In D. Marca and G. Bock (Eds.), *Groupware: Software for computer-supported cooperative work* (pp. 316-328). Los Alamitos, CA: IEEE Computer Society Press. (rr)

An environment for cooperative work is described by the author. The conceptual framework is approached by way of a transaction cost theory of organizations. The authors argue that this approach corresponds to a broad variety of cooperative work settings. A cooperative work information system is designed using an office building metaphor. The authors see cooperative work a processes characterized by communication and by opportunistic behavior among participants.

Madsen, K. H., & Aiken, P. H. (1993). Experiences using cooperative interactive storyboard prototyping. *Communications of the ACM, 36*(4), 57-64. (rr)

This workgroup approach was inspired by Scandanavian research into cooperative design. Users and developers are interactively involved in the design process. The system allows consensus and design closure via an interactive systems concept. Development review times are shortened and the final products are less prone to errors and omissions.

used to design CSCW tools. The paper describes what is meant by coordination theory and shows how previous CSCW work can be interpreted in light of this theory. Malone and Crowston then suggest ways to develop the theory further by proposing tentative solutions for coordination. Additional analysis is performed.

Malone, T. W., & Crowston, K. (1990). What is coordination theory and how can it help design cooperative work systems? In F. Halasz (Ed.), *CSCW 90: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 357-370). Los Angeles, CA: ACM Press. (rr)

Presents one perspective on the study of coordination and how this theory may be used to design CSCW tools. The paper describes what is meant by coordination theory and shows how previous CSCW work can be interpreted in light of this theory. Malone and Crowston then suggest ways to develop the theory further by proposing tentative solutions for coordination. Additional analysis is performed.

Malone, T. W., Grant, K. R., Lai, K. Y., Rao, R., & Rosenblitt, D. A. (1989). The information Lens: An intelligent system for information sharing. In M. H. Olson (Ed.), *Technological support for workgroup collaboration* (pp. 65-88). Hillsdale, NJ: Lawrence Erlbaum.

This article describes a prototype information sharing system called Information Lens. It discusses Information Lens' information sharing capabilities. The author also shows how information sharing can be used to support task training and meeting scheduling.

Malone, T., Lai, K., & Fry, C. (1992). Experiment with Oval: A radically tailorable tool for cooperative work. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 289-297). Toronto, Canada: ACM Press.

This cooperative work tool is "radically tailorable" (p. 289). Users can create applications by combining objects, views, agents, and links in a building block approach. Oval functions much like The Coordinator, Lotus Notes, and Information Lens. Established primitives are used as a tailoring language for application construction.

Mantei, M. M., Baecker, R. M., Sellen, A. J., Buxton, W. A. S., & Milligan, T. (1991). Experiences in the use of a media space. In S. P. Robertson, G. M. Olson and J. S. Olson (Eds.), *CHI '91 Conference Proceedings* (pp. 203-208). New Orleans, LA: ACM Press. (rr)

Media spaces allow groups to work together through the use of computers, video, and audio. The authors' media space, CAVECAT (Computer Audio Video Enhanced Collaboration and Telpresence), allows small groups to perform collaborative work without leaving their offices. Experiences during implementation and use are discussed. The psychological

and social aspects of the system are also addressed. Implications for the design of future systems are presented.

Marshak, R. T. (1992). Requirements for workflow products. In David D. Coleman (Ed.), *Groupware 92* (pp. 281-285). San Mateo, CA: Morgan Kaufmann.

Workflow software can be considered to be a type of groupware. This paper provides background information and measurement criteria for workflow software.

McGrath, J. E., & Hollingshead, A. B. (1993). Putting the "group" back in group support systems: Some theoretical issues about dynamic processes in groups with technological enhancements. In L.M. Jessup and J. S. Valacich (Eds.), *Group support systems: New perspectives* (pp. 78-111). New York: Macmillan.

The authors provide background on group theory and they attempt to provide balance to the technological implementation of group support systems. Group issues are discussed in detail.

Medford, C. (1993, May). Groupware: Growing and dividing. *Varbusiness*, pp. 85-89. (rr)

This article discusses the trends in groupware. Products include E-mail, calendaring, and group scheduling products. A leading product is Lotus Notes which is a top down

environment for workgroups. Notes is setting the defacto standard for these types of workgroup products.

Messmer, H. (1993). Bellcore, cable execs detail plans for networks of the future. *Network World*, pp. 25-29.

To support networks of the future, communication and cable companies are converging on systems that use fiberoptics and coaxial designs. This article discusses the future of high bandwidth systems that can handle video transmissions.

Minneman, S. L., & Bly, S. A. (1991). Managing a trois: A study of a multi-user drawing tool in a distributed design work. In S. P. Robertson, G. M. Olson and J. S. Olson (Eds.), *CHI '91 Conference Proceedings* (pp. 217-231). New Orleans, LA: ACM Press. (rr)

Minneman and Bly used a shared drawing tool, Commune, to test whether there were any differences between two or three simultaneous users. The study also contrasted the use of audio/video and audio only connections to discover new behaviors associated with these technologies. There were no difficulties with the third user and the audio only interface appeared to meet the interface requirements. Interactions between participants pointed out areas for further study.

National Research Council, Committee on Human Factors, Commission on Behavioral and Social Sciences and Education (1990). *Distributed decision making: Report of a workshop*. Washington, DC: National Academy Press.

This report includes theories, definitions, and models for distributed decision making.

Neuwirth, C. M., Kaufer, D. S., Chandhok, R., & Morris, J. H. (1990). Issues in the design of computer support for co-authoring and commenting. In F. Halasz (Ed.), *CSCW '90: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 183-196). Los Angeles, CA: ACM Press.

The authors report on a project to develop a "work in preparation" editor that will allow co-authoring and commenting to be studied. Issues in designing computer support for these processes have been identified. They include support for social interaction, support for cognitive aspects of authoring, and support for practicality in both types of interaction.

Nunamaker, J. F., Jr., Briggs, R. O., & Romano, N. C., Jr. (1993). Meeting environments of the future. In D. Coleman (Ed.), *Groupware '93* (pp. 125-144). San Mateo, CA: Morgan Kaufmann.

This paper discusses the integration of three levels of workgroup meeting technologies and how they can improve organizational productivity. The paper presents a vision of future meeting environments.

Nunamaker, J. F., Dennis, A. R., Valacich, J. S., Vogel, D. R., & George, J. F. (1993). Group support systems research: Experience from the lab and field. In L.M. Jessup and J. S. Valacich (Eds.), *Group support systems: New perspectives* (pp. 125-145). New York: Macmillan.

This chapter summarizes the major conclusions drawn from laboratory and field research on Group Support Systems (GSS). GSS implementation issues are also addressed.

Nunamaker, J. F., Dennis, A. R., Valacich, J. S., Vogel, D. R., & George, J. F. (1991). Electronic meeting systems to support group work. *Communications of the ACM, 34*(7), 40-61.

Almost all important decisions in organizations are made by groups. Group meetings may have lack of clear goals and focus, lack of participation, and individuals may have hidden agendas. Often meetings end without a clear understanding or record of what was discussed. An Electronic Meeting Support (EMS) system developed by the authors at the University of Arizona attempts to make meetings more productive through the application of information technology. This paper presents results of EMS research and is an excellent source for defining group processes.

Ohkubo, M., & Ishii, H. (1990). Design and implementation of a shared workspace by integrating individual workspaces. In F. Lochovsky and R. B. Allen (Eds.), *Conference on Office Information Systems* (pp. 142-146). Cambridge, MA: ACM Press.

The authors have proposed a system called TeamWorkStation (TWS) as an approach to shared workspace. Design includes integrated real and virtual workspaces, a shared drawing surface, and smooth transitions between individual and shared workspaces. The paper describes design objectives, implementation methods, and experimental results.

Olson, G. M., Olson, J. S. (1991). User-centered design of collaboration technology. In D. Marca and G. Bock (Eds.), *Groupware: Software for computer-supported cooperative work* (pp. 119-141). Los Alamitos, CA: IEEE Computer Society Press. (rr)

User needs and capabilities should be used as the focus when designing groupware. The authors present user-centered system design concepts that consist of observation and analysis of users at work, assistance in design from relevant aspects of theory, and iterative testing with users. Extensive studies were done of designers at work and these helped to develop the beginnings of a theory of distributed cognition that can form the basis for the first stages of iterative testing and redesign of a prototype shared editor to support design work.

ON Technology Corporation (1993). *Meeting Maker XP*. Cambridge, MA: Author.

User's manual for Meeting Maker XP. Performs group scheduling across Windows, DOS, and Macintosh computer nertworks.

Opper, S. (1988, December). A groupware toolbox. *Byte*, pp. 275-282. (rr)

This article reviewed first generation groupware products which included several that promoted work group communication management. The factor that had the most impact was connectivity since workgroups do not work in isolation.

Products such as The Communicator are designed to afford easy access to the computer system. In addition, the more effective products attempt to change the way people work by nudging them to make decisions concerning recommended actions. Opper stresses that future decision times will be shorter and work will be accomplished by smaller groups. Opper emphasizes that when these types of systems allow people to do work they have never done before, in a more synergistic fashion, then groupware will be truly useful. This article provides insight on the types of Groupware tools that are available for implementation.

Opper, S. (1992). Can groupware enhance productivity and offer competitive advantage? In David D. Coleman (Ed.), *Groupware 92* (pp. 21-24). San Mateo, CA: Morgan Kaufmann. (rr)

The author stresses that the "time to get serious about groupware's real benefit." (p. 21) has come. Clear benefits to the organization must be identified to promote future interest. When groupware solutions are implemented in an organization, base-line interviews should be conducted to determine who will use the groupware and to design applications once groupware has been inaugurated.

Opper, S., & Fersko-Weiss, H. (1992). *Technology for teams: Enhancing productivity in networked organizations*. New York: Van Nostrand Reinhold.

This publication explores groupware uses and implementation issues. The author would like to give "access to the potential power of networked computer technology as a way to transcend limited resources and bottlenecked communications" (p. xvii).

Orlikowski, W. J. (1992). Learning from notes: Organizational issues in groupware implementation. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 362-369). Toronto, Canada: ACM Press. (rr)

Orlikowski undertakes an exploratory field study of the impact of groupware on organizational effectiveness. Lotus Notes was implemented in a large office organization to determine the changes the collaborative process had on social interactions. Results suggest that groupware caused an interaction between the cognitive and structural aspects of the office environment causing people to assess and think about the value of new technology. People act towards technology based on their understanding of it. It may be necessary to change individuals' technological framework to accommodate new technology. Although the introduction of Notes did not change the way people work in this office environment, it has positioned them to take advantage of future technology initiatives.

Orr, J. D. (1992). Graphics and groupware: Increasing intimacy through broadening bandwidth. In David D. Coleman (Ed.), *Groupware 92* (pp. 73-76). San Mateo, CA: Morgan Kaufmann. (rr)

How well people communicate is a function of the bandwidth of the communication paths. How much data can be exchanged in a unit of time is a central issue in groupware support. Orr discusses the goals of groupware that include a medium for communication, metacommunication, audit trails, and security. The addition of graphics to communication will increase the intimacy of those communications and help reach acceptance of groupware by larger portions of the user community.

Palermo, A. M., & McCready, S. C. (1992). Workflow software: A primer. In David D. Coleman (Ed.), *Groupware 92* (pp. 155-159). San Mateo, CA: Morgan Kaufmann.

Palermo and McCready define and discuss types of workflow software. Workflow software can be used by both individuals and groups to manage a wide variety of business processes in and integrated environment. Workflow software provides a language and tools for structuring work processes.

Pastor, E., & Jager, J. (1992). Architectural framework for CSCW. In R. J. D. Power (Ed.), *Cooperation among organizations: The potential of computer supported cooperative work* (pp. 103-118) (Project 5660 Research Report, PECOS, Volume 1). Berlin:

Springer-Verlag.

Chapter 6 of this book describes an architectural framework for cooperative work.

Patel, D., & Kalter, S. D. (1992). A toolkit for synchronous distributed groupware applications. In David D. Coleman (Ed.), *Groupware 92* (pp. 225-227). San Mateo, CA: Morgan Kaufman. (rr)

This paper describes a framework for development of concurrent groupware applications. Features common to all applications are included in the toolkit. This allows the developer to concentrate on higher-level issues concerning collaboration and functionality.

Pederson, E. (1994). Taligent meshes multiple personalities. *Midrange Systems*, pp. 36-37.

Discusses capabilties and structure of the Taligent object-oriented operating system. Also, the article discusses the Taligent API structure.

Pederson, E. R., McCall, K., Moran, T. P., & Halasz, F. G. (1993). Tivoli: An electronic whiteboard for informal workgroup meetings. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (Eds.), *INTERCHI '93: Conference Proceedings on Human Factors in Computing Systems* (p. 391-398). Amsterdam, The Netherlands: ACM Press. (rr)

Tivoli is an electronic whiteboard application designed to support workgroups. It runs on Xerox Liveboard which is a large screen, pen-based interactive display. The system provides the functionality of a whiteboard while taking advantage of the computational power of Liveboard to support and augment informal meeting practices. This paper discusses the reasoning behind the development of Tivoli. Development issues are discussed along with the operation of Tivoli 1.0.

Petras, K., & Petras, R. (1993). *Jobs '93.* New York: Prentice-Hall.

Used as a source for workgroup work area designations.

Pinsonneault, A., Kraemer, K. L. (1989). The impact of technological support on groups: An assessment of the empirical research. *Decision Support Systems*, *5*(2), 197-216. (rr)

The study of group meetings provides insight into group processes and the relationship between group cohesion and task performance. Pinsonneault and Kraemer review the empirical research and findings on the impact to technological support of groups. Both Group Decision Support Systems (GDSS) and Group Communication Support Systems (GCSS) are assessed. Results show there is a lack of research on the formal and informal factors of groups and how technology support group communication and interpersonal processes. There is also a lack of research on the structure imposed on groups by the technological supports.

Pituro, M. (1989, August). Groupware: Computer support for teams. *Data Training*, pp. 13-16. (rr)

Piturro is extremely enthusiastic about developing technology that helps people work together. His view is that groupware facilitates teamwork across "time and space" (p. 13). Success, expressed by Piturro "depends on understanding how people work" (p. 14). Piturro discusses the use of several Groupware products ranging from WordPerfect Office to The Coordinator. The Coordinator makes every message a request and is very structured. The program tries to change the way groups work and has been called "Fascistware" (p. 15) and "digital whip" (p. 15) by some users.

Posner, I. R., & Baecker, R. M. (1992). How people write together. In *Proceedings of the Twenty-Fifth Annual Hawaii International Conference on the System Sciences*, Vol. IV (pp. 127-138). Hawaii: IEEE Computer Society Press.

This paper presents a taxonomy of group writing that will help groupware builders understand the collaborative writing process. Discusses the four components including collaboration roles, the writing process, document control, and writing strategies. The paper also lists a set of design requirements for collaborative writing.

Post, B. Q. (1992). Building the business case for group support technology. In *Proceedings of the Twenty-Fifth International Conference on the System Sciences*, Vol. IV (pp. 34-45). Hawaii: IEEE Computer Society Press. (rr)

Post discusses issues concerning the implementation of group support systems relative to group performance and return on investment. Business case variables such as efficiency, quality, effectiveness, customer satisfaction, and decision-making are useful for measuring the potential contribution of group support systems. Post's framework is useful to researchers and implementors trying to deploy these technologies in complex business environments.

Power, R., & Carminati, L. (1992). Computer supported cooperative work. In R. J. D. Power (Ed.), *Cooperation among organizations: The potential of computer supported cooperative work* (pp. 13-25) (Project 5660 Research Report, PECOS, Volume 1). Berlin: Springer-Verlag.

This research report discusses CSCW history, features of CSCW systems, system needs and workgroup concerns. An important point made by the authors is that it may be possible to develop CSCW tools with application to a "common environment" (p. 25) that include shared functions.

Prakash, A., & Knister, M. J. (1992). Undoing actions in collaborative work. In J. Turner & R Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 273-280). Toronto, Canada: ACM Press. (rr)

Many multi-user applications lack undo capabilities. Prakash and Knister propose a general framework for undo that allows for conflict between various users. Undo selections are proposed for actions depending on who performed them, where they originated, or other appropriate criterion. Additional research is required to determine appropriate interfaces for supporting undo in a workgroup environment.

Queen, J. (1993). *Building local area networks with Windows for Workgroups.* San Mateo, CA: M & T Books.

This manual discusses the advantages of Windows for Workgroups. It provides a detailed examination of the program's features.

Rangan, P. V. (1992). Managing multimedia collaboration. In D. Coleman (Ed.), *Groupware 92* (pp. 418-420). San Mateo, CA: Morgan Kaufmann.

The author discusses high bandwidth communications that are paving the way for digital multimedia. This will result in a large number of multimedia collaborative applications. These developments need to take into account the modeling of collaborative semantics, collaborative mechanisms, and techniques for enhancing collaborations.

Reinhard, W., Schweitzer, J., & Volksen, G. (1994, May). CSCW tools: Concepts and architectures. *IEEE Computer*, pp. 28-36.

Approaches for cooperative work vary greatly. The authors developed a taxonomy that could be used to evaluate groupware tools and how they relate to the workgroup computing process. They define application and functional criteria and requirements. Results describe two implementation approaches: multiuser access to shared single user applications and a network of shared data objects.

Resnik, P. (1992). Hypervoice, a phone-based cscw platform. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 218-225). Toronto, Canada: ACM Press. (rr)

During the next few years telephones will be used increasingly for cooperative work applications. Hypervoice is an application generator for phone-based cooperative work applications. Resnik discusses Hypervoice functions and the programming language through the use of specific cases. Resnik concludes that ". . .it is possible and worthwhile to use the telephone as a platform for input and retrieval of semi-structured information objects." (p. 224).

Rhyne, J. R., & Wolf, C. G. (1992). Tools for supporting the collaborative process. *Proceedings of the ACM Symposium on the User Interface Software and Technology* (pp. 161-170). Monterey, CA, New York: ACM Press.

The authors present a model for collaborative processes that includes both synchronous and asynchronous software as submodels. An object-oriented toolkit implements the model and applies it to a pen-based application. According to Rhyne and Wolf "the lack of asynchronous capability in a fully synchronous application manifests itself in an inability for a person to join a collaboration in process. . ." (p. 161). Asynchronous collaboration allows activity by a subgroup or individual to be documented in a permanent record that can be commented on by other members of the group. The authors discuss application of this model to individual users, groups, and teams.

Rimmer, S. (1991). *Mastering CorelDraw 2* (Third Edition). San Francisco, CA: Sybex.

Includes detailed information on CorelDraw for Windows, a graphical drawing program.

Robinson, M. (1991). Computer supported co-operative work: Cases and concepts. In P. R. H. Hendricks (Ed.), *Groupware 1991: The Potential of Team and Organizational Computing* (pp. 59-75), Utrecht, The Netherlands: Software Engineering Research Centre.

Robinson illustrates the content of the field by reviewing first generation CSCW applications. Group authoring, calendar management, meeting scheduling, action co-ordination in organizations, informal conversations, and large meetings are some of the areas covered. Robinson then outlines ten concepts for CSCW that account for the past work in the field. These concepts form an agenda for the research, design, and implementation of future CSCW systems and applications.

Rodden, T. (1993). Technological support for cooperation. In D. Diaper and C. Sanger (Eds.) *CSCW in practice: An introduction and case studies* (pp. 1-22). London: Springer-Verlag.

The author reviews computer technology necessary to support CSCW. Two CSCW approaches exist. One supports the exchange of information between users. The other develops systems that allow the cooperative sharing of information. He considers technologies exploited by each approach. Rodden also discusses iterative meeting systems that allow information sharing and group communication.

Rodden, T., & Blair, G. (1991). CSCW and distributed systems: The problem of control. In L. Bannon, L. Robinson & K. Schmidt (Eds.), *Proceedings of the Second European Conference on Computer Supported Cooperative Work* (pp. 49-64). Amsterdam, The Netherlands: Kluwer Academic.

Rodden and Blair focus on distributed computing and its interrelationship with CSCW. Control was the major problem emerging from this study. The authors contend that "existing approaches to control in distributed systems are inadequate given the rich patterns of cooperation found in CSCW." (p. 49). Recommendations are provided.

Rogers, E. (1992). Ghosts in the network: Distributed troubleshooting in a shared working environment. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 346-355). Toronto, Canada: ACM Press. (rr)

Networks can provide for improved transmission of information and the sharing of resources. A measure of effectiveness for distributed systems is how well the users of a network can coordinate their activities with respect to one another. Rogers examines the interactive processes that take place when there is a breakdown in the network. Socio-cognitive issues of shared understanding, the transmission of knowledge, and distributed problem solving are addressed.

Roseman, M., & Greenberg, S. (1992). Groupkit, a groupware toolkit for building real-time conferencing applications. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 43-50). Toronto, Canada: ACM Press.

Roseman and Greenberg present an approach for design of a real-time groupware toolkit. Three strategies are presented for building the toolkit's components. They are: an extendable, object-oriented run-time architecture; transparent overlays for adding general components to groupware applications; and open protocols to allow the designers a wider range of interface and interaction policies.

Roseman, M., & Greenberg, S. (1993). Building flexible groupware through open protocols. In S. Kaplan (Ed.), *Conference on Organizational Computing Systems* (pp. 279-287). Milpitas, CA: ACM Press.

Flexible groupware is built using open protocols. The authors describe this implementation technique and illustrate three examples: floor control, conference registration, and brainstorming. Issues are also addressed.

Russell, D., & Gangemi, G. T., Sr. (1991). *Computer security basics*. Sebastopol, CA: O'Reilly & Associates.

This book provides an introduction to computer security. It provides details of access controls and communication, network, and encryption security.

Sanderson, D. (1992). The cscw implementation process: An interpretive model and case study of the implementation of a videoconference system. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 370-377). Toronto, Canada: ACM Press. (rr)

Sanderson develops a model of the CSCW implementation process. The model is then applied to the implementation of a videoconference system. Observations are important for the design, testing, evaluation , and effective use of CSCW technology. Areas for future research are proposed.

Sarin, S. K., Abbott, K. R., & McCarthy, D. R. (1991). A process model and system for supporting collaborative work. In P. de Jong (Ed.), *Conference on Organizational Computing Systems* (pp. 213-224). Atlanta, GA: ACM Press.

Sarin, Abbott, and McCarthy present a model for collaborative work that breaks down the process into units of work. The model concept includes flexible routing of work to personnel who will do the work including presentation and manipulation of documents. Their model is implemented as an object oriented network service. For example a document is considered an abstract object which can be manipulated by members of a task group. Another important idea is the automatic performance of a certain level of action when specified events occur. This model is important because it is one view of the collaborative process that can help to explain the group dynamics in a CSCW environment.

Sarin, S., & Greif, I. (1988). Computer-based real-time conferencing systems. In I. Greif (Ed.), *Computer-supported cooperative work: A book of readings* (pp. 397-420). San Mateo, CA: Morgan Kaufmann.

Discusses the use of real-time conferencing systems to support joint work in numerous application areas. This research identifies design and implementation principles for real-time conferences.

Sathi, A., Morton, T. E., & Roth, S. F. (1988). Callisto: An intelligent project management system. In I. Greif (Ed.), *Computer-supported cooperative work: A book of readings* (pp. 269-309). San Mateo, CA: Morgan Kaufmann.

Discusses project management and the Callisto project whose goal was to support management of large projects. Callisto uses a program requiring significant interaction and cooperation between users. Project management tasks are broken down into three areas: activity management, product configuration management, and resource management.

Schael, T., & Zeller, B. (1993). Workflow management systems for financial services. In S. Kaplan (Ed.), *Conference on Organizational Computing Systems* (pp. 142-165). Milpitas, CA: ACM Press.

This paper introduces a functional architecture for workflow management in an Italian bank. It discusses several different functional modules: message handling, data management, and document management.

Schildt, H., Pappas, C.H., & Murray, W. H. (1994a). *Osborne windows programming series: Volume 2, general purpose API functions.* Berkeley, CA: Osborne Mcgraw-Hill.

This volume describes general purpose Windows' application progamming interface (API) functions. APIs are discussed subsystem by subsystem.

Schildt, H., Pappas, C.H., & Murray, W. H. (1994b). *Osborne windows programming series: Volume 3, Special purpose API functions.* Berkeley, CA: Osborne Mcgraw-Hill.

This volume describes special Windows' application progamming interface (API) functions. APIs are discussed subsystem by subsystem.

Schnaidt, P. (1992). *Enterprise-wide networking.* Carmel, IN: SAMS publishing.

This book presents a detailed overview of enterprise-wide networking concepts and detailed networking explanations and examples.

Schulman, M. (1994). *Using Lotus Notes.* Indianapolis, IN: Que.

This manual provides an introduction to the basic features of Lotus Notes. Notes is a leading groupware product that is ideal for sharing databases and electronic mail over networks.

Schwartz, R. (1992). Data access in workgroup environments. In David D. Coleman (Ed.), *Groupware 92* (pp. 237-240). San Mateo, CA: Morgan Kaufmann.

As workgroup computing matures a major area of discussion will be how data is stored, maintained, and accessed in workgroup settings. Schwartz addresses the wide range of issues surrounding this subject. He states that "Close attention should be paid to the underlying technologies and the data delivery models each approach uses so that there is a close fit with the data needs... an organization can support." (p. 240).

Sharples, M. (1993). Adding a little structure to collaborative writing. In D. Diaper & C. Sanger (Eds.). *CSCW in practice: An introduction and case studies* (pp. 51-67). London: Springier-Verlag.

This work describes basic collaborative writing strategies. It suggests simple techniques for structuring and coordinating writing. A case study explores collaborative writing for an academic paper.

Sharples. M., Goodlet, J. S., Beck, E. E., Wood, C. C., Easterbrook, S. M., & Plowman, L. (1993). Research issues in the study of collaborative writing. In M. Sharples (Ed.), *Computer supported collaborative writing* (pp. 9-28). London: Springier-Verlag.

The authors studied collaborative writing issues and indicate that strategies related to writing are hard to uncover and analyze. In addition, it is hard to design computer systems that support these processes. Implementors need to understand the broad issues concerning cognitive and social processes that motivate collaborative writing.

Sheldon, T. (1994, November). MAPI blooms in Chicago. *Byte*, pp. 163-174.

Discusses Microsoft's Messaging API (MAPI) that is set to become the new industry messaging standard.

Shen, H., & Dewan, P. (1992). Access control for collaborative environments. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 51-58). Toronto, Canada: ACM Press.

Existing access models for collaborative writing are not satisfactory. The authors have developed a new model for access control. It associates displayed data with a set of collaborative rights and it provides a scheme for specifying access.

Shu, L., & Flowers, W. (1992). Groupware experiences in three-dimensional computer-aided design. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference* *on Computer-Supported Cooperative Work* (pp. 170-186). Toronto, Canada: ACM Press.

Shu and Flowers developed a graphically rich, three-dimensional computer-aided design tool to study groupware interface issues. Experiments determined that users preferred the simultaneous mode of editing versus turn taking. In addition, independent points of view optimized parallel activity. Experiments led to development of another tool, Viewpoint, that allows effective resolution between contrasting and arbitrary points of view.

Smith, J. B. (1994). *Collective intelligence in computer-based collaboration.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Smith proposes a new approach for CSCW and a research agenda for developing and testing the approach. The book's main emphasis is on parallel processing of user tasks. Smith considers a group to be a distributed information processing system that can have a particular level of awareness and control.

Sobiesiak, R., & Myopoulos, J. (1991). A conceptual modeling approach to authoring-in-the-large for hypertext documents. In P. de Jong (Ed.), *Conference on Organizational Computing Systems* (pp. 225-239). Atlanta, GA: ACM Press.

Hypertext is a promising approach for handling large, complex document sets that are used and maintained over a long period of time. This paper describes a system called ThyDoc that allows knowledge-based software engineering technologies to the document authoring process. ThyDoc treats the authoring process as a knowledge acquisition process that captures formally and informally all of the knowledge needed by authors to design, develop, and maintain large hypertext documents. A prototype ThyDoc model has been developed to illustrate document engineering.

Software Publishing Corporation (1991). *Using Harvard Graphics for Windows: Harvard Graphics for Windows, version 1.0.* Santa Clara, CA: Author.

Includes detailed information on Harvard Graphics for Windows, a graphical presentation and drawing program.

Sproull, L., & Kiesler, S. (1991, September). Computers, networks and work. *Scientific American*, pp. 116-123. (rr)

The authors studied how computer networks can affect the nature of work. Sproull and Kiesler recognize that "Electronic interactions differ significantly from face-to-face exchanges. As a result, computer networks will profoundly affect the structure of organizations and the conduct of work" (p. 166). They also saw networks as devoid of social and contextual clues that regulate group dynamics.

To test their theories, Sproull et al. performed a combination of laboratory experiments and field studies. Their research showed that, in the computer arena, social and contextual clues were either missing or severely lowered. Another observation made by Sproull et al. was that the use of computers allowed the organization to become much more flexible and less structured. Information tended to be more available and it flowed more easily to all levels of the organization.

Sproull, R. E. (1991). A lesson in electronic mail. In R. E. Sproull and S. Kiesler (Eds.), *Connections: New ways of working in the networked organization* (pp. 177-184). Cambridge, MA: The MIT Press.

This paper describes the technology for electronic mail including system and social features.

Stallings, W., & Van Slyke, R. (1994). *Business data communications* (Second Edition). New York: Macmillan.

A presentation of data communications and telecommunications from a business perspective. Includes voice, data, image, and video communications and applications.

Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., & Tatar, D. (1987). WYSIWIS revisited: Early experiences with multi-user interfaces. *ACM Transactions of Office Information Systems, 5*(2), 32-47.

Users input data and choices to interact with programs to retrieve information and solve problems The authors discuss multi-user interfaces for access to various workgroup computing information systems. They also discuss synchronous and asynchronous interchanges between multiple users.

Stefik, M., Foster, G., Bobrow, D. G.,Kahn, K., Lanning, S., & Suchman, L. (1988). Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. In I. Greif (Ed.), *Computer-supported cooperative work: A book of readings* (pp. 335-366). San Mateo, CA: Morgan Kaufmann.

Meetings provide coordination or mediate intellectual decisions between groups of people. This article discussed the advantages of using computers to support functions that used to be supported by chalkboards. An experimental meeting room known as Colab was setup at Xerox PARC. Collaborative processes were studied using face-to-face meetings. The project resulted in a usable meeting room and several tools to support collaboration.

Stevenson, T. (1993, June 15). Groupware: Are we ready? *PC Magazine*, pp. 267-299. (rr)

This report reviews leaders of the PC groupware pack. A key point is that groupware is not well defined, so selection of the competing programs was difficult. Products that were reviewed include Beyond Mail for DOS, Beyond Mail for Windows, CM/I, Keyfile, Lotus Notes, Office IQ, and Futurus Team. Selection of a final product depends on the type of workgroup situation and knowledge of what problems must be solved. Products address different areas with some overlap; however, no single product addresses all aspects of workgroup computing.

Takemura, H., & Kishino, F. (1992). Cooperative work environment using virtual workspace. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 226-232). Toronto, Canada: ACM Press. (rr)

Takemura and Kishino developed a cooperative working environment using a virtual environment. An image of a conference room was generated using real time computer graphics, stereoscopic head position trackers, and hand gesture input devices. Users of the system appear to be sharing the same space and they can simultaneously grasp, move, or release objects in the virtual space. Goal was to apply the interface to a telconferencing system. Potential bottlenecks and their solutions are discussed. The system has potential for implementation in many areas including teleoperations, telecommunications, and real time simulations.

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

Tang studied the work activity of small groups using a shared drawing space. Video-based interaction analysis techniques were used to study the collaborative drawing activity. Design assumptions for individual users were re-examined to help build tools for collaborative drawing processes. The experiments showed that participants required more than the shared work environment for effective collaboration. Hand gestures and the use of the drawing space to help mediate interaction were two of the important elements to emerge from this study.

Tatar, D. G., Foster, G., & Bobrow, D. G. (1991). Design for conversation: Lessons from Cognoter. *International Journal of Man-Machine Studies*, *34*(2), 185-209. (rr)

Cognoter is a multi-user idea organizing tool. This project was an attempt to provide computational support for small design teams working in the same area. The project included networked computers, video network facilities, and a specially designed room. Initial results with the system showed serious breakdowns in the system. Breakdowns were traced to differences in the cognitive communication models and the Cognoter model. The authors analyzed traditional sociological communication models. Results were used to successfully redesign the Cognoter system.

Taylor, D. A. (1990). *Object-oriented technology: A manager's guide.* Reading, MA: Addison-Wesley.

A basic guide to object-oriented technology specifically written for managers.

Tesler, L. (1991, September). Networked computing in the 1990s. *Scientific American*, pp. 86-93. (rr)

Tesler contends that in the future computers will play more active role in collaborating with the user. He explains the four different eras of computing and discusses how it will be more natural to collaborate through a computer in the next "five or six years" (p. 91). Tesler's article lends credence to the collaborative computing theories of other authors; however, Tesler optimistically views the collaborative computer as something that will happen.

Turoff, M. (1991). Computer-mediated communication requirements for group support (excerpts). *Journal of Organizational Computing, 1*, 85-113.

The author presents a historical perspective of computer-mediated communication and its relationship to designing for group support. Design feature examples that support specific tasks are included. He also discusses the advantages of asynchronous communication process support.

Valacich, J. R., Dennis, A. R., & Nunamaker, J. F. (1991). Electronic meeting support: The groupsystems concept. In S. Greenberg (Ed.), *Computer-supported cooperative work and groupware* (Chap. 2, pp. 11-28). London: Harcourt Brace Jovanovich. (rr)

There is growing interest in using information technology to support face-to-face group meetings. This paper discusses the evaluation research conducted by the University of Arizona that has led to the installation of over 30 Electronic Meeting System (EMS) environments at sites around the world The researchers are convinced that EMS technology has the potential to change the way people work by supporting larger groups, reducing meeting time, and enhancing group member satisfaction.

Viller, S. (1991). The group facilitator: A cscw perspective. In L. Bannon, L. Robinson & K. Schmidt (Eds.), *Proceedings of the Second European Conference on Computer Supported Cooperative Work* (pp. 81-95). Amsterdam, The Netherlands: Kluwer Academic. (rr)

Facilitating computer conferencing systems helps reduce problems due to lack of face-to-face communications. The role of facilitator has been absent in CSCW systems. The author explains the benefits of a group facilitator and addresses the issues inherent in the groupwork process. The facilitator role must be considered when designing these systems.

Vin, H. M., & Chen, M. (1992). System support for computer mediated multimedia collaborations. In J. Turner & R. Kraut (Eds.), *CSCW '92: Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 203-209). Toronto, Canada: ACM Press.

Vin and Chen present a model for multimedia collaborations. The model consists of a hierarchal abstraction of data streams, sessions, and conferences. The model supports asynchronous and synchronous workgroups and provides sophisticated access control and building blocks for multimedia applications and collaborations.

Vin, H. M., Rangan, P. V., & Ramanathan, S. (1991). Hierarchical conferencing architectures for inter-group multimedia collaboration. In P. de Jong (Ed.), *Conference on Organizational Computing Systems* (pp. 43-54). Atlanta, GA: ACM Press. (rr)

Communication and computer advances have stimulated the integration of digital video and audio with computing. This has lead to a number of computer-assisted collaborations. This paper proposes a multi-level conferencing system for supporting collaborative interactions. The performance of this system is studied and limits are derived on the number of participants in a group and the number of groups in a super conference. Limits are based on the bandwidth and delay requirements that can be tolerated by multimedia.

Von Worley, W. (1994). *An analysis of workgroup computing applications and issues.* Unpublished practicum. Nova Southeastern University, School of Computer and Information Sciences, Ft. Lauderdale, FL.

This study identifies workgroup computing applications and analyzes workgroup computing problems and issues. It discusses the technological basis for workgroup computing including networking and operating systems. Outputs include a list of workgroup computing applications, an analysis of workgroup computing issues, and a 120 article workgroup computing annotated bibliography.

Wacker, S. (1992). Groupware in a law office: A user's experiences. In David D. Coleman (Ed.), *Groupware 92* (pp. 425-427). San Mateo, CA: Morgan Kaufmann. (rr)

The author progresses through the perils and pitfalls of implementing a set of groupware tools in a law office setting. After several years the rudimentary tools are in place; however, E-mail is still the most popular function being used. Wacker thinks "the biggest challenge is to keep our minds open to the possibilities of groupware." (p. 427).

Wagner, G. R., Wynne, B. E., & Menecke, B. E. (1993). Group support system facilities and software. In L.M. Jessup and J. S. Valacich (Eds.), *Group support systems: New perspectives* (pp. 8-57). New York: Macmillan.

Provides background on various group support theories. Provides capsule descriptions for nine group support systems research sites.

Wastell, D. G., & White, P. (1993). Using process technology to support cooperative work: Prospects and design issues. In D. Diaper and C. Sanger (Eds.), *CSCW in practice: An introduction and case studies* (Chap. 8, pp. 105-126). New York: Springer-Verlag. (rr)

This chapter describes a form of technology known as Process Support Technology. This process provides a set of techniques to allow modelling cooperative work. Case studies are provided that illustrate this technique. Issues arising from field studies are also discussed.

Weiss, J., & Schremp, D. (1993, August). Putting data on a diet. *IEEE Spectrum*, pp. 36-40.

Discusses a variety of data compression techniques. Includes lossless and lossy compression.

Whittaker, S., Brennan, S. E., & Clark, H. E. (1991). Co-ordinating activity: An analysis of interaction in computer-supported co-operative work. In S. P. Robertson, G. M. Olson and J. S. Olson (Eds.), *CHI '91 Conference Proceedings* (pp. 361-367). New Orleans, LA: ACM Press. (rr)

The authors explored shared media interaction with the use of an electronic whiteboard with and without the use of a speech channel. The users were separated geographically. Results showed that the users could construct shared data structures to organize activity. Parallel activity was possible, and the speech channel was used for coordination of activity.

Whittier, R. J. (1992). Supplying the right stuff for groupware. In David D. Coleman (Ed.), *Groupware 92* (pp. 34-42). San Mateo, CA: Morgan Kaufmann. (rr)

As organizations continue to network, business decisions times are being reduced. Whittier who works for Intel calls this "Just-in-Time Business" (p. 34). Data must move quickly to the right people as well as give them access to it. Computer supported collaboration enables Just-in-Time Business through inter-personal communications, information sharing, inter-office automation, and real-time conferencing. Intel has focused its CSCW enhancement efforts in several areas including processor power, connectivity, mobility, and natural data.

Winograd, T. (1988, December). Where the action is. *Byte*, pp. 256A-258. (rr)

Groupware is viewed in a different way in this article. Winograd looked at the concept of work as being ". . . a network of interlinked actions which are embodied in language" (p. 256A). Winograd's work is based on a theory of language developed by Flores and Winograd. The basic principle theorizes that when you write or speak you are performing "speech acts" (p. 256B) that have consequences for your future actions and actions of other people you are addressing. These speech acts are a series of requests and promises that are the building blocks for larger systems founded on the same principles.

For people to work effectively together they must be able to "affect and anticipate" (p. 256A) the actions of others in the work group through the use of language. Winograd feels that computers are action machines that offer a promising approach for improving group interaction.

Witherspoon, J. P. (1992). Electronic collaborations in a multi-campus network. In David D. Coleman (Ed.), *Groupware 92* (pp. 465-469). San Mateo, CA: Morgan Kaufmann. (rr)

The author describes a system that uses the Internet to support education and research collaborations between nine different schools. The Bestnet system is based on existing VAX technologies such as electronic mail, conferencing and videotext, and riding the Internet.

Wreden, N. (1993, March/April). Regrouping for groupware. *Beyond Computing*, pp. 52-55. (rr)

Wreden addresses potential benefits and problems implementing groupware in organizations. Each groupware implementation has its own special problems depending on the type of installed equipment base and the computer maturity of the organization. Wreden states that "Successful groupware implementation depends on a shared collaborative vision that is actively supported--both in word and deed--by everyone in the organization" (p. 55).

Yager, T. (1993, March). Better than being there. *Byte*, pp. 129-134. (rr)

Yager discusses desktop video teleconferencing that requires high-speed networks and video compression technology. Corporations and the government are making strides toward implementation of this revolutionary technology. Operating systems that support this technology are also required.

Yavatkar, R., & Lakshman, K. (1994). Communication support for distributed collaborative applications. *Multimedia Systems*, *2*(2), 74-88.

Distributed multimedia systems require resolution of communication issues such as concurrency control and temporal and causal synchronization. The authors propose a protocol suite called multiflow conversation protocol (MCP) that addresses these issues.

Appendix A

Workgroup Computing Taxonomy without Annotations

The complete workgroup computing taxonomy follows without annotations. Appendix B includes annotations that explain individual tasks and primitives.

The Taxonomy

Computer-aided design, drafting, and rendering.

Glickman and Kumar (1993), Goetsch (1986), Kyng (1991), and Shu and Flowers (1992) provided source material for the computer-aided design, drafting, and rendering section of the workgroup computing taxonomy.

Design session setup

Start workgroup session

Unrestricted

Restricted

Define access list

Enter "User IDs"

Change access

Add "User ID"

Delete "User ID"

Send invitation

Request user information

Logon to workgroup session

Enter "User ID"

Enter "Password"

Logoff workgroup session

Request to join group

Approval to join group

Join

- Do not join
- Change session leader
- Editing Mode
  - Designated
  - Baton mode
  - First-come, first-served
  - Free
- Gesture on
  - Point
  - Write
  - Erase
  - Direct attention
  - List workgroup notes
- Gesture off
- Status information
  - Who is on-line
  - Topic
  - Group leader
  - Who is in control
- Audio/Video on
- Audio/Video off
- Voice mute
- Video mute

File

- New
- Open
- Layout
  - Single window
    - Specify drawing size
  - Multiple windows
    - Specify number
    - Specify drawing size
  - Work in
    - 2D
    - 3D
    - Layers
      - Add layer
      - Delete layer
- Recall Session
- Merge Session
- Split session
- Save as

Save current snapshot

Retrieve a snapshot

Save session

Archive session

Compression on

Compression off

Import file or object

Export file or object

Import private view into workspace

Drawing setup

Draw

All session design objects have the following:

Send description

Coupling status

Owner ID

Acquisition status

Request change status

Do change status

Request change owner

Do change owner

Request to grab

Do grab

Where grabbed

End grab

Owned objects list

Text

Type text

Shared workspace text

Request select text

Do select text

Change text

Arrange

Clear workgroup surface

Display

Access private view

Move to empty workgroup view

Move to common workgroup view

Show virtual positionpace.

Drawing, graphical design, and presentations.

Bly & Minneman (1990), Greenberg et al. (1992), Lakin (1988), Rimmer (1991), Software Publishing Corporation (1991), and Tang (1991) provided source material for the drawing, graphical design, and presentations section of the workgroup computing taxonomy.

Drawing session setup

Start workgroup session

- Unrestricted
- Restricted
    - Define access list
        - Enter "User IDs"
- Change access
    - Add "User ID"
    - Delete "User ID"
- Send invitation

Logon to workgroup session

- Enter "User ID"
- Enter "Password"

Logoff workgroup session

Request to join group

Approval to join group

- Join
- Do not join

Change session leader

Editing Mode

- Designated
- Baton mode
- First-come, first-served
- Free

Gesture on

- Point
- Write
- Erase
- Direct attention
- List workgroup notes

Gesture off

Status information

Who is on-line

Topic

Group leader

Who is in control

Audio/Video on

Audio/Video off

File

New

Open

Layout

Single window

Specify drawing sheet size

Multiple windows

Specify number of windows

Specify drawing sheet size

Work in

2D

3D

Layers

Add layer

Delete layer

Recall Session.

Merge Session

Split session

Save as

Save current snapshot

Retrieve a snapshot

Save session

Archive session

Compression on

Compression off

Import file or object

Export file or object

Import private view into workspace

Draw

All session drawing objects have the following:

Send description

Coupling status

Owner ID

Acquisition status

Request change status

Do change status

Request change owner

Do change owner

Request to grab

Do grab

Where grabbed

End grab

Owned objects list

Text

Type text

Shared workspace text

Request select text

Do select text

Change text

Arrange

Clear workgroup surface

Display

Access private view

Move to empty workgroup view

Move to common workgroup view

Show virtual position

Decision support.

Benest and Dukic (1993), Borenstein and Thyberg (1991), the Committee on Human Factors, National Research Council, Committee on Human Factors, Commission on Behavioral and Social Sciences and Education (1990), Gery (1991), Huber (1990), Khoshafian, et al. (1992), Turoff (1989), and Vin and Chen (1992) provided source material for the decision support section of the workgroup computing taxonomy.

Decision support workgroup setup

- Start workgroup session
  - Unrestricted
  - Restricted
    - Define access list
      - Enter "User IDs"
  - Change access
    - Add "User ID"
    - Delete "User ID"
  - Assign roles
    - Organizer
    - Contributor
  - Send invitation
  - Request user information
- Logon to workgroup session
  - Enter "User ID"
  - Enter "Password"
- Logoff workgroup session
- Request to join group
- Approval to join group
  - Can join
  - Cannot join
- Change session leader
- Communication mode
  - Whiteboard session
  - Send E-mail
  - Send to bulletin board
- Editing Mode
  - Designated
  - Baton mode

- First-come, first-served
- Free

Gesture on

- Point
- Write
- Erase
- Direct attention
- List workgroup notes

Gesture off

Audio/Video on

Audio/Video off

Voice mute

Video mute

Status information

- Who is on-line
- Topic
- Group leader
- Who is in control

File

New

Open

Layout

- Single window
- Multiple windows
  - Specify number of windows

Recall Session

Merge Session

Split session

Save current snapshot

Retrieve a snapshot

Save session

Archive session

Import file or object

Export file or object

Import private view into workspace

Select decision style

Use available information

Obtain information from subordinates

- Explain the problem
- Do not explain the problem

Share problem with individuals,

obtain ideas and suggestions

Share problem with the group,

obtain collective ideas

Share problem with all, generate

and estimate alternatives, reach

consensus

Select decision model

None

Analysis

Brainstorming

Model

Simulation

Decision making tools

Freeze contributions

Poll the group

Obtain group consensus

Obtain group vote

Task assignment

Task

Responsibility

Deadline

Send task

Intelligent decision functions

Access decision making database

Filters

Sorting

Develop declarative decision rules

Electronic mail.

Borenstein and Thyberg (1991), Borland, et al. (1993), Khoshafian, et al. (1992), Sproull (1991), and Turoff (1989) provided source material for the electronic mail (E-mail) section of the workgroup computing taxonomy.

Start E-mail session

Logon

Enter "User IDs"

Enter password

Logoff

Send mail

Message type

Send E-mail

Send to bulletin board

Bulletin board

Voice mail

Multimedia message

Priority

High

Medium

Low

Compose message

Formal

External memo

Departmental memo

Memo to manager

Voice message

Reply

Reply to all

Reply to meeting request

Custom templates

Edit message text

Edit attachments

Addresses

Select addressees from group list

Specify new addressee

Return receipt

Response requested

Voting

Deferred delivery

Attach multimedia object

- Object in sender's file
- Object in common area
- Object in library
- Object in database

Group address book

- Add name
- Edit name
- Remove name

Encryption on

Encryption off

Compression on

Compression off

Send message

Read messages

Scan unread mail

Read

- Entire message
- Summary

Autoreply

Permission to access mailbox

Forward messages

Subject

Message name

Addresses

- Select addressees from group list
- Specify new addressee

Attachments

- Forward
- Do not forward

Signature authority

Send user ID

- Public key
- Encryption key
- Request certificate
- Request new name and public key

Message sorting

Append a message

- Sort by
  - Date
  - Sender
  - Subject
  - Priority
  - Keywords
  - Declarative rules
  - Attributes
    - Interests
    - Expertise
    - Background
    - "Reports-to" relationship
    - "Works-with" relationship
- Share a folder
  - By time
    - User ID
  - By person

Interactive communication.

Benest and Dukic (1993), Dourish and Bellotte (1992), Ohkubo and Ishii (1990), Stefik et al. (1987), Vin and Chen (1992), and Rodden (1993) provided source material for the interactive communication section of the workgroup computing taxonomy.

Communication session setup

Start communication session

- Unrestricted
- Restricted
  - Define access list
    - Enter "User IDs"
- Change access
  - Add "User ID"
  - Delete "User ID"
- Assign roles
  - Organizer
  - Primary user
  - Contributor
- Send invitation

Logon to workgroup session

- Enter "User ID"
- Enter "Password"

Logoff workgroup session

Request to join group

Approval to join group

- Can join
- Cannot join

Change leader

Communication mode

- Text only (synchronous)
- Multimedia session (synchronous)
- Message (synchronous or asynchronous)

Editing Mode

- Designated
- Baton mode
- First-come, first-served
- Free

Gesture on

- Point
- Write
- Erase
- Direct attention
- List workgroup notes

Gesture off

Status information

- Who is on-line
- Topic
- Group leader
- Who is in control

Audio/Video on

Audio/Video off

Voice mute

Video mute

Merge Session

Split session

Save current snapshot

Retrieve a snapshot

Save session

Archive session

Encryption on

Encryption off

Import private view into workspace

File transfer

Send file or object

Receive file or object

Import file or object

Export file or object

Link to E-mail

Drawing/sketching

All session drawing objects have the following:

- Send description
- Coupling status
- Owner ID
- Acquisition status
- Request change status
- Do change status
- Request change owner
- Do change owner

Request to grab

Do grab

Where grabbed

End grab

Owned objects list

Text

Type text

Shared workspace text

- Request select text
- Do select text
- Change text

On-line decision making

Poll the group

Obtain group consensus

Obtain group vote

Task assignment

- Task
- Responsibility
- Deadline
- Send task

Access database

Meeting and conference support.

Crowley et al. (1990), Dubs and Hayne (1992), Jay (1976), Nunamaker et al. (1993), Nunamaker et al. (1991), Sarin and Greif (1988), and Stefik et al. (1988) provided source material for the meeting and conference support section of the workgroup computing taxonomy.

Meeting/conference setup

Plan meeting/conference session

- Open
- Closed
  - Define access list
    - Enter "User IDs"
- Change access
  - Add "User ID"
  - Delete "User ID"
- Date
  - Specify date
- Time
  - Specify time
  - Autoselect
- Topic
- Assign roles
  - Facilitator
  - Organizer
  - Chairperson
  - Attendee
  - Contributor
  - Observer

Send invitation

Agenda

- Develop
- Send

Respond to invitation

- Reply
- Will attend
- Cannot attend
- Don't know right now
- Comments

- Request user/schedule information
- Connect meeting/conference
- Logon to meeting/conference session
  - Enter "User ID"
  - Enter "Password"
- Logoff meeting/conference session
- Request to join meeting/conference
- Approval to join meeting/conference
  - Can join
  - Cannot join
- Change leader
- Communication mode
  - Text only (synchronous)
  - Multimedia session (synchronous)
  - Multiple windows
    - Number
- Floor control
  - Implicit request, implicit grant
  - Explicit request, implicit grant
  - Explicit request, explicit grant
  - No floor
- Editing Mode
  - Designated
  - Baton mode
  - First-come, first-served
  - Free
- Gesture on
  - Point
  - Write
  - Erase
  - Direct attention
  - List workgroup notes
- Gesture off
- Status information
  - Who is attending

Meeting/conference
  - Topic
  - Chairperson

Facilitator

Floor holder

Audio/Video on

Audio/Video off

Voice mute

Video mute

Phone support

Access phone list

Dial number

Hangup

Redial

Hold

Mute

Merge Session

Split session

Define private space

Activate private space window

Deactivate private space window

Import private view into workspace

Ideas

Generation

Brainstorming

Topic commentor

Group outliner

Organization

Idea organizer

Issue analyzer

Group writer

Prioritizing

Vote selection

Yes/no

Multiple choice

10-point scale

Rank order

Alternative evaluation

Rate on a 1-10 point scale

Criterion

Weights

Questioning

Individual questionnaire

Group questionnaire

Group matrix

Add rating

Change rating

Policy development

Draft policy

Combine policy

Editing

Document editor

Video editor

Post-meeting documents

Electronic meeting/conference transcript

Complete transcript

Snapshot

Summary

Highlights

File

Save current snapshot

Retrieve a snapshot

Save session

Archive session

Playback a session

Send file or object

Receive file or object

Import file or object

Export file or object

Link to E-mail

Sketching

Activate

Deactivate

All meeting/conference session drawing objects have the following:

Send description

Coupling status

Owner ID

Acquisition status

Request change status

Do change status

Request change owner

Do change owner

Request to grab

Do grab

Where grabbed

End grab

Owned objects list

Text

Type text

Shared workspace text

  Request select text

  Do select text

  Change text

Decision making

Access decision analysis software

Make proposal

  Public

  Private

  Import proposal

Poll the group

Obtain group consensus

Obtain group vote

Task assignment

  Task

  Responsibility

  Deadline

  Send task

Access decision making database

  Database browser

Project management.

Kerzner (1992), Digital Tools (1993), and Sathi et al. (1988) provided source material for the project management section of the workgroup computing taxonomy.

Project management workgroup setup

- Define project
  - Project name
  - Project title
  - Project start date
  - Project manager's name
  - E-mail address
  - Access code (read only)
  - Access code (read/write)
  - Access code (write only)
- Define project workgroup
- Define access list
  - Enter "User IDs"
- Change access
  - Add "User ID"
  - Delete "User ID"
- Assign roles and access to project files
  - Project manager
  - Subproject manager
  - Workgroup member
  - Top management
  - Other users
- Assign access to subproject files
  - Project manager
  - Subproject manager
  - Workgroup member
  - Top management
  - Other users
- File locking
- Logon to project management files
  - Enter "User ID"
  - Enter "Password"
- Logoff project management files
- New

Open

Request define project

Import

Export

Subproject

Explode

Previous

Top level

Project management

meeting/conference setup

Plan meeting/conference session

Open

Closed

Define access list

Enter "User IDs"

Change access

Add "User ID"

Delete "User ID"

Date

Time

Specify time

Autoselect

Topic

Location

Assign roles

Facilitator

Organizer

Chairperson

Attendee

Contributor

Observer

Send invitation

Agenda

Develop

Send

Respond to invitation

Reply

Will attend

Cannot attend

Don't know right now

Comments

Request schedule information

Connect meeting/conference

Logon to meeting/conference

    session

    Enter "User ID"

    Enter "Password"

Logoff meeting/conference session

Request to join meeting/conference

Approval to join

    meeting/conference

    Can join

    Cannot join

Change leader

Communication mode

    Text only (synchronous)

    Multimedia session

    (synchronous)

    Multiple windows

        Number

Floor control

    Implicit request, implicit grant

    Explicit request, implicit grant

    Explicit request, explicit grant

    No floor

Gesture on

    Point

    Write

    Erase

    Direct attention

    List workgroup notes

Gesture off

Audio/Video on

Audio/Video off

Voice mute

Video mute

Status information

    Who is on-line

    Topic

    Group leader

    Who is in control

Save session

Archive session

Import file or object

Export file or object

Import private view into workspace

Project management functions

Send E-mail

Send project files to

Send committee plan to workgroup members

Request inputs from subproject managers

Request for reports

Plan schedule

Plan resources

Generate ideas

Organize ideas

Transfer file

Scheduling and calendaring.

Campbell (1992), Greenwood (1992), Lange (1992), and On technology (1993) provided source material for the scheduling and calendaring section of the workgroup computing taxonomy.

Schedule and calendar workgroup setup

Define schedule workgroup

Define access list

Enter "User IDs"

Change access

Add "User ID"

Delete "User ID"

Assign roles

Manager

Workgroup member

Top management

Other users

File locking

Logon to schedule system

Enter "User ID"

Enter "Password"

Logoff schedule system

New

Open

Import

Export

Calendar

Organize appointments

Reminder notices

"To do" items

Auto-date programming

Alarm

Show appointments and free time

Show free time only

Block view

Scheduling an activity

Title of activity

Date

Time

Duration

Frequency

Private

Flexible

Select open time

Change activity to a meeting

Invite quests

Send invitation

Request RSVP

Automatic acceptance

No schedule conflicts

Scheduler

Define users

Define resources

Conference rooms

Equipment

Wait list for resources

Define resource owner

Search for available meeting times

List available resources

Notify participants

Accept or reject schedule

inputs

Track acceptance or rejection

Creating a "to do" list

New

Priority

Reminder

Notes

Send "to do" request to other

workgroup members

Select participants

Done

Responding to "to do" requests

Open "to do" request

Check on appropriate "to do" item

Read notes

Comments

Options

Will do

Won't do

- I'll decide later
- Done

Reply

Proxies

Proxy list

- Select names

Provide access

- Read only
- Read/write
- Accept proposals

View proxy calendar

Send message to principal

Planning a meeting/conference session

Open

Closed

- Define access list
  - Enter "User IDs"

Define guest type

- Required
- Optional
- Carbon copy
- Blind carbon copy

Change access

- Add "User ID"
- Remove "User ID"

Date

- Specify date

Time

- Specify time
- Autoselect

Topic

Location

Assign roles

- Facilitator
- Organizer
- Chairperson
- Attendee
- Contributor
- Observer

Notes

Send invitation

Agenda

- Develop
- Send

Respond to invitation

- Received, not yet replied
- Reply
- Will attend
- Cannot attend
- Will decide later
- Comments
- Define proxy

In/Out board

View schedules

Shared databases.

Celentano et al. (1991), Greif (1992), Greif and Sarin (1988), Holtham (1993), Khoshafian et al. (1992), and Schwartz (1992) provided source material for the shared databases section of the workgroup computing taxonomy.

Shared database workgroup setup

Define shared database workgroup

- Database name
- User name

Define access rights

- User names
- Owner
- Transfer ownership
- Share ownership
- Delete permission

Access restrictions

- "User ID"
- Attribute
    - Filed
    - Category
    - Class
- Read-only
- Read and write
- Append

Logon to database system

- Enter "User ID"
- Enter "Password"

Logoff database system

File locking

- Lock database
- Request lock release
- Negotiate lock
- Reserve

Data compression

- On
    - Text
    - Video
    - Graphics
    - Audio

Combination

All

Off

Transactions

Begin transaction

Commit transaction

Roll back transaction

Manipulation

Filing

Public area

Private area

Filing rules

Sender

Receiver

Reference to

Copy of

View of

Archive information

Retrieval

Define query

Retrieve based on "as a result of"

Retrieve based on "by"

Retrieve based on "submitted by"

Retrieve based on "came from"

Retrieve based on "signed by"

Retrieve based on unique query

Search

Full text search

Semantic search

Workflow management.

Bair (1993), Bock (1992), Khoshafian (1992), Mashak (1992), Palermo and McCready (1992), and Schael and Zeller (1993) provided source material for the workflow management section of the workgroup computing taxonomy.

Workflow design

Define workflow cycle

Task definition

Workflow rules

Static

Dynamic

Define rules

Use electronic forms

Define criteria

Procedures that state

What must precede these

Activity for it to execute

Define action items

Modify workflow

Copy workflow

Routing or branching

Define group

Assign roles

Completor

Information provider

Reviewer

Coordinator

Approver

Report generator

Assign action to an individual

Assign actions to a group

Group task assignment

Load balancing

Next available

Designate role member

Designate specific member

Describe activities

Refuse role or action

Claim role

- Create document
- Integrate with personal "to do" list

Define deadlines

Flow control

- Serial
- Parallel

Access control

Define access

- By group
- By user
- Permission by workflow

Report permission

- By workflow
- By report

Modification rights

- By group
- By user

Permissions

Delegate

Reject

Correct mistake by objection

Withdraw

View attached documents

Add to attached documents

Modify

Complete

Logon to workflow system

- Enter "User ID"
- Enter "Password"

Logoff workflow system

Approvals

Digital Signature

Encryption

Encryption on

Encryption off

Status monitoring

View steps

- Last step
- Current step
- Next step
- View deadlines

By date

By project

By type of action

Define views

Overdue alert

Pending actions

Completed

Filing

Archive to database

Access database

Utilities

Send E-mail message

Writing and editing.

Baecker et al. (1993), Baydere et al. (1993), Denley et al. (1993), Knister and Prakash (1990), Neuwirth et al. (1990), Posner and Baecker (1992), Sharples (1993), Sharples et al. (1993), Shen and Dewan (1992), and Sobiesiak and Myopoulos (1991) provided source material for the writing and editing section of the workgroup computing taxonomy.

Writing and editing workgroup setup

Define writing and editing workgroup

Document name

User name

Session control

Roles

Lead writer

Co-writer

Writer

Consultant

Editor

Reviewer

Activities

Brainstorm

Research

Initial plant

Write

Control changes

Edit document

Final edit

Review

Writing strategies

Single writer

Scribe

Separate writers

Joint writing

Access control

Document control

Centralized

Relay

Independent

Shared

Permissions

User ID

Read/write

Comment

Read only

Change access

Logon to writing system

Enter "User ID"

Enter "Password"

Logoff writing system

Request to join writing group

Approval to join conference

Can join

Cannot join

Sessions

Create session

Merge session

Leave session

Document segmentation

Separate document sections

Join document sections

Writing rules

Style rules

Content rules

Editing rules

Lead writer select rules

Editor select rules

Both select rules

Revision control

Small change

Many additions

Delete

Change history

Version control

Versions allowed

Single

Multiple

Parallel

Sequential

Reciprocal

Automatic conflict extraction

Present to lead author

Present to all authors

Drawing/sketching

All session drawing objects have the following:

Send description

Coupling status

Owner ID

Acquisition status

Request change status

Do change status

Request change owner

Do change owner

Request to grab

Do grab

End grab

Owned objects list

Status monitoring

Revision status

Tools

Import text or objects

Export text or objects

Information filtering

Send E-mail message

Audio/video

Audio/video on

Audio/video off

Video channel mode

Multiple image mode

Single image mode

Preview

Audio select

All channels

Selected channel

Select

Selected video channel audio only

Audio mute

Video mute

Open multimedia conference

Appendix B

Workgroup Computing Taxonomy with Annotations

The Taxonomy

Computer-aided design, drafting, and rendering.

Glickman and Kumar (1993), Goetsch (1986), Kyng (1991), and Shu and Flowers (1992) provided source material for the computer-aided design, drafting, and rendering section of the workgroup computing taxonomy.

Design session setup

Start workgroup session: Workgroup session design leader initiates a unique workgroup session and establishes registration criteria. Selected participants register when they logon.

Unrestricted: Any authorized user can access the shared design workspace.

Restricted: Only designated users can access the shared design workspace.

Define access list: Workgroup session leader defines an access list.

Enter "User IDs": Workgroup session leader enters a list of user IDs or E-mail addresses.

Change access: Workgroup session design leader can change the access list.

Add "User ID": Self-explanatory.

Delete "User ID": Self-explanatory.

Send invitation: Send specified individuals an invitation to join the group design session.

Request user information: Request pre-specified information on workgroup members. Allow workgroup leader to identify appropriate group members.

Logon to workgroup session: Users can logon to shared workspace. The system passes host name, port number, and user name, to the session registrar.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff workgroup session: Users logoff of shared design workspace. The system sends a logoff notification message to other users that remain on the shared design workspace.

Request to join group: A user can request permission from the session leader to join the group after the session begins.

Approval to join group: Session leader approves or disapproves joining the group.

Join: Self-explanatory.

Do not join: Self-explanatory.

Change session leader: Specify another workgroup session user as the leader.

Editing Mode: Workgroup leader specifies access to shared design objects.

Designated: Only one owner can own a design object or text.

Baton mode: Object owner can pass an owned object or text to another group member.

First-come, first-served: The first requestor can access a design object or text. Another user can request permission to access a non-owned object or text.

Free: More than one user can have simultaneous access to a design object or text. Used for brainstorming.

Gesture on: User can use gestures on the shared design workspace.

Point: User can point at an object or text.

Write: User can write a note.

Erase: User can erase a note.

Direct attention: User can make a motion to direct attention. This function is not available if audio/video is turned on.

List workgroup notes: List all shared workspace notes for all users.

Gesture off: Gesturing is turned off.

Status information:

Who is on-line: Self-explanatory.

Topic: Self-explanatory.

Group leader: Self-explanatory.

Who is in control: Self-explanatory.

Audio/Video on: Video and sound links are turned on. Opens a video window at each user station.

Audio/Video off: Video and sound links are turned off. Session leader can terminate video access. Users can close their video connection.

Voice mute: User turns off the voice portion of the multimedia session.

Video mute: User turns off the video portion of the multimedia session.

File

New: Start a new design session in private workspace. Workgroup session leader can start a new design in shared workspace.

Open: Open an existing design in private workspace.

Layout: Workgroup session leader can setup and change shared design workspace layout. User can change private design workspace layout.

Single window: Open a new single workgroup design window.

Specify drawing size: A, B, C, D, E, F, G, etc.

Multiple windows: Open new multiple workgroup design windows.

Specify number: Self-explanatory.

Specify drawing size: A, B, C, D, E, F, G, etc.

Work in: Specify a drawing as 2D or 3D.

2D: Self-explanatory.

3D: Self-explanatory.

Layers: Add or delete layers to a shared workspace design.

Add layer: Session leader can add a specified design layer.

Delete layer: Session leader can delete a specified design layer.

Recall Session: Workgroup leader opens a previous design session in shared workspace.

Merge Session: Workgroup leader can merge two or more design sessions in shared workspace.

Split session: Workgroup leader can split one session into several sessions.

Save as: Saves a private drawing under a new filename. Workgroup session leader can save a shared workspace design under a new filename.

Save current snapshot: Save a copy of the current workgroup drawing in a user's private file space.

Retrieve a snapshot: Open a copy of a saved snapshot in the user's private workspace.

Save session: Save a copy of all session activities to a video disk file. Includes audio/visual activities, if turned on.

Archive session: Save a compressed file of session activities to a video disk archive file.

Compression on: Saved files compressed.

Compression off: Saved files not compressed.

Import file or object: Import a file or object into the selected design workspace.

Export file or object: Export a file or object from the selected design workspace.

Import private view into workspace: User can import his private workspace design into the shared workspace.

Drawing setup: User can setup his private design workspace.

Draw

All session design objects have the following:

Send description: User can send a design object description to other workgroup members.

Coupling status: Show if a design object is private, public, or has limited access.

Owner ID: Show who owns a design object.

Acquisition status: Show if a design object is being manipulated.

Request change status: Request the design object owner to change coupling status.

Do change status: Actually change coupling status and send change to users.

Request change owner: Request the design object to change its owner.

Do change owner: Change the owner and send change to users.

Request to grab: Request the design object for permission to grab it.

Do grab: Grant or deny permission to grab a design object.

Where grabbed: Check where the design object was grabbed.

End grab: Release the grabbed design object.

Owned objects list: Show a list of design objects owned by the user.

Text

Type text: Type new text into private or shared workspace.

Shared workspace text: Allow editing of shared workspace text.

Request select text: Request permission from text owner to change selected text.

Do select text: Grant or deny permission to change selected text.

Change text: Edit the selected text.

Arrange

Clear workgroup surface: User can clear a private workspace. Workgroup session leader can clear the shared drawing workspace.

Display

Access private view: User can access a private view if it has been opened without logging off the workgroup session.

Move to empty workgroup view: User can move to an empty part of the shared drawing workspace.

Move to common workgroup view: User can move back to the common view of the shared drawing workspace.

Show virtual position: Show the virtual position of designers in the shared workspace.

Grid: Display the workspace grid.

Snap to private grid: Force endpoints of a drawing object to the nearest point of a private workspace grid.

Drawing, graphical design, and presentations.

Bly & Minneman (1990), Greenberg et al. (1992), Lakin (1988), Rimmer (1991), Software Publishing Corporation (1991), and Tang (1991) provided source material for the drawing, graphical design, and presentations section of the workgroup computing taxonomy.

Drawing session setup

Start workgroup session: Workgroup session drawing leader initiates a unique workgroup session and establishes registration criteria. Selected participants register when they logon.

Unrestricted: Any authorized user can access the shared drawing workspace.

Restricted: Only designated users can access the shared drawing workspace.

Define access list: Workgroup session leader defines an access list.

Enter "User IDs": Workgroup session leader enters a list of "User Ids" or E-mail addresses.

Change access: Workgroup session drawing leader can change the access list.

Add "User ID": Self-explanatory.

Delete "User ID": Self-explanatory.

Send invitation: Send specified individuals an invitation to join the group drawing session.

Logon to workgroup session: Users can logon to shared drawing workspace. The system passes host name, port number, and user name, to the session registrar.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff workgroup session: Users logoff of shared drawing workspace. The system sends a logoff notification message to other users that remain on the shared drawing workspace.

Request to join group: A user can request permission from the session leader to join the group after the session begins.

Approval to join group: Session leader approves or disapproves joining the group.

Join: Self-explanatory.

Do not join: Self-explanatory.

Change session leader: Specify another workgroup session user as the leader.

Editing Mode: Workgroup leader specifies access to shared drawing objects.

Designated: Only one owner can own a drawing object or text.

Baton mode: Object owner can pass an owned object or text to another group member.

First-come, first-served: The first requestor can access a drawing object or text. Another user can request permission to access a non-owned drawing object or text.

Free: More than one user can have simultaneous access to a drawing object or text. Used for brainstorming.

Gesture on: User can use gestures on the shared drawing workspace.

Point: User can point at an object or text.

Write: User can write a note.

Erase: User can erase a note.

Direct attention: User can make a motion to direct attention. This function is not available if audio/video is turned on.

List workgroup notes: List all shared workspace notes for all users.

Gesture off: Gesturing is turned off.

Status information: Provide status of session.

Who is on-line: Self-explanatory.

Topic: Self-explanatory.

Group leader: Self-explanatory.

Who is in control: Self-explanatory.

Audio/Video on: Video and sound links are turned on. Opens a video window at each user station.

Audio/Video off: Video and sound links are turned off. Session leader can terminate video access. Users can close their video connection.

File

New: Start a new drawing session in private workspace. Workgroup session leader can start a new drawing in shared workspace.

Open: Open an existing drawing in private workspace.

Layout: Workgroup session leader can setup and change shared drawing workspace layout. User can change private drawing workspace layout.

Single window: Open a new single workgroup drawing window.

Specify drawing sheet size: 1, 2, 3, 4, 5, 6, 7, etc.

Multiple windows: Open new multiple workgroup drawing windows.

Specify number of windows: Self-explanatory.

Specify drawing sheet size: 1, 2, 3, 4, 5, 6, 7, etc.

Work in: Specify a drawing as 2D or 3D.

2D: Self-explanatory.

3D: Self-explanatory.

Layers: Add or delete layers in a shared drawing workspace .

Add layer: Session leader can add a specified drawing layer.

Delete layer: Session leader can delete a specified drawing layer.

Recall Session: Workgroup session leader opens a previous drawing session in shared workspace.

Merge Session: Workgroup session leader can merge two or more drawing sessions in shared workspace.

Split session: Workgroup session leader can split one session into several sessions.

Save as: Save a private drawing under a new filename. Workgroup session leader can save a shared workspace drawing under a new filename.

Save current snapshot: Save a copy of the current workgroup drawing in a user's private file space.

Retrieve a snapshot: Open a copy of a saved snapshot in the user's private workspace.

Save session: Save a copy of all session activities to a video disk file. Includes audio/visual activities, if turned on.

Archive session: Save a compressed file of session activities to a video disk archive file.

Compression on: Saved files compressed.

Compression off: Saved files not compressed.

Import file or object: Import a file or object into the selected drawing workspace.

Export file or object: Export a file or object from the selected drawing workspace.

Import private view into workspace: User can import his private workspace drawing into the shared workspace.

Draw

All session drawing objects have the following:

Send description: User can send a drawing object description to other workgroup members.

Coupling status: Show if a drawing object is private, public, or has limited access.

Owner ID: Show who owns a drawing object.

Acquisition status: Show if a drawing object is being manipulated.

Request change status: Request the drawing object owner to change coupling status

Do change status: Actually change coupling status and send change to users.

Request change owner: Request the drawing object to change its owner.

Do change owner: Change the owner and send change to users.

Request to grab: Request the drawing object for permission to grab it.

Do grab: Grant or deny permission to grab a drawing object.

Where grabbed: Check where the drawing object was grabbed.

End grab: Release the grabbed drawing object.

Owned objects list: Show a list of drawing objects owned by the user.

Text

Type text: Type new text into private or shared workspace.

Shared workspace text: Allow editing of shared workspace text.

Request select text: Request permission from text owner to change selected text.

Do select text: Grant or deny permission to change selected text.

Change text: Edit the selected text.

Arrange

Clear workgroup surface: User can clear a private workspace. Workgroup session leader can clear the shared drawing workspace.

Display

Access private view: User can access a private view if it has been opened without logging off the workgroup session.

Move to empty workgroup view: User can move to an empty part of the shared drawing workspace.

Move to common workgroup view: User can move back to the common view of the shared drawing workspace.

Show virtual position: Show the virtual position of group members in the shared drawing workspace.

Decision support.

Benest and Dukic (1993), Borenstein and Thyberg (1991), the Committee on Human Factors, National Research Council, Committee on Human Factors, Commission on Behavioral and Social Sciences and Education (1990), Gery (1991), Huber (1990), Khoshafian, et al. (1992), Turoff (1989), and Vin and Chen (1992) provided source material for the decision support section of the workgroup computing taxonomy.

Decision support workgroup setup

Start workgroup session: Workgroup session leader initiates a unique synchronous workgroup session and establishes registration criteria. Selected participants register when they logon.

Unrestricted: Any authorized user can access the workgroup session.

Restricted: Only designated users can access the workgroup session.

Define access list: Workgroup session leader defines an access list.

Enter "User IDs": Workgroup session leader enters a list of "User Ids" or E-mail addresses.

Change access: Workgroup session leader can change the access list.

Add "User ID": Self-explanatory.

Delete "User ID": Self-explanatory.

Assign roles: Session leader assigns group roles.

Organizer: Leader of the decision making session.

Contributor: A user who can add comments, send messages, but cannot see other users' inputs.

Send invitation: Sends specified individuals an invitation to join the group decision making session.

Request user information: Request pre-specified information on workgroup members. Allows workgroup leader to identify appropriate group members and define their roles.

Logon to workgroup session: Users logon to workgroup session. The system passes host name, port number, and user name to the session registrar.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff workgroup session: Users logoff workgroup session. The system sends a logoff notification message to other users that remain on workgroup session.

Request to join group: A user can request permission rom the session leader to join the group after the session begins.

Approval to join group: Session leader approves or disapproves joining the group.

Can join: Self-explanatory.

Cannot join: Self-explanatory.

Change session leader: Specify another workgroup session user as the leader.

Communication mode: Specify a synchronous or asynchronous session.

Whiteboard session: A synchronous session using an electronic whiteboard.

Send E-mail: Self-explanatory.

Send to bulletin board: Self-explanatory.

Editing Mode: Workgroup leader specifies access to shared objects and text on the electronic whiteboard.

Designated: Only one owner can own an object or text.

Baton mode: Object owner can pass an owned object or text to another group member.

First-come, first-served: The first requestor can access an object or text. Another user can request permission to access a non-owned object or text.

Free: More than one user can have simultaneous access to an object or text. Used for brainstorming.

Gesture on: User can use gestures on the shared session whiteboard.

Point: User can point at an object or text.

Write: User can write a note.

Erase: User can erase a note.

Direct attention: User can make a motion to direct attention on the electronic whiteboard. This function is not available if audio/video is turned on.

List workgroup notes: Lists all shared workspace notes for all users.

Gesture off: Gesturing is turned off.

Audio/Video on: Video and sound links are turned on. Opens a video window at each user station.

Audio/Video off: Video and sound links are turned off. Session leader can terminate video access. Users can close their video connection.

Voice mute: User turns off the voice portion of the multimedia session.

Video mute: User turns off the video portion of the multimedia session.

Status information: Provide status of the session.

Who is on-line: Self-explanatory.

Topic: Self-explanatory.

Group leader: Self-explanatory.

Who is in control: Self-explanatory.

File

New: Start a new drawing session in private workspace. Workgroup session leader can start a new drawing in shared workspace.

Open: Open an existing drawing in private workspace.

Layout: Workgroup session leader can setup and change shared whiteboard workspace layout.

Single window: Open a new single workgroup whiteboard window.

Multiple windows: Open new multiple workgroup whiteboard windows.

Specify number of windows: Self-explanatory.

Recall Session: Workgroup session leader opens a previous decision making session.

Merge Session: Workgroup session leader can merge two or more decision making sessions.

Split session: Workgroup session leader can split one session into several sessions.

Save current snapshot: Save a copy of the current whiteboard contents in a user's private file space.

Retrieve a snapshot: Open a copy of a saved snapshot in the user's private workspace.

Save session: Save a copy of whiteboard session activities to a video disk file. Includes audio/visual activities, if turned on.

Archive session: Save a compressed file of session activities to a video disk archive file.

Import file or object: Import a file or object into the whiteboard workspace.

Export file or object: Export a file or object from the whiteboard workspace.

Import private view into workspace: User can import private files into the shared whiteboard workspace.

Select decision style: Allows the session leader to select a decision style that fits his management style.

Use available information: Make a decision using only available on-line information.

Obtain information from subordinates: Information from subordinates is requested for decision making. There are two modes.

Explain the problem: Session leader explains the problem.

Do not explain the problem: Session leader requests information, but does not explain the problem.

Share problem with individuals, obtain ideas and suggestions: The problem is explained to specific individuals and the session leader solicits ideas and suggestions then makes a decision.

Share problem with the group, obtain collective ideas: The problem is shared with the entire group. The session leader gets collective ideas and suggestions then makes a decision.

Share problem with all, generate and estimate alternatives, reach consensus: The problem is shared with the entire group. The group generates alternatives and reaches consensus. Polling and voting can take place.

Select decision model

None: Select no decision model.

Analysis: Use analysis to arrive at a decision.

Brainstorming: Select to define problem symptoms, define the problem, define goals, develop decision criteria, determine alternative solutions, or select the best alternative.

Model: Use selected models to support decision making.

Simulation: Run a simulation model to support decision making.

Decision making tools

Freeze contributions: Freeze contribution at a particular point selected by the session leader.

Poll the group: Poll the group on a specific question.

Obtain group consensus: Allow the group to achieve consensus.

Obtain group vote: Allow the group to vote on an issue or specific question.

Task assignment: Session leader can assign specific tasks to one or more group members

Task: Define the task.

Responsibility: Specify who is responsible for task completion.

Deadline: Show the task deadline.

Send task: Transmit the task to designated workgroup users.

Intelligent decision functions

Access decision making database: Access a shared decision support database management system.

Filters: Sort group or database information by specifying case-based filters.

Sorting: Sort group or database information by specifying keyword searches or alpha-numeric indices.

Develop declarative decision rules: Develop declarative rules to support group decision making or information sorting.

Electronic mail.

Borenstein and Thyberg (1991), Borland, et al. (1993), Khoshafian, et al. (1992), Sproull (1991), and Turoff (1989) provided source material for the electronic mail (E-mail) section of the workgroup computing taxonomy.

Start E-mail session

Logon: User logon to an asynchronous E-mail session.

Enter "User IDs": Enter user ID.

Enter password: Enter user password.

Logoff: User logoff from an E-mail session.

Send mail

Message type:

Send E-mail: Send text message via E-mail.

Send to bulletin board: Send text message to a bulletin board.

Voice mail: Send a voice mail message.

Multimedia message: Send a multimedia message. Video and sound links are turned on. Embeds objects using OLE.

Priority: Set the message priority.

High: Self-explanatory.

Medium: Self-explanatory.

Low: Self-explanatory.

Compose message: Select message type.

Formal: Formal style defined by the user.

External memo: Memo to an external user or group.

Departmental memo: Memo to user's department.

Memo to manager: Memo to a manager.

Voice message: Send a voice message.

Reply: Reply to a previously received message.

Reply to all: Send the reply to all previous message addressees.

Reply to meeting request: Reply to a meeting request.

Custom templates: Prepare custom message template.

Edit message text: Edit a message.

Edit attachments: Edit message attachments.

Addresses: Specify message address.

Select addressees from group list: Self-explanatory.

Specify new addressee: Self-explanatory.

Return receipt: Request a return receipt on transmitted message.

Response requested: Ask for a response.

Voting: Take a vote on an issue.

Deferred delivery: Hold the message until the recipient returns.

Attach multimedia object: Self-explanatory.

Object in sender's file: Self-explanatory.

Object in common area: Self-explanatory.

Object in library: Self-explanatory.

Object in database: Self-explanatory.

Group address book: Build a group address book.

Add name: Self-explanatory.

Edit name: Self-explanatory.

Remove name: Self-explanatory.

Encryption on: Encryption on for outgoing messages.

Encryption off: Encryption off for outgoing messages.

Compression on: Outgoing messages compressed.

Compression off: Outgoing messages not compressed.

Send message: Send the message to designated addressees.

Read messages

Scan unread mail: Self-explanatory.

Read: Self-explanatory.

Entire message: Self-explanatory.

Summary: Self-explanatory.

Autoreply: Automatically send a prepared reply to the message sender.

Permission to access mailbox: Allow other users to access user's mailbox.

Forward messages

Subject: Select the subject for messages that should be forwarded.

Message name: Select a message name that should be forwarded.

Addresses: Select who gets the forwarded message.

Select addressees from group list: Self-explanatory.

Specify new addressee: Self-explanatory.

Attachments: Attachments on the forwarded message.

Forward: Self-explanatory.

Do not forward: Self-explanatory.

Signature authority

Send user ID: Self-explanatory.

Public key: Send public key to the recipient.

Encryption key: Create a new encryption key.

Request certificate: Request a certificate of signature.

Request new name and public key: Request a user to forward a new signature name with public key.

Message sorting

Append a message: Append to an attachment.

Sort by: Self-explanatory.

Date: Self-explanatory.

Sender: Self-explanatory.

Subject: Self-explanatory.

Priority: Self-explanatory.

Keywords: Self-explanatory.

Declarative rules: Define declarative rules for message sorting.

Attributes: Sort by attributes.

Interests: Self-explanatory.

Expertise: Self-explanatory.

Background: Self-explanatory.

"Reports-to" relationship: Self-explanatory.

"Works-with" relationship Self-explanatory.

Share a folder: Share a message folder with another user.

By time: Share over a period of time.

User ID: Self-explanatory.

By person: Share with an individual.

***Interactive communication*.**

Benest and Dukic (1993), Dourish and Bellotte (1992), Ohkubo and Ishii (1990), Stefik et al. (1987), Vin and Chen (1992), and Rodden (1993) provided source material for the interactive communication section of the workgroup computing taxonomy.

Communication session setup

Start communication session: Workgroup session leader initiates a unique synchronous communication session and establishes registration criteria. Selected participants register when they logon.

Unrestricted: Any authorized user can access the communication session.

Restricted: Only designated users can access the communication session.

Define access list: Communication session leader defines an access list.

Enter "User IDs": Communication session leader enters a list of "User IDs" or E-mail addresses.

Change access: Communication session leader can change the access list.

Add "User ID": Self-explanatory.

Delete "User ID": Self-explanatory.

Assign roles: Session leader assigns group roles.

Organizer: Leader of the communication session.

Primary user: A user who can comment, send messages, and can see all other user inputs.

Contributor: A user who can add comments, send messages, but cannot see other users' inputs.

Send invitation: Send specified users an invitation to join the group communication session.

Logon to workgroup session: Users logon to communication session. The system passes host name, port number, and user name, to the session registrar.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff workgroup session: Users logoff communication session. The system sends a logoff notification message to other users that remain on workgroup session.

Request to join group: A user can request permission from the session leader to join the group after the session begins.

Approval to join group: Organizer approves or disapproves joining the group.

Can join: Self-explanatory.

Cannot join: Self-explanatory.

Change leader: Specify another communication session leader.

Communication mode:

Text only (synchronous): Self-explanatory.

Multimedia session: (synchronous): Self-explanatory.

Message (synchronous or asynchronous): Self-explanatory.

Editing Mode: Workgroup leader specifies access to shared objects and text on the electronic whiteboard.

Designated: Only one owner can own an object or text.

Baton mode: Object owner can pass an owned object or text to another group member.

First-come, first-served: The first requestor can access an object or text. Another user can request permission to access a non-owned object or text.

Free: More than one user can have simultaneous access to an object or text. Used for brainstorming.

Gesture on: User can use gestures on the shared session whiteboard.

Point: User can point at an object or text.

Write: User can write a note.

Erase: User can erase a note.

Direct attention: User can make a motion to direct attention on the electronic whiteboard. This function is not available if audio/video is turned on.

List workgroup notes: Lists all shared workspace notes for all users.

Gesture off: Gesturing is turned off.

Status information: Provide status about the session.

Who is on-line: Self-explanatory.

Topic: Self-explanatory.

Group leader: Self-explanatory.

Who is in control: Self-explanatory.

Audio/Video on: Video and sound links are turned on. Opens a video window at each user station.

Audio/Video off: Video and sound links are turned off. Session leader can terminate video access. Users can close their video connection.

Voice mute: User turns off the voice portion of the multimedia session.

Video mute: User turns off the video portion of the multimedia session.

Merge Session: Workgroup session leader can merge two or more communication sessions.

Split session: Workgroup session leader can split one session into several sessions.

Save current snapshot: Save a copy of the current whiteboard contents in a user's private file space.

Retrieve a snapshot: Open a copy of a saved snapshot in the user's private workspace.

Save session: Save a copy of whiteboard communication session activities to a video disk file. Includes audio/visual activities, if turned on.

Archive session: Save a compressed file of session activities to a video disk archive file.

Encryption on: Encryption turned on for workgroup session files.

Encryption off: Encryption turned off for workgroup session files.

Import private view into workspace: User can import private files into the shared whiteboard workspace.

File transfer

Send file or object: Send a file or object into the whiteboard workspace.

Receive file or object: Receive a file or object from the whiteboard workspace.

Import file or object: Import a file or object into the whiteboard workspace.

Export file or object: Export a file or object from the whiteboard workspace.

Link to E-mail: Self-explanatory.

Drawing/sketching

All session drawing objects have the following:

Send description: User can send a drawing object description to other workgroup members.

Coupling status: Show if a drawing object is private, public, or has limited access.

Owner ID: Show who owns a drawing object.

Acquisition status: Show if a drawing object is being manipulated.

Request change status: Request the drawing object owner to change coupling status

Do change status: Actually change coupling status and send change to users.

Request change owner: Request the drawing object to change its owner.

Do change owner: Change the owner and send change to users.

Request to grab: Request the drawing object for permission to grab it.

Do grab: Grant or deny permission to grab a drawing object.

Where grabbed: Check where the drawing object was grabbed.

End grab: Release the grabbed drawing object.

Owned objects list: Show a list of drawing objects owned by the user.

Text

Type text: Type new text into private or shared workspace.

Shared workspace text: Allow editing of shared workspace text.

Request select text: Request permission from text owner to change selected text.

Do select text: Grant or deny permission to change selected text.

Change text: Edit the selected text.

On-line decision making

Poll the group: Poll the group on a specific question.

Obtain group consensus: Allow the group to achieve consensus.

Obtain group vote: Allow the group to vote on an issue or specific question.

Task assignment: Session leader can assign specific tasks to one or more group members

Task: Define the task.

Responsibility: Specify who is responsible for task completion.

Deadline: Show the task deadline.

Send task: Transmit the task to designated workgroup users.

Access decision making database: Access a shared decision support database management system.

Meeting and conference support.

Crowley (1990), Dubs and Hayne (1992), Jay (1976), Nunamaker et al. (1993), Nunamaker et al. (1991), Sarin and Greif (1988), and Stefik et al. (1988) provided source material for the meeting and conference support section of the workgroup computing taxonomy.

Meeting/conference setup

Plan meeting/conference session: Conference or meeting session leader plans a unique synchronous conference or meeting session and establishes registration criteria. Selected participants register when they logon.

Open: Any authorized user can access the meeting/conference session.

Closed: Only designated users can access the meeting/conference session.

Define access list: Session leader defines an access list.

Enter "User IDs": Session leader enters a list of user IDs or E-mail addresses.

Change access: Session leader can change the access list.

Add "User ID": Self-explanatory.

Delete "User ID": Self-explanatory.

Date: Date of meeting/conference.

Specify date: Self-explanatory.

Time: Time of meeting/conference.

Specify time: Self-explanatory.

Autoselect: Selects a meeting/conference date and time by comparing invitees calendars to choose the best time for all participants.

Topic: Topic of meeting/conference.

Assign roles: Session leader assigns group roles.

Facilitator: Moderates meeting/conference activities.

Organizer: Organizes meeting/conference session.

Chairperson: Controls meeting/conference activities

Attendee: A user who can comment, send messages, and can see all other user inputs.

Contributor: A user who can add comments, send messages, but cannot see other users' inputs.

Observer: Observes meeting/conference activities. Cannot comment or vote.

Send invitation: Sends designated individuals an invitation to join the meeting/conference session.

Agenda: Meeting/conference organizer can develop an agenda.

Develop: Draft the agenda.

Send: Send the agenda to invitees.

Respond to invitation:

Reply: Send a message to meeting/conference organizer. Includes comments.

Will attend: Self-explanatory.

Cannot attend: Self-explanatory.

Don't know right now: Self-explanatory.

Comments: Comment on the agenda or some other issue.

Request user/schedule information: Request pre-specified information on workgroup members. Allows workgroup leader to identify appropriate meeting/conference members, see their calendars, and define their roles.

Connect meeting/conference: Session leader connects the meeting/conference. The system passes a message to invitees that the meeting/conference is starting.

Logon to meeting/conference session: Users logon to meeting/conference session. The system passes host name, port number, and user name to the session registrar.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff meeting/conference session: Users logoff meeting/conference session. The system sends a logoff notification message to other users that remain on meeting/conference session.

Request to join meeting/conference: A user can request permission from the session leader to join the meeting/conference after the session begins.

Approval to join meeting/conference: Leader approves or disapproves joining the meeting/conference.

Can join: Self-explanatory.

Cannot join: Self-explanatory.

Change leader: Specify another meeting/conference session leader.

Communication mode:

Text only (synchronous): Open a text only session.

Multimedia session (synchronous): Open a multimedia session.

Multiple windows: Session leader can open multiple sessions.

Number: Designate sessions.

Floor control: Determine how the floor is requested and how it is relinquished.

Implicit request, implicit grant: Inactive user requests floor by typing or moving the mouse. Session monitor notices the request and always grants it.

Explicit request, implicit grant: Inactive user explicitly requests the floor. Request is always granted.

Explicit request, explicit grant: Inactive user explicitly request the floor. Active user explicitly relinquishes it.

No floor: More than one user can have simultaneous access to the floor. Used for brainstorming and arguing.

Editing Mode: Meeting/conference leader specifies access to shared objects and text on the meeting/conference electronic whiteboard.

Designated: Only one owner can own an object or text.

Baton mode: Object owner can pass an owned object or text to another group member.

First-come, first-served: The first requestor can access an object or text. Another user can request permission to access a non-owned object or text.

Free: More than one user can have simultaneous access to an object or text. Used for brainstorming.

Gesture on: User can use gestures on the shared meeting/conference whiteboard.

Point: User can point at an object or text.

Write: User can write a note.

Erase: User can erase a note.

Direct attention: User can make a motion to direct attention on the electronic whiteboard. This function is not available if audio/video is turned on.

List workgroup notes: Lists all shared workspace notes for all users.

Gesture off: Gesturing is turned off.

Status information: Provide information about meeting/conference attendees.

Who is attending meeting/conference: Self-explanatory.

Topic: List the meeting/conference topic.

Chairperson: List the meeting/conference chairperson.

Facilitator: List the meeting/conference facilitator.

Floor holder: Show current floor holder and time remaining.

Audio/Video on: Video and sound links are turned on. Opens a video window at each user station.

Audio/Video off: Video and sound links are turned off. Session leader can terminate video access. Users can close their video connection.

Voice mute: User turns off the voice portion of the multimedia session.

Video mute: User turns off the video portion of the multimedia session.

Phone support: User can make a phone call via computer during the meeting/conference session.

Access phone list: Call up the user's pre-defined phone list.

Dial number: Dial a selected number or manually input a number.

Hangup: Terminate the call.

Redial: Redial the previous number.

Hold: Put caller on hold.

Mute: Mute the call.

Merge Session: Workgroup session leader can merge two or more meeting/conference sessions.

Split session: Workgroup session leader can split one meeting/conference session into several sessions.

Define private space: User can define a private workspace.

Activate private space window: Self-explanatory.

Deactivate private space window: Self-explanatory.

Import private view into workspace: User can import private files into the meeting/conference whiteboard workspace.

Ideas

Generation:

Brainstorming: Self-explanatory.

Topic commentor: Individuals place comments in a card file for sorting and review.

Group outliner: The group assembles comments in a hierarchical structure for group review.

Organization:

Idea organizer: Group member writes and organizes ideas.

Issue analyzer: Group members identify and consolidate issues and ideas for group consensus.

Group writer: Group members write to organize ideas.

Prioritizing:

Vote selection: Allows issues to be prioritized using several schemes.

Yes/no: Self-explanatory.

Multiple choice: Self-explanatory.

10-point scale: Self-explanatory.

Rank order: Self-explanatory.

Alternative evaluation:

Rate on a 1-10 point scale: Self-explanatory.

Criterion: Define ranking criteria.

Weights: Apply weighting factors to the alternatives.

Questioning:

Individual questionnaire: Question an individual.

Group questionnaire: Question the group.

Group matrix:

Add rating: Place a rating on the rating matrix.

Change rating: Change the rating on the rating matrix.

Policy development:

Draft policy: Users individually draft policy inputs.

Combine policy: Session leader combines the draft policy statements.

Editing

Document editor: Edit documents on-line in the private or group mode.

Video editor: Edit video clips on-line in the private or group mode.

Post-meeting documents: prepare post-meeting documents.

Electronic meeting/conference transcript: Produce a meeting transcript in electronic format.

Complete transcript: Produce a hardcopy meeting/conference transcript.

Snapshot: Produce a snapshot of a portion of the meeting/conference.

Summary: Produce a meeting/conference summary.

Highlights: Produce a meeting/conference highlights report.

File

Save current snapshot: Save a copy of the current whiteboard contents in a meeting/conference user's private file space.

Retrieve a snapshot: Open a copy of a saved snapshot in the meeting/conference user's private workspace.

Save session: Save a copy of whiteboard meeting/conference session activities to a video disk file. Includes audio/visual activities, if turned on.

Archive session: Save a compressed file of meeting/conference activities to a video disk archive file.

Playback a session: Play back a recorded meeting/conference session.

Send file or object: Self-explanatory.

Receive file or object: Self-explanatory.

Import file or object: Import a file or object into the meeting/conference whiteboard workspace.

Export file or object: Export a file or object from the meeting/conference whiteboard workspace.

Link to E-mail: Self-explanatory.

Sketching

Activate: Self-explanatory.

Deactivate: Self-explanatory.

All meeting/conference session drawing objects have the following:

Send description: User can send a drawing object description to other workgroup members.

Coupling status: Show if a drawing object is private, public, or has limited access.

Owner ID: Show who owns a drawing object.

Acquisition status: Show if a drawing object is being manipulated.

Request change status: Request the drawing object owner to change coupling status

Do change status: Actually change coupling status and send change to users.

Request change owner: Request the drawing object to change its owner.

Do change owner: Change the owner and send change to users.

Request to grab: Request the drawing object for permission to grab it.

Do grab: Grant or deny permission to grab a drawing object.

Where grabbed: Check where the drawing object was grabbed.

End grab: Release the grabbed drawing object.

Owned objects list: Show a list of drawing objects owned by the user.

Text

Type text: Type new text into private or shared meeting/conference workspace.

Shared workspace text: Allow editing of shared meeting/conference workspace text.

Request select text: Request permission from text owner to change selected text.

Do select text: Grant or deny permission to change selected text.

Change text: Edit the selected text.

Decision making

Access decision analysis software: Access available decision making software.

Make proposal: Self-explanatory.

Public: Self-explanatory.

Private: Self-explanatory.

Import proposal: Import a user's proposal from his private space.

Poll the group: Poll the group on a specific question.

Obtain group consensus: Allow the group to achieve consensus.

Obtain group vote: Allow the group to vote on an issue or specific question.

Task assignment: Session leader can assign specific tasks to one or more group members

Task: Define the task.

Responsibility: Specify who is responsible for task completion.

Deadline: Show the task deadline.

Send task: Transmit the task to designated workgroup users.

Access decision making database: Access a shared decision support database management system.

Database browser: Browse the decision making database.

Project management.

Kerzner (1992), Digital Tools (1993), and Sathi et al. (1988) provided source material for the project management section of the workgroup computing taxonomy.

Project management workgroup setup

Define project: Project manager defines the project and access codes.

Project name: Self-explanatory.

Project title: Self-explanatory.

Project start date: Self-explanatory.

Project manager's name: Self-explanatory.

E-mail address: Self-explanatory.

Access code (read only): Code allowing read only access.

Access code (read/write): Code allowing read/write access.

Access code (write only): Code allowing write only access.

Define project workgroup: Project manager defines a unique workgroup for a particular project and establishes membership criteria. Selected participants can access the project files depending on access level granted by the project manager.

Define access list: Project manager defines an access list.

Enter "User IDs": Project manager enters a list of user IDs or E-mail addresses.

Change access: Project manager can change the access list.

Add "User ID": Self-explanatory.

Delete "User ID" Self-explanatory.

Assign roles and access to project files: Project manager assigns group roles and access privileges.

Project manager: Read/write access. Owns the master project.

Subproject manager: Read only or write only.

Workgroup member: Read only or write only.

Top management: Read only.

Other users: None or read only.

Assign access to subproject files: Project manager assigns group roles and access privileges.

Project manager: Read only or read/write access.

Subproject manager: Read/write access. Owns the subproject.

Workgroup member: None, read only or write only.

Top management: Read only.

Other users: None or read only.

File locking: Limit read/write access to one user at a time to prevent working on out-of-date files.

Logon to project management files: Users logon to project management files.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff project management files: Users logoff project management files.

New: Open a new project file.

Open: Open an existing project file.

Request define project: Project manager requests subproject managers to define project schedule, work breakdown structure, and resources.

Import: Import files into the current project.

Export: Export current project files.

Subproject:

Explode: Open selected subproject.

Previous: Returns to previous project file.

Top level: Reopen master project.

Project management meeting/conference setup

Plan meeting/conference session: Project group member plans a unique synchronous conference or meeting session and establishes registration criteria. Selected participants register when they logon.

Open: Any authorized project group member can access the meeting/conference session.

Closed: Only designated group members can access the meeting/conference session.

Define access list: Meeting leader defines an access list.

Enter "User IDs": Session leader enters a list of user IDs or E-mail addresses.

Change access: Session leader can change the access list.

Add "User ID": Self-explanatory.

Delete "User ID": Self-explanatory.

Date: Specify date of meeting/conference.

Time: Time of meeting/conference.

Specify time: Self-explanatory.

Autoselect: Selects a meeting/conference date and time by comparing invitees calendars to choose the best time for all participants.

Topic: Specify topic of meeting/conference.

Location: Specify location of the meeting/conference.

Assign roles: Meeting leader assigns group roles.

Facilitator: Moderates meeting/conference activities.

Organizer: Organizes meeting/conference session.

Chairperson: Controls meeting/conference activities.

Attendee: A user who can comment, send messages, and can see all other user inputs.

Contributor: A user who can add comments, send messages, but cannot see other users' inputs.

Observer: observes meeting/conference activities. Cannot comment or vote.

Send invitation: Sends designated individuals an invitation to join the meeting/conference session.

Agenda: Meeting/conference organizer can develop an agenda.

Develop: Draft the agenda.

Send: Send the agenda to invitees.

Respond to invitation:

Reply: Send a message to meeting/conference organizer. Includes comments.

Will attend: Self-explanatory.

Cannot attend: Self-explanatory.

Don't know right now: Self-explanatory.

Comments: Comment on the agenda or some other issue.

Request user/schedule information: Request pre-specified information on workgroup members. Allows workgroup leader to identify appropriate meeting/conference members, see their calendars, and define their roles.

Connect meeting/conference: Meeting/conference leader connects the meeting/conference. The system passes a message to invitees that the meeting/conference is starting.

Logon to meeting/conference session: Users logon to meeting/conference session. The system passes host name, port number, and user name, to the session registrar.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff meeting/conference session: Users logoff meeting/conference session. The system sends a logoff notification message to other users that remain on meeting/conference session.

Request to join meeting/conference: A user can request permission from the session leader to join the meeting/conference after the session begins.

Approval to join meeting/conference: Leader approves or disapproves joining the meeting/conference.

Can join: Self-explanatory.

Cannot join: Self-explanatory.

Change leader: Specify another meeting/conference session leader.

Communication mode:

Text only (synchronous): Open a text only session.

Multimedia session (synchronous): Open a multimedia session.

Multiple windows: Session leader can open multiple sessions.

Number: Designate sessions.

Floor control: Select floor control mode.

Implicit request, implicit grant: Only one user can speak at a time. Floor time can be set.

Explicit request, implicit grant: Speaker can pass the floor to another group member. Floor time can be set.

Explicit request, explicit grant: The first requestor can control the floor Another user can request permission to control the floor. Floor time can be set.

No floor: More than one user can have simultaneous access to the floor. Used for brainstorming and arguing.

Gesture on: User can use gestures on the shared session whiteboard.

Point: User can point at an object or text.

Write: User can write a note.

Erase: User can erase a note.

Direct attention: User can make a motion to direct attention on the electronic whiteboard. This function is not available if audio/video is turned on.

List workgroup notes: Lists all shared workspace notes for all users.

Gesture off: Gesturing is turned off.

Audio/Video on: Video and sound links are turned on. Opens a video window at each user station.

Audio/Video off: Video and sound links are turned off. Session leader can terminate video access. Users can close their video connection.

Voice mute: User turns off the voice portion of the multimedia session.

Video mute: User turns off the video portion of the multimedia session.

Status information:

Who is on-line: Self-explanatory.

Topic: Self-explanatory.

Group leader: Self-explanatory.

Who is in control: Self-explanatory.

Save session: Save a copy of meeting/conference session activities to a video disk file. Includes audio/visual activities, if turned on.

Archive session: Save a compressed file of session activities to a video disk archive file.

Import file or object: Import a file or object into the meeting/conference whiteboard workspace.

Export file or object: Export a file or object from the meeting/conference whiteboard workspace.

Import private view into workspace: User can import private files into the shared whiteboard workspace.

Project management functions

Send E-mail: Self-explanatory.

Send project files to: Sends project files to designated recipients.

Send committee plan to workgroup members: Self-explanatory.

Request inputs from subproject managers: Self-explanatory.

Request for reports: Self-explanatory.

Plan schedule: Provide group schedule planning.

Plan resources: Provide group resource planning.

Generate ideas: Group idea generation tools.

Organize ideas: Group idea organization tools.

Transfer file: Transfer a project file to a non-workgroup user.

Scheduling and calendaring.

Campbell (1992), Greenwood (1992), Lange (1992), and On technology (1994) provided source material for the scheduling and calendaring section of the workgroup computing taxonomy.

Schedule and calendar workgroup setup

Define schedule workgroup: Workgroup leader defines a unique workgroup for a particular schedule and establishes membership criteria. Selected participants can access the schedule depending on access level granted by the workgroup leader.

Define access list: Workgroup leader defines an access list.

Enter "User IDs": Workgroup leader enters a list of user IDs or E-mail addresses.

Change access: Workgroup leader can change the access list.

Add "User ID": Self-explanatory.

Delete "User ID": Self-explanatory.

Assign roles: Workgroup leader assigns group roles.

Manager: Self-explanatory.

Workgroup member: Self-explanatory.

Top management: Self-explanatory.

Other users: Self-explanatory.

File locking: Limit read/write access to one user at a time to prevent working on out-of-date schedules.

Logon to schedule system: Users logon to workgroup schedule system.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff schedule system: Users logoff workgroup schedule system.

New: Open a new group schedule file.

Open: Open an existing group schedule file.

Import: Import files into the current group schedule.

Export: Export files from current group schedule files.

Calendar

Organize appointments: Organize appointments by type, date, or other attribute.

Reminder notices: Place a reminder notice on the calendar.

"To do" items: Add a "to do" item completion date to the calendar.

Auto-date programming: Allow a recurring date to be programmed for a pre-determined length of time.

Alarm: Self-explanatory.

Show appointments and free time: Allow other users to see existing appointments and free time.

Show free time only: Show only free time to other users.

Block view: Make the calendar private.

Scheduling an activity:

Title of activity: Self-explanatory.

Date: Self-explanatory.

Time: Self-explanatory.

Duration: Self-explanatory.

Frequency: Self-explanatory.

Private: The activity remains private.

Flexible: The activity can be viewed by other users.

Select open time: Select an open time on a user's calendar.

Change activity to a meeting: Self-explanatory.

Invite quests: Self-explanatory.

Send invitation: Self-explanatory.

Request RSVP: Self-explanatory.

Automatic acceptance: Fills calendar space with meeting requests based on a first come, first served philosophy.

No schedule conflicts: No schedule conflicts are allowed.

Scheduler

Define users: Define users in the current workgroup.

Define resources: Define resources that can be scheduled.

Conference rooms: Self-explanatory.

Equipment: Self-explanatory.

Wait list for resources: Reserve rooms when they become free.

Define resource owner: Self-explanatory.

Search for available meeting times:

List available resources: Self-explanatory.

Notify participants: Self-explanatory.

Accept or reject schedule inputs: Self-explanatory.

Track acceptance or rejection: Self-explanatory.

Creating a "to do" list

New: Add a new item to the "to do" list.

Priority: Self-explanatory.

Reminder: Add a reminder to the "to do" item.

Notes: Add a note to the "to do" item.

Send "to do" request to other workgroup members: Self-explanatory.

Select participants: Select who to send the "to do" item to.

Done: Self-explanatory.

Responding to "to do" requests

Open "to do" request: Self-explanatory.

Check on appropriate "to do" item: Self-explanatory.

Read notes: Self-explanatory.

Comments: Send information back to preparer.

Options: Determine recipient action.

Will do: Self-explanatory.

Won't do: Self-explanatory.

I'll decide later: Self-explanatory.

Done: Self-explanatory.

Reply: Send reply to "to do" preparer.

Proxies

Proxy list: Select proxy from an available list.

Select names: Choose the proxy.

Provide access: Provide the proxy with appropriate access.

Read only: Proxy can view the file.

Read/write: Proxy can view or change the file.

Accept proposals: Proxy can propose meetings and accept or reject meeting proposals on behalf of the principal.

View proxy calendar: View the proxy's calendar.

Send message to principal: Proxy send a message to the principal.

Planning a meeting/conference session: Workgroup member plans a unique synchronous conference or meeting session and establishes registration criteria. Selected participants register when they logon.

Open: Any authorized workgroup member can access the meeting/conference session.

Closed: Only designated group members can access the meeting/conference session.

Define access list: Meeting/conference leader defines an guest list.

Enter "User IDs": Session leader enters a list of user IDs or E-mail addresses.

Define guest type: Define required or optional attendance.

Required: The guest must be present to hold the meeting/conference.

Optional: The guest does not have to be present for the meeting/conference to be held.

Carbon copy: User is provided a copy of the invitation. Other users are informed.

Blind carbon copy: User is provided a copy of the invitation. other users are not informed of the copy.

Change access: Session leader can change the guest list.

Add "User ID": Self-explanatory.

Remove "User ID": Self-explanatory.

Date: Date of meeting/conference.

Specify date: Self-explanatory.

Time: Time of meeting/conference.

Specify time: Self-explanatory.

Autoselect: Selects a meeting/conference date and time by comparing invitees calendars to choose the best time for all participants.

Topic: Topic of meeting/conference.

Location: Specify location of the meeting/conference.

Assign roles: Meeting leader assigns group roles.

Facilitator: Moderates meeting/conference activities.

Organizer: Organizes meeting/conference session.

Chairperson: Controls meeting/conference activities.

Attendee: A user who can comment, send messages, and can see all other user inputs.

Contributor: A user who can add comments, send messages, but cannot see other users' inputs.

Observer: Observes meeting/conference activities. Cannot comment or vote.

Notes: Meeting leader can compose a note to add to the invitation.

Send invitation: Sends designated individuals an invitation to join the meeting/conference session.

Agenda: Meeting/conference organizer can develop an agenda.

Develop: Draft the agenda.

Send: Send the agenda to invitees.

Respond to invitation:

Received, not yet replied:

Reply: Send a message to meeting/conference organizer. Includes comments.

Will attend: Self-explanatory.

Cannot attend: Self-explanatory.

Will decide later: Self-explanatory.

Comments: Comment on the agenda or some other issue.

Define proxy: User can define a proxy to attend the meeting/conference.

In/Out board

View schedules: View schedules at a higher level. Who is in town, on vacation, and sick. Show information on how to locate people.

Shared databases.

Celentano et al. (1991), Greif (1992), Greif and Sarin (1988), Holtham (1993), Khoshafian et al. (1992), and Schwartz (1992) provided source material for the shared databases section of the workgroup computing taxonomy.

Shared database workgroup setup

Define shared database workgroup: Workgroup leader defines a unique workgroup for a particular set of databases and establishes membership criteria. Selected participants can access the database files depending on access level granted by the database manager.

Database name: Self-explanatory.

User name: Self-explanatory.

Define access rights: Define user access rights.

User names: Self-explanatory.

Owner: The database owner.

Transfer ownership: Current owner can transfer database ownership to another user.

Share ownership: Owner can share ownership with another user.

Delete permission: Self-explanatory.

Access restrictions: Limit access to database files.

User ID: Self-explanatory.

Attribute: Self-explanatory.

Filed: Only designated users can access filed records.

Category: Only designated users can access defined categories of records.

Class: Only designated users can access defined classes of records.

Read-only: Allow a user read-only access to the database.

Read and write: Allow user to read and change to the database.

Append: Allow user to append information to the database, but not change existing information.

Logon to database system: Users logon to workgroup database files.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff database system: Users logoff workgroup database files.

File locking: Limit read/write access to one user at a time to prevent working on out-of-date databases.

Lock database: Only one user has access at a time.

Request lock release: User can request immediate access to a database.

Negotiate lock: Lock is automatically passed between users.

Reserve: User can request access and wait until lock is released by another user.

Data compression:

On: Turn on data compression.

Text: Self-explanatory.

Video: Self-explanatory.

Graphics: Self-explanatory.

Audio: Self-explanatory.

Combination: Self-explanatory.

All: Self-explanatory.

Off: Turn off data compression.

Transactions

Begin transaction: Transaction is initiated by a user. Lock is released after a specified period of inactivity or no transaction committal. Database is not given a new version number.

Commit transaction: Transaction is entered into the database file. Lock is released after a specified period of inactivity. Database is given a new version number.

Roll back transaction: Transaction is removed from the database. Lock is released after a specified period of inactivity. Database returns to previous version number.

Manipulation

Filing: Place record information into the database.

Public area: Records are in an area accessible to most workgroup users.

Private area: Records are filed in an area accessible to designated workgroup users.

Filing rules: File by a particular class.

Sender: Self-explanatory.

Receiver: Self-explanatory.

Reference to: Self-explanatory.

Copy of: Self-explanatory.

View of: Self-explanatory.

Archive information: Place portions of the database into archive files.

Retrieval: Retrieve information from the database.

Define query: Define the retrieval query.

Retrieve based on "as a result of": Self-explanatory.

Retrieve based on "by": Self-explanatory.

Retrieve based on "submitted by": Self-explanatory.

Retrieve based on "came from": Self-explanatory.

Retrieve based on "signed by": Self-explanatory.

Retrieve based on unique query: Self-explanatory.

Search: Search the database for specific records or information.

Full text search: Self-explanatory.

Semantic search: Search using semantic rules.

Workflow management.

Bair (1993), Bock (1992), Khoshafian (1992), Mashak (1992), Palermo and McCready (1992), and Schael and Zeller (1993) provided source material for the workflow management section of the workgroup computing taxonomy.

Workflow design

Define workflow cycle: Responsible user defines the workflow process.

Task definition: Define the workflow tasks.

Workflow rules: Define the workflow rules.

Static: Use predetermined flow rules.

Dynamic: Program new workflow rules.

Define rules: Define new rules using a rule-based language.

Use electronic forms: Select available electronic forms to embed into workflow cycle.

Define criteria: Define criteria that are used to manage the workflow.

Procedures that state: Procedures must state this before work can be routed.

What must precede these: State what action must happen before work can continue the flow.

Activity for it to execute: State what activity must occur before work can continue the flow.

Define action items: Define known action items.

Modify workflow: Self-explanatory.

Copy workflow: Self-explanatory.

Routing or branching

Define group: Define the workflow group or groups.

Assign roles: Assign roles to a group or its members.

Completor: Complete a workflow action or electronic form.

Information provider: Provide information on a form or to another user.

Reviewer: Review a workflow action or electronic form.

Coordinator: Coordinate on a workflow action or electronic form.

Approver: Approve a workflow action or electronic form.

Report generator: Generate a workflow report.

Assign action to an individual: Assign specific actions to a user.

Assign actions to a group: Assign action to a workflow group.

Group task assignment: Select a designated role member of allow system to load balance work within the group.

Load balancing: Load balance work within a specified group based on actual workload or user availability.

Next available: Select the next available group user to process the action.

Designate role member: Select a particular group role member to process the action.

Designate specific member: Select a specific workgroup user to process the action.

Describe activities:

Refuse role or action: Refuse to process the action.

Claim role: Request permission to process the action as a role member.

Create document: Create a document to attach to the workflow action.

Integrate with personal "to do" list: Integrate workflow actions into a user's personal "to do" list.

Define deadlines: Define action deadlines.

Flow control: Select how the work will flow between users or groups.

Serial: Tasks or actions processed serially.

Parallel: Tasks process in parallel based on the workflow rules and criteria.

Access control

Define access: Designate the workflow users or groups.

By group: Self-explanatory.

By user: Self-explanatory.

Permission by workflow: Permit groups or users access to specific workflows.

Report permission: Specify who can generate reports based on workflow data.

By workflow: Self-explanatory.

By report: Self-explanatory.

Modification rights: Specify who can modify a workflow.

By group: Self-explanatory.

By user: Self-explanatory.

Permissions

Delegate: Allow a user or group to delegate workflow task to another user or group.

Reject: Allow a user or group to reject processing a workflow task.

Correct mistake by objection: Allow a user or group to object to a mistake and recommend a correction.

Withdraw: Allow a user or group to withdraw a workflow action.

View attached documents: Allow a user or group to view documents attached to the workflow task.

Add to attached documents: Allow a user or group to add to attached documents.

Modify: Allow a user or group to modify attached documents.

Complete: Allow a user or group to complete a task designated for completion by another user or group.

Logon to workflow system: Users logon to workgroup workflow system.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff workflow system: Users logoff workflow system.

Approvals

Digital Signature: Request a signature on the task or document.

Encryption

Encryption on: Self-explanatory.

Encryption off: Self-explanatory.

Status monitoring

View steps: Allow workflow leader to view status of the workflow task.

Last step: Self-explanatory.

Current step: Self-explanatory.

Next step: Self-explanatory.

View deadlines: Self-explanatory.

By date: Self-explanatory.

By project: Self-explanatory.

By type of action: Self-explanatory.

Define views: Self-explanatory.

Overdue alert: Activate alert showing overdue tasks and actions.

Pending actions: Show pending tasks and actions.

Completed: Show competed tasks and actions.

Filing

Archive to database: Archive workflow session to database file.

Access database: Retrieve the workflow session database file.

Utilities

Send E-mail message: Self-explanatory.

Writing and editing.

Baecker et al. (1993), Baydere et al. (1993), Denley et al. (1993), Knister and Prakash (1990), Neuwirth et al. (1990), Posner and Baecker (1992), Sharples (1993), Sharples et al. (1993), Shen and Dewan (1992), and Sobiesiak and Myopoulos (1991) provided source material for the writing and editing section of the workgroup computing taxonomy.

Writing and editing workgroup setup

Define writing and editing workgroup: Lead writer or editor defines a unique workgroup for a particular set of documents and establishes membership criteria. Selected participants can access the document files depending on access level granted by the workgroup leader.

Document name: Self-explanatory.

User name: Self-explanatory.

Session control

Roles: Define writing or editing session roles.

Lead writer: Session lead writer.

Co-writer: A writer with the same privileges as the lead writer.

Writer: A writer that has restricted privileges.

Consultant: Can provide comments, but cannot change the document.

Editor: Edits the document.

Reviewer: Reviews and makes comments, but cannot change the document.

Activities: Define writing activities.

Brainstorm: Brainstorm ideas and issues.

Research: Research ideas and issues. Writers add comments.

Initial plant: Draft an outline.

Write: Write the document.

Control changes:

Edit document: Self-explanatory.

Final edit: Edit the document for the last time then freeze contributions.

Review: Users review the document.

Writing strategies:

Single writer: One writer writes most of a document while other writers assist.

Scribe: One writer writes down most of a group's thoughts and comments while other writers discuss the ideas that will be expressed in a document.

Separate writers: Document is broken down into parts that are authored by separate writers.

Joint writing: Several group members compose the text together.

Access control

Document control: Sets up how the document will be accessed depending on the writing strategy.

Centralized: One document is accessed by one writer at a time.

Relay: Document is passed between users.

Independent: Document sections are accessed by one writer at a time.

Shared: Several writers can access a document simultaneously.

Permissions:

User ID: Self-explanatory.

Read/write: Allow a user both read and write access.

Comment: Allow a user to make comments, but not change the document.

Read only: Allow a user to read the document, but not make comments or changes.

Change access: Change a user's access.

Logon to writing system: Users logon to writing system files.

Enter "User ID": Self-explanatory.

Enter "Password": Self-explanatory.

Logoff writing system: Users logoff writing system files.

Request to join writing group: A user can request permission from the session leader to join the writing group after the session begins.

Approval to join conference: Leader approves or disapproves joining the writing group.

Can join: Approve joining the session. Sends a message to session users that someone has joined the session once they logon.

Cannot join: Self-explanatory.

Sessions: Manage document sessions.

Create session: Create a new document session.

Merge session: Merge two or more document sessions.

Leave session: Leave a document session.

Document segmentation: Segment the document.

Separate document sections: Separate a document into two or more sessions.

Join document sections: Join two or more document sections.

Writing rules

Style rules: Lead writer defines workgroup style rules.

Content rules: Lead writer defines workgroup content rules.

Editing rules: Lead writer or editor defines workgroup editing rules.

Lead writer select rules: Self-explanatory.

Editor select rules: Self-explanatory.

Both select rules: Self-explanatory.

Revision control

Small change: Make change without marking the change. An example is a spelling error.

Many additions: Mark change so that it stands out from text. Let the responsible writer make the changes.

Delete: Mark text for deletion. Let responsible writer approve the deletion.

Change history: Mark top of text with change history.

Version control

Versions allowed: Allow lead writer to select how many document versions are allowed.

Single: Self-explanatory.

Multiple: Self-explanatory.

Parallel: Write on different sections of a document in real-time.

Sequential: Pass the document between writers in real-time or at the convenience of a writer.

Reciprocal: Write on the same document section at the same time.

Automatic conflict extraction: Automatically identifiy a writing conflict and present it to the lead writer or all authors for resolution.

Present to lead author: Self-explanatory.

Present to all authors: Self-explanatory.

Drawing/sketching

All session drawing objects have the following:

Send description: User can send a drawing object description to other workgroup members.

Coupling status: Show if a drawing object is private, public, or has limited access.

Owner ID: Show who owns a drawing object.

Acquisition status: Show if a drawing object is being manipulated.

Request change status: Request the drawing object owner to change coupling status

Do change status: Actually change coupling status and send change to users.

Request change owner: Request the drawing object to change its owner.

Do change owner: Change the owner and send change to users.

Request to grab: Request the drawing object for permission to grab it.

Do grab: Grant or deny permission to grab a drawing object.

Where grabbed: Check where the drawing object was grabbed.

End grab: Release the grabbed drawing object.

Owned objects list: Show a list of drawing objects owned by the user.

Status monitoring

Revision status: Show number of documents and revisions.

Tools

Import text or objects: Import text or objects into the document file.

Export text or objects: Export text or objects from the document file.

Information filtering: Develop information filtering criteria and rules.

Send E-mail message: Self-explanatory.

Audio/video

Audio/video on: Video and sound links are turned on.

Audio/video off: Video and sound links are turned off. Session leader can terminate video or audio access. Users can close their video or audio connection.

Video channel mode: Select how many video windows are displayed on the terminal.

Multiple image mode: Select how many video windows.

Single image mode: Select single image and toggle between several video images.

Preview: Check on a user's camera output.

Audio select:

All channels: Hear all open audio channels.

Selected channel: Select a specific audio channel.

Select: Select an audio channel. It does not have to match the video channel.

Selected video channel audio: Select the audio channel that matches the selected video channel.

Audio mute: User can mute audio output.

Video mute: User can mute video output.

Open multimedia conference: Open a link to a multimedia conference.

Appendix C

Workgroup Computing Functional Areas Showing

Workgroup Computing Tasks and Primitives

Black squares show workgroup computing tasks or primitives that are common to different workgroup computing functional areas. White squares show where tasks and primitives can be extended to unsupported workgroup functional areas.

Workgroup Computing Functional Area / **Workgroup Computing Task/Primitive**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Session setup											
Define session files	■	■	❑		❑	❑	■	■	■	■	■
Start/connect session	■	■	■		■	■	■	❑	❑	❑	❑
Access control	■	■	■	■	■	■	■	■	■	■	■
Change access	■	■	■		■	■	■	■			
Assign roles			■		■	■	■	■		■	■

Workgroup Computing Functional Area / **Workgroup Computing Task/Primitive**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Send invitation	■	■	■		■	■	□	□			□
Respond to invitation	□	□	□		■	■	□	□			□
Logon to session	■	■	■	■	■	■	■	■	■	■	■
Logoff session	■	■	■	■	■	■	■	■	■	■	■
Request to join group	■	■	■		■	■	■			■	■
Approval to join group	■	■	■		■	■	■			■	■
Change session leader	■	■	■		■	■	■	□		■	■
Define editing mode	■	■	■		■	■	■	□	■	□	■
Gesturing	■	■	■		■	■	■				□
Status information	■	■	■	■	■	■	■	□	□	■	□
Audio/video	■	■	■	■	■	■	■	□			■
Voice mute	■	□	■		■	■	■	□			■

Workgroup Computing Functional Area / **Workgroup Computing Task/Primitive**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Video mute	■	❑	■		■	■	■	❑			■
Files/sessions											
New/open	■	■	■		■	■	■	■	■		■
Workspace layout	■	■	■		■	■	❑	■	❑	■	■
Merge session	■	■	■		■	■	❑	❑	❑	❑	■
Split session	■	■	■		■	❑	❑	❑	❑	❑	■
Save session	■	■	■		■	■	■	❑	❑	❑	■
Recall session	■	■	■		❑	❑	❑	❑	❑	❑	❑
Save current snapshot	■	■	■		■	■	❑	❑	❑		❑
Retrieve snapshot	■	■	■		■	■	❑	❑	❑		❑
Archive session	■	■	■		■	■	■	❑	❑	■	❑
Playback session	❑	❑	❑		❑	■	❑	❑	❑	❑	❑

Workgroup Computing Functional Area / Workgroup Computing Task/Primitive	Computer-aided design	Drawing and presentations	Decision support	Electronic mail	Interactive communication	Meeting/conference support	Project management	Scheduling and calendaring	Shared databases	Workflow management	Writing and editing
Import file or object	■	■	■	■	■	■	■	■	■	❑	■
Export file or object	■	■	■	■	■	■	■	■	■	❑	■
Import private view	■	■	■		■	■	■	❑			❑
Define private space	■	■	❑		❑	■	❑	❑	❑		❑
Arrange											
Clear group surface	■	■	❑		❑	❑	❑	❑			❑
Communication mode											
Whiteboard	❑	❑	■		■	■	❑	❑			❑
E-mail	❑	❑	■	■	■	❑	■	■	❑	■	■
Bulletin board	❑	❑	■	■	❑	❑	❑	❑	❑	❑	❑
Text only					■	■				■	■
Multimedia	■	■	■	■	■	■	❑	❑	■	■	■

Workgroup Computing Functional Area / **Workgroup Computing Task/Primitive**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Voice mail	❑	❑	❑	■	❑	❑	❑	❑	❑	❑	❑
Phone support	❑	❑	❑		❑	■	❑	❑	❑	❑	❑
Conference	❑	❑	❑		❑	■	■	■	❑	❑	❑
Floor control	❑	❑	❑		❑	■			■		■
Decision tools											
Select decision style			■		❑	❑	❑				
Select decision model			■		❑	❑	❑				
Freeze contributions	❑	❑	■		❑	❑	❑			❑	❑
Poll the group	❑	❑	■		■	■	❑			❑	❑
Obtain consensus	❑	❑	■		■	■	❑			❑	❑
Obtain vote	❑	❑	■	■	■	■	❑			❑	❑
Assign tasks	❑	❑	■		■	■	❑	❑	❑	■	❑

Workgroup Computing Functional Area / **Workgroup Computing Task/Primitive**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Make proposal	❑	❑	❑		❑	■	❑	❑	❑	❑	❑
Develop decision rules			■			❑	■			■	
Display views											
Access private view	■	■	❑		❑	❑	❑	❑			❑
Move to empty workgroup view	■	■									❑
Move to common workgroup view	■	■									❑
Show virtual position	■	■			❑	❑					❑
Draw											
Drawing object status	■	■			■	■					■
Get object	■	■			■	■					■
Owned object list	■	■			■	■					■

Workgroup Computing Functional Area / **Workgroup Computing Task/Primitive**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Text											
Type shared text	■	■			■	■				■	■
Select shared text	■	■			■	■				■	■
Change shared text	■	■			■	■				■	■
Writing/editing											
Activities	❑	❑	❑		❑	❑	❑	❑		❑	■
Strategies	❑	❑	❑		❑	❑	❑	❑		❑	■
Document control	❑	❑	❑		❑	❑	❑	❑		❑	■
Permissions	❑	❑	❑		❑	❑	❑	❑		❑	■
Writing rules	❑	❑	❑		❑	❑	❑	❑		❑	■
Revision control	❑	❑	❑		❑	❑	❑	❑		❑	■
Version control	❑	❑	❑		❑	❑	❑	❑		❑	■

Workgroup Computing Functional Area / **Workgroup Computing Task/Primitive**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Document editing	❑	❑	❑		❑	■	❑	❑		■	■
Post-meeting documents			❑		❑	■	❑		❑		
Video editing	❑	❑			❑	■	❑		❑		
Database											
Access database	❑	❑	■	❑	■	■	❑	❑	■	■	❑
Browse database	❑	❑	❑	❑	❑	■	❑	❑	■	❑	❑
Manipulate database	❑	❑	■	❑	❑	■	❑	❑	■	❑	❑
Filter database	❑	❑	■	❑	❑	■	❑	❑	■	❑	❑
Sort database	❑	❑	■	❑	❑	❑	❑	❑	■	❑	❑
Mail											
Mail link	❑	❑	❑	■	■	■	■	❑	❑	■	■
Group address book	❑	❑	❑	■	■	■	■	❑	❑	■	■

Workgroup Computing Functional Area / **Workgroup Computing Task/Primitive**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Return receipt	❑	❑	❑	■	■	■	■	❑	❑	■	■
Response request	❑	❑	❑	■	■	■	■	❑	❑	■	■
Access permission	❑	❑	❑	■	■	■	■	❑	❑	■	■
Signature authority	❑	❑	❑	■	❑	❑	❑	❑	❑	❑	❑
Share a folder	❑	❑	❑	■	■	■	■	❑	❑	■	■
Schedule/calendar											
Calendar	❑	❑	❑		❑	■ Link	■ Link	■	❑	❑	❑
Scheduler	❑	❑	❑		❑	■ Link	■ Link	■	❑	❑	❑
"To do" list	❑	❑	❑		❑	■ Link	■ Link	■	❑	❑	❑
Proxy			❑		❑	❑	❑	■		❑	
Ideas											
Idea generation	❑	❑	❑		❑	■	❑			❑	❑

Workgroup Computing Functional Area **Workgroup Computing Task/Primitive**	**Computer-aided design**	**Drawing and presentations**	**Decision support**	**Electronic mail**	**Interactive communication**	**Meeting/conference support**	**Project management**	**Scheduling and calendaring**	**Shared databases**	**Workflow management**	**Writing and editing**
Idea organization	❑	❑	❑		❑	■	❑			❑	❑
Prioritizing	❑	❑	❑		❑	■	❑			❑	❑
Policy development	❑	❑	❑		❑	■	❑			❑	❑
Workflow											
Define rules	❑	❑	❑	❑	❑	❑	❑	❑	❑	■	❑
Define criteria	❑	❑	❑	❑	❑	❑	❑	❑	❑	■	❑
Routing	❑	❑	❑	❑	❑	❑	❑	❑	❑	■	❑
Assign action	❑	❑	❑	❑	❑	❑	❑	❑	❑	■	❑
Support functions											
Encryption	❑	❑	❑	■	■	❑	❑	❑	❑	■	❑
Data compression	■	■	❑	■	❑	❑	❑	❑	■	❑	❑
In/out board	❑	❑	❑	❑	❑	❑	❑	■	❑	❑	❑

Workgroup Computing Functional Area / Workgroup Computing Task/Primitive	Computer-aided design	Drawing and presentations	Decision support	Electronic mail	Interactive communication	Meeting/conference support	Project management	Scheduling and calendaring	Shared databases	Workflow management	Writing and editing
Digital signature	❑	❑	❑	❑	❑	❑	❑	❑	❑	■	❑

Appendix D

Workgroup Computing Application Programming Interface (API) Specifications

The following workgroup computing API specifications were developed for workgroup computing tasks and primitives identified in the workgroup computing taxonomy and architecture. APIs are divided into two groups: workgroup management and workgroup support. API groupings for workgroup management include session setup, file and session management, display and arrange views, communication mode, and other tasks including encryption, data compression, and a digital signature. Workgroup support contains API groupings for decision tools, drawing and graphics, text, writing and editing, databases, mail, calendar and scheduling, ideas, and workflow support.

Each API includes a purpose, description, uses, the value the API returns when it is executed by an application, and other related APIs. Returns defines a value or set of values that is returned to the program as the API function is executed. Returned values can be integers, a previous value, no returned value (a null or void), Boolean True, if a value or configuration for a function is determined or successful, or Boolean False, if an error or alternate configuration occurs. An API may return a dialog box, a file handle, a window handle, or a handle to a pop-up window or menu (Conger, 1992).

Workgroup Management APIs

SESSION SETUP

Define session files

CreateSessionFile

Purpose: Open or create a workgroup session file.

Description: This function opens an existing workgroup session file. If necessary, it will create a new file.

Uses: Normally used to open a workgroup file related to a specific workgroup session.

Returns: If valid, this function returns a handle to the session file. If an error occurs, it returns an invalid handle value.

See Also: *CloseSessionFile*

Start/connect session

WGSessionOpen

Purpose: Start a workgroup session.

Description: This function starts a workgroup session defined by workgroup leader.

Uses: Normally used to initiate a specific workgroup session.

Returns: If valid, this function returns a handle to the workgroup session. It returns a Null on error.

See Also: *WGSessionClose*

WGSessionClose

Purpose: End a workgroup session.

Description: This function ends a workgroup session.

Uses: Normally used to terminate a specific workgroup session.

Returns: If the workgroup session is closed, this function returns Boolean True. It returns Boolean False if the workgroup session does not close.

See Also: *WGSessionOpen*

Access control

DefineAccessList

Purpose: Define the workgroup session access list.

Description: The workgroup leader can specify access to the workgroup session by listing user IDs for individuals that are permitted access to the workgroup session.

Uses: Used to compare user IDs or E-mail addresses with actual logon values to control access to the workgroup session.

Returns: If a match is successful, this function returns Boolean True. It returns Boolean False if a match does not occur for a user ID, E-mail address, or a user's password.

See Also: *ChangeAccess*

Change access

ChangeAccess

Purpose: Change the workgroup session access list.

Description: Allow the workgroup leader to change access to the workgroup session by adding or deleting user IDs or E-mail addresses.

Uses: Used to change and control access to the workgroup session once the session has been setup.

Returns: This function returns a dialog box that allows the workgroup leader to add or delete specific user IDs or E-mail addresses.

See Also: *DefineAccessList*

Assign roles

AssignWGRoles

Purpose: Assign user roles for workgroup activities.

Description: Allow the workgroup leader to specify user roles for various workgroup activities. Normally, this functions permits the workgroup leader to specify various users different privileges or access to workgroup files.

Uses: Used to define user roles for writing, decision making, project management, or other workgroup functions.

Returns: This function returns a dialog box that allows the workgroup leader to specify user roles for each specific workgroup activity where a role is required.

See Also: *ChangeWGRoles*

ChangeWGRoles

Purpose: Change user roles for workgroup activities.

Description: Allow the workgroup leader to change user roles for various workgroup activities either based on his prerogative or a request from a user. Normally, this function permits the workgroup leader to change user privileges or access to workgroup files.

Uses: Used to change user roles for writing, decision making, project management, or other workgroup functions.

Returns: This function returns a dialog box that allows the workgroup leader to change user roles for any specified activity.

See Also: *AssignWGRoles*

Send invitation

SendWGInvitation

Purpose: Send an invitation to join a workgroup session.

Description: Sends an invitation to workgroup users to join a specific workgroup session. The session could be a meeting, conference, decision making session, communication session, or another synchronous or asynchronous activity.

Uses: Used by a workgroup member to send an invitation to join a specific workgroup session.

Returns: This function returns a dialog box to a designated user terminal that allows the user to choose a response to the sender.

See Also: *RepondWGInvitation*

Respond to invitation

RepondWGInvitation

Purpose: Respond to the sender's invitation.

Description: Allow the user to respond to the sender stating the user's intent to attend, not attend, or decide later about attending an activity. A reply can include comments to the workgroup session organizer.

Uses: Normally used by the invited user to state a desire to attend the workgroup activity.

Returns: This function returns a message handle to the sender for a message stating the user's intention and comments about joining the workgroup session.

See Also: *SendWGInvitation*

Logon to session

LogonWGSession

Purpose: Logon to shared workspace.

Description: Users Enter user ID and password to logon to a specified workgroup session. The system checks the user ID, E-mail address, and associated user password for a match on the access list. After logon the system passes host name, port number, and user name, to the workgroup session registrar.

Uses: Used by workgroup users to logon to a specified workgroup session.

Returns: If a logon match is successful, this function returns an integer that is the ID number of the workgroup user, and a status message handle to the system. If a match does not occur for a user ID, E-mail address, or a users's password, it returns a negative value.

See Also: *LogoffWGSession*

Logoff session

LogoffWGSession

Purpose: Logoff shared workspace.

Description: Users enter logoff or exit from a menu for a specified workgroup session. The system logs off the user and passes host name, port number, and user name to the workgroup session registrar.

Uses: Used by workgroup users to logoff a specified workgroup session.

Returns: If logoff is successful, this function returns an integer, zero, and a status message handle to the system. If logoff fails, it returns a negative value.

See Also: *LogonWGSession*

Request to join group

RequestJoinWG

Purpose: Request permission to join a workgroup session.

Description: A user can send a message to the workgroup leader requesting permission to join a specific workgroup session. The session could be a meeting, conference, decision making session, communication session, or another synchronous or asynchronous activity.

Uses: Used to join a workgroup session that a user was not invited to.

Returns: This function returns a message handle to the workgroup leader for a message stating the user's desire to join the workgroup session.

See Also: *ApproveJoinGroup*

Approval to join group

ApproveJoinWG

Purpose: Approve or disapprove a request to join a workgroup session.

Description: Sends a message to a user stating approval or disapproval to join a specific workgroup session. The session could be a meeting, conference, decision making session, communication session, or another synchronous or asynchronous workgroup activity.

Uses: Used to approve or disapprove a user's request to join a workgroup session.

Returns: This function returns a message handle to the user for a message stating the workgroup leader's approval or disapproval to join the workgroup session. If joining is approved, this function returns an integer, the ID number of the workgroup user, and a status message handle to the system. If joining is not approved, it returns a negative value.

See Also: *ApproveJoinGroup*

Change session leader

DefineSessionLeader

Purpose: Define the workgroup session leader.

Description: Authorized users can specify the workgroup session leader. Allows the authorized user to select from a predefined set of users or by typing a user ID.

Uses: Used to designate the leader of a specific workgroup session.

Returns: This function returns a dialog box that permits the selection and a message handle to the system for a message stating who the workgroup leader is for a particular session. This function also returns an integer, the ID number of the workgroup leader. It returns a Null on error.

See Also: *ChangeSessionLeader*

ChangeSessionLeader

Purpose: Change the workgroup session leader.

Description: The current workgroup session leader can specify a new workgroup session leader. Allows the selection to be made from a predefined set of users or by typing a user ID.

Uses: Used by the current workgroup session leader to designate the new leader of a workgroup session.

Returns: This function returns a dialog box that permits the selection and a message handle to the system for a message stating who the new workgroup leader is for a particular session. This function also returns an integer, the ID number of the new workgroup leader. It returns a Null on error.

See Also: *DefineSessionLeader*

Define editing mode

DefineWGEditingMode

Purpose: Specify or change access to shared workgroup objects.

Description: Allows the workgroup leader to specify or change the object editing mode that will be used in the workgroup session. A dialog box allows the selection of one of the following modes: designated mode; baton mode; first-come, first-served; or free.

Uses: Used by the workgroup session leader to determine how objects are accessed in a workgroup session. In the designated mode, only one owner can own an object or text. In the baton mode, an object owner can pass an owned object or text to another group member. First-come, first-served allows the first requestor access to a design object or text. Another user can request permission to access a non-owned object or text. In the free mode, more than one user can have simultaneous access to a design object or text. Free mode can be used for brainstorming.

Returns: This function returns a dialog box that allows mode selection and an integer, the selected editing mode. It returns a Null on error.

See Also: None

Gesturing

GestureOn

Purpose: Turn on gesturing functions.

Description: Allow a user to turn on gesturing functions including pointing, writing a note, erasing a note, directing attention using the cursor, and obtaining a list of all shared workspace notes. Gesturing is not available if audio/video is turned on.

Uses: Allow a user to write on or direct attention by pointing to some portion of the shared workgroup area during a session.

Returns: This function returns a handle to the created gesture. It returns a Null on error.

See Also: *GestureOff*

GestureOff

Purpose: Turn off gesturing functions.

Description: Allows a user to turn off gesturing functions including pointing, writing a note, erasing a note, directing attention using the cursor, and obtaining a list of all shared workspace notes.

Uses: Used to turn off gesturing functions.

Returns: This function returns a Boolean True if gesture is turned off. It returns a Boolean False on error.

See Also: *GestureOn*

Status information

ObtainWGStatusInfo

Purpose: Obtain system status information.

Description: Allows a user to determine who is on-line, the topic of the current session, the name of the group leader, and who is in control of the session ID that is being facilitated.

Uses: Used to obtain workgroup status information and display it in a status window.

Returns: This function returns a pop-up status window that displays the requested workgroup status information. It returns a null on error.

See Also: None

Audio/video

WGVideoOn

Purpose: Turn on audio/video system.

Description: Open workgroup audio and video devices and turn on video and sound links. Open a video window at each user station.

Uses: Used to establish an audio/video connection between local or distance workgroup session locations.

Returns: This function returns a variable pointing to the handle of each audio/video output device. It also returns a handle for the window that processes the output from the audio/video device. If successful, this function returns a zero. If unsuccessful, it returns an error constant.

See Also: *WGVideoOff*

WGVideoOff

Purpose: Turn off audio/video system.

Description: Closes workgroup audio and video output devices and turns off video and sound links. Closes the video window at each user station.

Uses: Used to close audio/video output devices and terminate an audio/video connection between local or distance workgroup session locations.

Returns: If successful, this function returns a zero. If unsuccessful, it returns an error constant.

See Also: *WGVideoOn*

Voice mute

WGAudioMute

Purpose: Turn off audio system. Video remains on.

Description: Close workgroup audio output devices and turn off sound links. This function can be toggled on and off.

Uses: Used to open or close audio output devices and begin or terminate an audio connection between local or distance workgroup session locations.

Returns: If successful, this function returns a zero. If unsuccessful, it returns an error constant.

See Also: *WGVideoMute*

Video mute

WGVideoMute

Purpose: Turn off video system. Audio system remains on.

Description: Open or close a workgroup video output device and turn off video links. This function can be toggled on and off.

Uses: Used to open or close a video output device and begin or terminate a video connection between local or distance workgroup session locations.

Returns: If successful, this function returns a zero. If unsuccessful, it returns an error constant.

See Also: *WGAudioMute*

FILE/SESSION MANAGEMENT

New/open

GetOpenFileName (Schildt, Pappas, & Murray, 1994a, p. 213)

Workspace layout

CreateWGWorkspaceLayout

Purpose: Setup or change the workgroup session workspace layout.

Description: Using a dialog box, the workgroup session leader can setup and change the shared workspace layout. The function provides a dialog box that allows selection of a single window or multiple windows. A drawing or sheet size and 2D or 3D views can be specified. Session leader can add or delete multiple document sections or workspace layers.

Uses: Used to configure the session workspace to accommodate a particular primary workgroup function including writing, drawing, management tasks, personal tasks, or resource management. This function sends all necessary messages to create a window, work layers, and views specified by the workgroup session leader.

Returns: This function establishes pointers to parameters specifying the workspace class and attributes. It sends messages to activate the selected windows and attributes. The function returns the handle of the new workspace. It returns a Null on error.

See Also: *CreatePrivateSpace*

Multiple windows

CreateWindow (Schildt et al., 1994a, p. 726)

Merge session

WGSessionMerge

Purpose: Combine two or more workgroup sessions into a single session.

Description: Allow the workgroup session leader to merge two or more workgroup sessions in shared workspace. This function can combine the physical activities of multiple workgroups into a single session.

Uses: Normally used by the workgroup session leader for combining various workgroup activities into a single session.

Returns: This function returns a dialog box that allows the workgroup session leader to designate workgroup sessions that should be combined. Grabs the handles of the specified session files and merges them into a single session. If valid, this function returns a handle to the combined session file. If an error occurs, it returns a Null.

See Also: *WGSessionSplit*

Split session

WGSessionSplit

Purpose: Split a single workgroup session into two or more workgroup sessions.

Description: Allow the workgroup session leader to divide a single workgroup session into several sessions in separate workspaces.

Uses: Normally used by the workgroup session leader for dividing various workgroup activities into more than one session based on complexity and sensitivity of the work.

Returns: This function returns a dialog box that allows the workgroup session leader to designate workgroup sessions that should be created. Grabs the handle of the specified session file and splits it into a designated number of sessions in separate workspace. If valid, this function returns a handle to each resulting session file. If an error occurs, it returns a Null.

See Also: *WGSessionMerge*

Save session

WGSessionSave

Purpose: Save the open workgroup session file.

Description: Allow the workgroup session leader to save a workgroup session to a designated storage device.

Uses: Normally used by the workgroup session leader to save the session file to the designated storage device.

Returns: This function returns a dialog box that allows the workgroup session leader to designate the file name and storage device. Grabs the handle of the specified session file and writes the file to a specified device. If the save is successful, this function returns an integer for the saved file. If an error occurs, it returns a zero.

See Also: *WGSessionRecall*, *WGSessionArchive*

Recall session

WGSessionRecall

Purpose: Retrieve and open a saved workgroup session file.

Description: Allow the workgroup session leader to retrieve and open a saved workgroup session from a designated storage device.

Uses: Normally used by the workgroup session leader to open a saved session file to allow additional work by the designated workgroup.

Returns: This function returns a dialog box that allows the workgroup session leader to designate the file name and storage device. Grabs the handle of the specified session file and opens the file in the designated workspace. If the retrieve is successful, this function returns a handle for the opened file. If an error occurs, it returns a Null.

See Also: *WGSessionSave*

Save current snapshot

SaveWGSnapshot

Purpose: Save a snapshot of an open workgroup session file.

Description: Allow the a workgroup user to save a snapshot of a workgroup session to a designated storage device.

Uses: Normally used by a workgroup user to save a snapshot of the current session file to a designated storage device.

Returns: This function returns a dialog box that allows the workgroup user to designate the file name and storage device. Grabs the handle of the specified session file and writes the file to a specified storage device. If the save is successful, this function returns an integer for the saved file. If an error occurs, it returns a zero.

See Also: *RetrieveWGSnapshot*

Retrieve snapshot

RetrieveWGSnapshot

Purpose: Retrieve and open a saved workgroup snapshot file.

Description: Allow a workgroup user to retrieve and open a saved workgroup session snapshot from a designated storage device.

Uses: Normally used by a workgroup user to open a saved session snapshot file to allow additional work either privately or by the current workgroup.

Returns: This function returns a dialog box that allows a workgroup user to designate the file name and storage device. Grabs the handle of the specified session file and opens the file in the designated private or public workspace. If the retrieve is successful, this function returns a handle for the opened file. If an error occurs, it returns a

Null.

See Also: *SaveWGSnapshot*

Archive session

WGSessionArchive

Purpose: Archive the open workgroup session file.

Description: Allow the workgroup session leader to archive a workgroup session in compressed format to a designated storage device.

Uses: Normally used by the workgroup session leader to archive the session file in compressed format to the designated storage device.

Returns: This function returns a dialog box that allows the workgroup session leader to designate the file name, storage device and data compression format. Grabs the handle of the specified session file, compresses the file, then writes the file to a specified device. If the archive is successful, this function returns an integer for the archived file. If an error occurs, it returns a zero.

See Also: *WGSessionRecall*

Record session

WGSessionRecord

Purpose: Record the entire workgroup session.

Description: Allow the workgroup session leader to record a workgroup session in compressed format to a designated high capacity storage device.

Uses: Normally used by the workgroup session leader to record the session file in compressed format to a designated storage device. Recording can be paused or ended using this function.

Returns: This function returns a dialog box that allows the workgroup session leader to designate the file name, storage device and data compression format. Grabs the handle of the specified session file and records the entire workgroup session in compressed format to a specified device. If the recording is successful, this function returns an integer for the recorded file. If an error occurs, it returns a zero.

See Also: *WGSessionPlayback*

Playback session

WGSessionPlayback

Purpose: Playback the entire workgroup session.

Description: Allow the workgroup session leader to playback a workgroup session from a high capacity storage device.

Uses: Normally used by the workgroup session leader to playback all or a portion of the session file. Playback can be paused or ended using this function.

Returns: This function returns a dialog box that allows the workgroup session leader to designate the file name and storage device for playback. Grabs the handle of the specified session file and plays back the entire workgroup session. If the playback access is successful, this function returns an integer and handle for the recorded file. If an error occurs, it returns a zero.

See Also: *WGSessionRecord*

Import object

OLECreateFrom (Schildt et al., 1994b, pp. 394-395)

Export object

OLECopyToClipboard (Schildt et al., 1994b, p. 394)

Define private space

CreatePrivateWorkspace

Purpose: Setup or change the private session workspace.

Description: Using a dialog box, the workgroup user can setup and change a private workspace view. The function provides a dialog box that allows selection of a single window or multiple windows. A drawing or sheet size and 2D or 3D views can be specified. User can add or delete document sections or workspace layers.

Uses: Used to configure a user's private workspace to accommodate a particular function that must be performed off-line including writing, drawing, management tasks, personal tasks, or resource management. This function sends all necessary messages to create the windows, layers, and views specified by the user.

Returns: This function establishes pointers to parameters specifying the workspace class and attributes. It then sends messages to activate the selected windows and attributes. The function returns the handle to the newly created private workspace. It returns a Null on error.

See Also: *MoveToPrivateView*

DISPLAY/ARRANGE VIEWS

Clear group surface

ClearWGSurface

Purpose: Clear the workgroup surface.

Description: Allow the workgroup session leader to clear all material from the shared session workspace. Prompt the session leader to save the session file before the workspace is cleared.

Uses: Used to clear the workspace of existing information to allow a new set of tasks to be started without opening a new workgroup session.

Returns: This function returns a Boolean True if the workspace is cleared. It returns a Boolean False on error.

See Also: none

Access private view

MoveToPrivateView

Purpose: Move to the private session workspace.

Description: Using a dialog box, the workgroup user can move to the private workspace view.

Uses: Used to call up a private view set up by an individual workgroup user.

Returns: This function sends messages to activate the selected private workspace windows and attributes. The function returns the handle of the private workspace. It returns a Null on error.

See Also: *CreatePrivateWorkspace*, *MoveToEmptyView*

Move to empty

MoveToEmptyView

Purpose: Create a semi-private workspace in an empty section of the workgroup workspace.

Description: Using a pop-up menu, the workgroup user can move to an empty section of the workgroup workspace and establish a semi-private work area within the shared workspace. At the user's discretion, this section can be used to perform tasks with or without inputs from other workgroup users.

Uses: Used to move to an empty section of the workgroup workspace.

Returns: This function establishes a pointer to the empty workspace section and establishes a window displaying this workspace section. It returns a Null on error.

See Also: *CreatePrivateWorkspace*, *MoveToPrivateView*

Workgroup view

MoveToWGView

Purpose: Move to the shared workspace view.

Description: Using a pop-up menu, the workgroup user can move from the semi-private workspace view to the shared workspace view.

Uses: Used to return to the shared workspace.

Returns: This function returns a pop-up window that allows the user to deactivate the selected private workspace window and attributes. The function returns the handle of the shared workspace window. It returns a Null on error.

See Also: *MoveToEmptyView*

Show virtual position

ShowVirtualPosition

Purpose: Show workgroup users' virtual positions.

Description: Allow a user to see the virtual position of all users in the shared workspace with respect to shared workspace objects and other users.

Uses: Primarily used when working in a 3-D coordinate system.

Returns: This function returns integers, the locations of the workspace users, and a pop-up window that shows the users' positions in relative to workspace objects and one another.

See Also: None

COMMUNICATION MODE

OpenComm (Conger, 1992, p. 815)

CloseComm (Conger, 1992, p. 811)

Phone support

MakePhoneCall

Purpose: Make a phone call.

Description: Allow a user to initiate a phone call while in a private or workgroup workspace. While the phone call is in process, the function automatically mutes the user's video and audio output to the workgroup session. Conference calls between workgroup members can be arranged and all normal phone functions are available including redial, mute, and hold.

Uses: Used to make phone calls or conference calls during a private or workgroup session. Opens the appropriate phone port and accesses the user's address book.

Returns: This function results in a dialog box that allows the phone call to be setup and initiated. Calls functions to mute audio and video output. If successful, this function returns an integer, the value of the opened communication device. If there is an error opening the phone device, the function returns a negative value.

See Also:

Conference

OpenWGConference

Purpose: Plan a meeting or conference session and send an invitation.

Description: Workgroup member plans a unique synchronous conference or meeting session, establishes registration criteria, and sends an invitation. Session can be new or within an existing workgroup session. Allow the user to set access levels for participants and input user IDs for invited guests. At the discretion of the lead user, the conference can be open or closed. The lead user specifies the date and time, conference topic, and location. The lead user also assigns conference roles including facilitator, organizer, chairperson, attendee, and observer. The function automatically invokes the *DefineAccessList*, *ChangeAccessList*, *AssignWGRoles*, *ChangeWGRoles*, *SendWGInvitation* and *RespondWGInvitation* functions.

Uses: Used by workgroup session users to set up specific conferences and meetings between workgroup session participants and other invitees.

Returns: This function returns a dialog box that allow the workgroup user to setup the conference or meeting, specify access, and send an invitation. If valid, this function returns a handle to the conference or meeting session. It returns a Null on error.

See Also: *DefineAccessList*, *DefineFloorControl*, *ChangeAccessList*, *AssignWGRoles*, *ChangeWGRoles*, *SendWGInvitation*, and *RespondWGInvitation*

Floor control

DefineFloorControl

Purpose: Select floor control mode.

Description: Allow the workgroup or conference session leader to define the floor control method. Several options are available. The system can allow only one user to speak at a time and floor time can be set. A speaker can pass the floor to another group member and floor time can be set. Another user can request permission to control the floor and floor time can be set or there can be no floor control that allows more than one user to have simultaneous access to the floor. The no floor mode can be used for brainstorming and arguing.

Uses: Used to moderate a conference or meeting depending on the desired output and activity.

Returns: This function returns a dialog box that allows selection of the conference floor control mode. This function returns an integer of the selected floor control mode. If an error occurs, it returns a zero.

See Also: None

OTHER SUPPORT

Encryption

WGEncryptionOn

Purpose: Initialize file encryption.

Description: Implement a file encryption algorithm for a designated file or workgroup session.

Uses: Used to encrypt a file or workgroup session file.

Returns: This function returns a dialog box allowing users to select the file or session that should be encrypted. Returns a Null on error. Returns a handle to the enrypted file.

See Also: *WGEncryptionOff*

Purpose: Decrypt a file or workgroup session.

Description: Implement a file decryption algorithm for a designated file or workgroup session.

Uses: Used to decrypt a file or workgroup session file.

Returns: This function returns a dialog box allowing users to select the file or session that should be decrypted. Returns a Null on error. Returns a handle to the decrypted file.

See Also: *WGEncryptionOn*

Data compression

WGCompressionOn

Purpose: Initialize file compression.

Description: Implement a file compression algorithm for a designated file or workgroup session.

Uses: Used to compress a file or workgroup session file.

Returns: This function returns a dialog box allowing users to select the file or session that should be compressed. Returns a Null on error. Returns a handle to the compressed file.

See Also: *WGEncryptionOff*

WGCompressionOff

Purpose: Decompress a file or workgroup session.

Description: Implement a file decompression algorithm for a designated file or workgroup session.

Uses: Used to decompress a file or workgroup session file.

Returns: This function returns a dialog box allowing users to select the file or session that should be decompressed. Returns a Null on error. Returns a handle to the decompressed file.

See Also: *WGCompressionOn*

In/out board

AccessI/OBoard

Purpose: Access the In/Out board.

Description: Allow users to access an In/Out board to change their work status or check the work status of another workgroup member.

Uses: Used to allow users provide their work status to other group members.

Returns: This function returns a dialog box allowing users to change their In/Out status or check status of another workgroup member. Returns a Null on error.

See Also: None

Digital signature

AddDigitalSignature

Purpose: Add a digital signature to a document.

Description: Allows a workgroup user to request a digital signature on a document or form. A user can also place a digital signature in a document file.

Uses: Used to sign a document or form or request a signature from another user.

Returns: This function returns a dialog box allowing users to add a digital signature to a document or request a signature from another workgroup member. Returns a Null on error. Transfers the public key for the authentication system and returns a pointer to the private key.

See Also: None

Workgroup Support APIs

DECISION TOOLS

Select decision style

DefineDecisionStyle

Purpose: Select workgroup session leader's decision style.

Description: Allow the workgroup leader to select a decision style for arriving at workgroup decisions. Multiple options exist to make decisions by collecting data, sharing the problem with individuals or the group, and by generating and evaluating alternatives. Style allows consensus to occur and polling and voting can take place.

Uses: Used by the workgroup session leader to select a preferred decision style.

Returns: This function returns a dialog box that allows the workgroup leader to specify a decision making style for workgroup activities. This function returns an integer of the selected decision making style. If an error occurs, it returns a zero.

See Also: *DefineDecisionModel*

Select decision support model

DefineDecisionModel

Purpose: Select a decision support model.

Description: Allow the workgroup leader to select a decision support model for arriving at workgroup decisions. Decision support models include none, analysis, brainstorming, modeling, and simulation.

Uses: Used by the workgroup session leader to select a preferred decision support model to support workgroup decision making.

Returns: This function returns a dialog box that allows the workgroup leader to specify a decision support model for workgroup activities. This function returns an integer of the selected decision support model. If an error occurs, it returns a zero.

See Also: *DefineDecisionStyle*

Freeze contributions

WGContributionFreeze

Purpose: Freeze group contributions.

Description: Allow the workgroup session leader to freeze group inputs at a selected time or decision point.

Uses: Normally used by the workgroup session leader to stop workgroup contributions based on time or decision point. This function can be toggled on and off.

Returns: No returned value.

See Also: *WGPoll*, *WGConsensus*, and *WGVote*

Poll the group

WGPoll

Purpose: Poll the group.

Description: Allow the workgroup session leader to poll a group's position on an issue or idea at a selected time or decision point.

Uses: Normally used by the workgroup session leader to poll the workgroup at any time or at a decision point. This function can be toggled on and off. Polling results can be displayed to all users at the discretion of the workgroup session leader.

Returns: This function returns a pop-up window that prompts a poll on a specific issue. Returns another pop-up window on the lead user's workstation displaying poll results. The function returns a Null on error.

See Also: *WGContributionFreeze*, *WGConsensus*, and *WGVote*

Obtain consensus

WGConsensus

Purpose: Achieve group consensus.

Description: Allow the workgroup session leader to request group consensus on an issue or idea at a selected time or decision point.

Uses: Normally used by the workgroup session leader to get workgroup consensus on an idea or issue at any time or decision point. This function can be toggled on and off. Consensus results can be displayed to all users at the discretion of the workgroup session leader.

Returns: This function returns a pop-up window that prompts consensus on a specific issues. It returns another pop-up window on the lead user's workstation that displays consensus results. The function returns a Null on error.

See Also: *WGContributionFreeze*, *WGPoll*, and *WGVote*

Obtain vote

WGVote

Purpose: Take a group vote.

Description: Allow the workgroup session leader to prompt the group to vote on an issue or idea at a selected time or decision point.

Uses: Normally used by the workgroup session leader take a workgroup vote on a specific issue. This function can be toggled on and off. Voting results can be displayed to all users at the discretion of the vote-taker.

Returns: This function returns a pop-up window that prompts a vote on a specific issues. Returns another pop-up window on the lead user's workstation that displays vote results. The function returns a Null on error.

See Also: *WGContributionFreeze*, *WGConsensus*, and *WGPoll*

Assign tasks

AssignWGTask

Purpose: Assign tasks to specific workgroup members.

Description: Session leader can assign and transmit specific tasks to one or more workgroup members. Tasks are defined, responsibility assigned, and deadlines provided.

Uses: Used to send workgroup tasks to workgroup members.

Returns: This function returns an integer for the assigned task and a handle for the message transmitting the task to the workgroup members. It returns a zero on error.

See Also: None

Make proposal

SendProposal

Purpose: Make a proposal to specific workgroup members.

Description: Workgroup session members can make proposals and transmit them to one or more workgroup members. Tasks are defined, responsibility assigned, and deadlines provided.

Uses: Used to send workgroup tasks to workgroup members.

Returns: This function returns an integer for the proposal and a handle for the message transmitting the proposal to the other workgroup members. The function returns a zero on error.

See Also: None

Develop decision rules

DefineDecisionRules

Purpose: Develop workgroup decision rules.

Description: Allow the workgroup leader and selected workgroup members to interactively develop declarative rules to support group decision making or information sorting.

Uses: Used by the workgroup session leader to develop declarative rules.

Returns: This function returns a pop-up window that allows users to write declarative decision rules for workgroup activities. This function returns an integer of the selected decision making rule. If an error occurs, it returns a zero.

See Also: *DefineDecisionModel*, *DefineDecisionStyle*

DRAWING/GRAPHICS

Drawing object status

WGObjectStatus

Purpose: Get status on a workspace object or request a change to an object's status.

Description: When initiated by a user, the system sends a workgroup object description to the workgroup member including object coupling status, object owner ID, and object acquisition status. The user can then request the object owner to change coupling status. The object owner or the system can change coupling status and send change to users. A user can also request the object to change its owner and send the change to other users.

Uses: Normally used to request the status of a workgroup object or request a change to an object's status.

Returns: This function returns a handle of the workgroup object, an object status message, and a request to change an object's status. Establishes a pointer to the new workgroup object owner. If successful, this function returns a integer for the object. It returns a Null on error.

See Also:

Get object

RequestGrab

Purpose: Request permission from an object to grab it.

Description: Requests the object for permission to grab it. It allows an object in shared workspace to grant or deny permission to be grabbed depending on object ownership and access designation. Results in a *DoGrab* or *EndGrab* function depending on access approval or denial.

Uses: Grants a workgroup user permission to access a particular object in shared workspace in order to move it or perform some other function.

Returns: If successful, this function returns a Boolean True. If unsuccessful, it returns a Boolean False.

See Also: *DoGrab*, *EndGrab*

DoGrab

Purpose: Get the object.

Description: Grabs the object depending on access approval.

Uses: Grants a workgroup user permission to access a particular object in shared workspace in order to move it or perform some other function.

Returns: If successful, this function returns a Boolean True. If unsuccessful, it returns a Boolean False.

See Also: *RequestGrab*, *EndGrab*

EndGrab

Purpose: Release the object.

Description: Release the grabbed object depending on object ownership and access designation. Allows another user to request or grab the object.

Uses: Used to release an object in shared workspace to allow other users access to it.

Returns: If successful, this function returns a Boolean True. If unsuccessful, it returns a Boolean False.

See Also: *DoGrab*, *RequestGrab*

Owned object list

ListOwnedObjects

Purpose: List workgroup objects and owners.

Description: Allow a user to determine who owns each workgroup object. This is similar to the *WGObjectStatus* function; however, the function does not allow an object's status to be changed.

Uses: Used to obtain workgroup object status and owner information and display it in a status window.

Returns: This function returns a pop-up status window that displays the requested workgroup object and owner status information. It returns a Null on error.

See Also: *WGObjectStatus*

TEXT

Type shared text

AddWGText

Purpose: Add a text string to a workgroup file.

Description: Allow a user to add a text string to the workgroup session file. This function is used to write short passages. It is not a writing tool for long documents.

Uses: Used to annotate objects and documents in the workspace session file.

Returns: Writes a text string to the specified window area. Returns a pointer to the string output and specifies how many characters are in the string. If successful, this function returns a Boolean True. If unsuccessful, it returns a Boolean False.

See Also: *GrabWGText, ChangeWGText*

Select shared text

GrabWGText

Purpose: Select a shared text string.

Description: Allow a user to grab a text string in a workgroup session file. This function allows a user to get text to move or change it.

Uses: Grants a workgroup user permission to access a particular text string in shared workspace in order to move it or perform some other function.

Returns: If permission is granted the function returns Boolean True. If permission is denied, it returns Boolean False.

See Also: *AddWGText, ChangeWGText*

Change shared text

ChangeWGText

Purpose: Change shared text string.

Description: Allow a user to change a text string in a workgroup session file. This function allows a user to move text or change it.

Uses: Grants a workgroup user permission to change a particular text string in shared workspace.

Returns: If successful, this function returns a Boolean True. If unsuccessful, it returns a Boolean False.

See Also: *GrabWGText, ChangeWGText*

WRITING/EDITING

Activities

DefineWGWritingActivity

Purpose: Define workgroup writing activity sequence.

Description: Allows the lead writer to sequence or segment the writing activity into brainstorming, research of ideas and issues, drafting an outline, writing a document, or use of an open style without boundaries.

Uses: Normally used by the lead writer to structure writing tasks into discrete segments and sequences. Can also be used to define an open writing style that allows any type of input.

Returns: This function returns a dialog box that allows mode selection and an integer, the selected writing activity. It returns a Null on error.

See Also: *DefineWGWritingStrategy*

Strategies

DefineWGWritingStrategy

Purpose: Define workgroup writing strategy.

Description: Allow the lead writer to define how the writing will take place. Options include a single writer who writes most of a document while other writers assist, a scribe who writes down most of a group's thoughts and comments while other writers discuss the ideas that will be expressed in a document, separate writers where document is broken down into sections that are authored separately, and joint writing where several group members compose the text together.

Uses: Normally used by the lead writer to define how a document will be written.

Returns: This function returns a dialog box that allows mode selection and an integer, the selected strategy. It returns a Null on error.

See Also: *DefineWGWritingActivity*, *DefineWGDocumentControl*

Document control

DefineWGDocumentControl

Purpose: Set up how the document can be accessed depending on the workgroup writing strategy.

Description: Determine how a document will be controlled depending on the selected writing strategy. Control can be centralized with only one writer, relayed between users, independent with document sections accessed by only one writer at a time, and shared with simultaneous access.

Uses: Used to control access to a document or its sections based on the selected writing strategy.

Returns: This function returns a dialog box that allows mode selection and an integer, the document control strategy. It returns a Null on error.

See Also: *DefineWGWritingActivity*

Permissions

DefineWGDocumentPermissions

Purpose: Set document access permissions.

Description: Allow the workgroup session leader or lead writer to designate document access levels by user ID. Access can be read/write, comment, or read only. Access permission can be changed.

Uses: Used to control access to a document or its sections based on user ID.

Returns: This function returns an integer, permission, for each affected user ID. During user logon, if a match is successful, this function returns Boolean True and appropriate access is granted. It returns Boolean False if a match does not occur for a user ID and no access is allowed.

See Also: *DefineWGWritingActivity*

Writing rules

DefineWGWritingRules

Purpose: Define workgroup writing rules.

Description: The lead writer define workgroup style and content rules. Either the lead writer, editor, or both define workgroup editing rules.

Uses: Normally used to define style content and editing rules.

Returns: This function returns an integer for each style, content, or editing rule for each affected document.

See Also: *None*

Revision control

DefineWGDocumentRevision

Purpose: Control how document revisions are made.

Description: Describe the types of document revisions and how they are handled by the system. Examples are a small document change where the change is made without marking it, many additions where a change is marked so that it stands out from the other text and the responsible writer makes the changes, and text marked for deletion where the responsible writer approves the deletion. Document change history is also displayed.

Uses: Used to control who approves and makes document changes.

Returns: Below a certain specified error limit the document is automatically revised by the system. For more extensive changes, the system marks the document, matches the user ID for the portion of the document written by the user with the user ID of the document changer. If a match occurs, then the document is changed. If a match does not occur the responsible writer is notified and prompted to approve the change.

See Also: None

Version control

DefineWGVersionControl

Purpose: Allow lead writer to select how many document versions are allowed.

Description: Allows the lead writer to define single version or multiple version documents and arbitrate writing conflicts with the lead author or all authors.

Uses: Used to define allowable document versions and resolve writing conflicts.

Returns: This function returns a dialog box allowing the lead author to select either single or multiple document versions. It returns a Null on error.

See Also: None

Post-meeting documents

CreateWGMeetingDocument

Purpose: Prepare post-meeting documents based on the meeting session.

Description: Allow the meeting/conference chairperson to select an output for the specified meeting. Selections include an electronic format meeting/conference transcript, complete hardcopy transcript, a meeting/conference snapshot report, a summary report, or a highlights report.

Uses: Used by the meeting/conference chairperson to document meeting and results.

Returns: This function returns a dialog box that allows the meeting/conference chairperson to select a format for the meeting/conference output. It returns a Null on error.

See Also: None

Video editing

EditWGVideo

Purpose: Edit workgroup video.

Description: Allow the workgroup session leader or designated users to interactively edit video clips or segments.

Uses: Used to edit workgroup video segments.

Returns: This function returns a dialog box that allows the workgroup session leader or designated users to select a video file and video editing tools. Returns a handle to the specified video file. It returns a Null on error.

See Also: None

DATABASES

Access database

OpenWGDatabase

Purpose: Access a shared workgroup database.

Description: Allow access to a shared workgroup database. Users were authorized database access during workgroup setup. Normal database functions are available including adding to, deleting, changing, manipulation, filtering, and sorting. Appropriate permissions are required.

Uses: Used to provide database access to workgroup users.

Returns: This function returns a dialog box allowing designated users to access a particular workgroup database. It returns a Null on error.

See Also: None

MAIL

Mail link

MAPILogon (Sheldon, 1994, p. 169)

MAPILogoff (Sheldon, 1994, p. 169)

Group address book

OpenAddressBook (Sheldon, 1994, p. 169)

CALENDAR/SCHEDULE

Calendar

OpenWGCalendar

Purpose: Open a workgroup calendar.

Description: Users can open a shared workgroup calendar, make additions, or change the calendar. The calendar function allows the user to organize appointments by type, date, or other attribute, set alarms, place a reminder notice on the calendar, or add a "to do" item and completion date to the calendar. Auto-date programming allows a recurring date to be programmed for a specified length of time.

Uses: Used to organize and track a user's calendar.

Returns: This function returns a dialog box allowing all users to access a calendar function from either inside or outside of the workgroup. Returns a handle to the specified calendar file. It returns a Null on error.

See Also: *CloseWGCalendar, OpenWGSchedule, CloseWGSchedule*

CloseWGCalendar

Purpose: Close the workgroup calendar.

Description: Remove a user's workgroup calendar from the existing private or shared workspace.

Uses: Close the calendar.

Returns: If successful, this function returns a Boolean True. If unsuccessful, it returns a Boolean False.

See Also: *OpenWGCalendar, OpenWGSchedule, CloseWGSchedule*

Scheduler

OpenWGSchedule

Purpose: Open a workgroup schedule.

Description: Users can open a shared workgroup schedule, make additions, or change the schedule.

Uses: Used to develop and track time and resource schedule items.

Returns: This function returns a dialog box allowing all users to access the workgroup schedule function from either in or out of the workgroup. Returns a handle to the specified schedule file. It returns a Null on error.

See Also: *CloseWGSchedule, OpenWGCalendar, CloseWGCalendar*

CloseWGSchedule

Purpose: Close the workgroup schedule.

Description: Remove a user's workgroup schedule file from the existing private or shared workspace.

Uses: Close the schedule function.

Returns: If successful, this function returns a Boolean True. If unsuccessful, it returns a Boolean False.

See Also: *OpenWGSchedule, OpenWGCalendar, CloseWGCalendar*

Proxy

DefineProxy

Purpose: Choose a proxy.

Description: Select a proxy from an available list and provide the proxy with appropriate access to include read only, read/write, or accept proposals. If selected, the proxy can view or change a user's file. The user can view the proxy's calendar.

Uses: Used to choose a proxy for a function, event, or meeting.

Returns: This function returns a dialog box allowing users to view a proxy's schedule and select a proxy from a predefined list or designation of a user ID. Returns an integer for the selected proxy. If unsuccessful, it returns a zero.

See Also: None

IDEAS

Idea generation

CreateIdeas

Purpose: Generate ideas.

Description: Develop ideas using brainstorming, topic commenting, or a group outliner.

Uses: Normally used during meetings to generate ideas and alternatives.

Returns: This function returns a dialog box allowing users to select the idea generation method. Returns an integer for the selected method. If unsuccessful, it returns a zero.

See Also: *OrganizeIdeas*

Idea organization

OrganizeIdeas

Purpose: Organize and prioritize ideas.

Description: Allow a workgroup session leader to organize and prioritize ideas using an idea organizer, issue analyzer to achieve group consensus, or have group members write to organize ideas.

Uses: Used to organize and prioritize ideas developed during a meeting or decision making session.

Returns: This function returns a dialog box allowing users to select the idea organization and prioritization method. Returns an integer for the selected method. If unsuccessful, it returns a zero.

See Also: *CreateIdeas*

WORKFLOW

Define workflow

DefineWorkflow

Purpose: Define the workflow process and routing.

Description: Users define the workflow process, tasks, and routing.

Uses: Used by designated users to develop a workflow process, tasks, and routing.

Returns: This function returns a dialog box allowing users to select a predetermined workflow, workflow tasks, workflow routing, or develop new or revised workflows, tasks, and routings. Returns an integer for the selected workflow, task, or routing and a handle to the designated workflow file. If unsuccessful, it returns a zero. The function also returns a pop-up window that allows development of new or revised workflow, tasks, or routing.

See Also: *DefineWorkflowCriteria*

Define criteria

DefineWorkflowCriteria

Purpose: Define criteria that are used to manage the workflow.

Description: Select or develop procedures that state what must occur before work can be routed. Allows the user to state what action or activity must happen before work can continue through the workflow.

Uses: Used to define criteria that allows work to move through the workflow routing.

Returns: This function returns a dialog box that allows users to select or develop workflow criteria. Returns an integer for the selected criteria. If unsuccessful, it returns a zero. Returns a pop-up window that allows development of new or revised workflow criteria.

See Also: *DefineWorkflow*

Assign action

AssignWorkflowAction

Purpose: Assign workflow actions to specific workgroup members.

Description: A workfow user can assign and transmit specific workflow actions to one or more workgroup members. Actions are defined, responsibility assigned, and deadlines assigned.

Uses: Used to send workflow actions to workgroup members.

Returns: This function returns an integer of the assigned action and a handle for the message transmitting the action to the workgroup member. The function returns a zero on error.

See Also: None

Appendix E

Correspondence

11505 Paseo Del Oso N.E.
Albuquerque, NM 87111
May 3, 1994

Director of Publications
Association for Computing Machinery
P. O. Box 12114
Church Street Station
New York, NY 10257

Dear Sirs:

I am a doctoral candidate at Nova Southeastern University, Ft. Lauderdale FL. I am writing my dissertation on A Taxonomy of Workgroup Computing. I would like permission to adapt and use figures 1 and 2 on page 41 of "Groupware: Some Issues and Experiences", Communications of the ACM 34(1) by C. A. Ellis, S. J. Gibbs, & G. L. Rein, 1991. New York: ACM Press.

The dissertation citation will read as follows:
Note. Adapted with permission from . . .and "Groupware: Some Issues and Experiences", Communications of the ACM 34(1) (p. 41) by C. A. Ellis, S. J. Gibbs, & G. L. Rein, 1991. New York: ACM Press. Copyright 1991 by the Association for Computing Machinery, Inc.

Sincerely,

Warren Von Worley

Office of Publications

May 13, 1994

Mr. Warren Von Worley
11505 Pasco Del Oso N.E.
Albuquerque, NM 87111

Dear Mr. Von Worley:

ACM has no objection to your including Figures 1 and 2 from "Groupware: Some Issues and Experiences," by Ellis et al published in the January 1991 issue of Communications of the ACM in your doctoral dissertation on A Taxonomy of Workgroup Computing. Your acknowledgment of the source is fine.

Sincerely,

Jane Carlton

Jane Carlton
Copyrights and Permissions

ASSOCIATION FOR COMPUTING MACHINERY

1515 Broadway, New York, New York 10036-5701 USA ◆ Telephone 1-212-869-7440 ◆ Facsimile 1-212-869-0481

11505 Paseo Del Oso N.E.
Albuquerque, NM 87111
May 3, 1994

Prentice Hall, Inc.
Englewood Cliffs, NJ 07632

Dear Sirs:

I am a doctoral candidate at Nova Southeastern University, Ft. Lauderdale FL. I am writing my dissertation on A Taxonomy of Workgroup Computing. I would like permission to adapt and use figure 13.9 on page 448 of Human-Computer Interaction by A. Dix, J. Finlay, G. Abowd, & R. Beale, 1993. New York: Prentice-Hall.

The dissertation citation will read as follows:
Note. Adapted with permission from Human-Computer Interaction (p. 448) by A. Dix, J. Finlay, G. Abowd, & R. Beale, 1993. New York: Prentice-Hall. Copyright 1993 by Prentice Hall International (UK) Limited .

Sincerely,

Warren Von Worley

Paramount Publishing

PRENTICE-HALL, INC.
Copyrights & Permissions
Englewood Cliffs, NJ 07632

June 24, 1994

Warren Von Worley
11505 Paseo Del Oso N.E.
Albuquerque, NM 87111

Dear Mr. Von Worley:

In accordance with the conditions outlined in your letter of 5/3/94, we are very glad to give you permission to quote from our book(s) **HUMAN-COMPUTER INTERACTION** by dix/Finley/Abowd/Beale

Conditions: **Permission is for your doctoral dissertation only.**

Please give credit to the author(s), the title(s), and the publisher with copyright year date(s). To this add, reproduced by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

Sincerely,

Emily McGee

Copyrights & Permissions

Biographical Sketch of the Author

Mr. Von Worley was born in Paterson, New Jersey on February 11, 1945. He holds a B.S. in Chemical Engineering from the New Jersey Institute of Technology and a M.S. in Systems Management from the University of Southern California.

Mr. Von Worley has 27 years of professional experience in management and system engineering of computer facilities and equipment including computer hardware and software integration; computer facilities design, installation, activation, and operations; communications and networking; and development of data and signal processors and sensor systems.

Mr. Von Worley is a Senior Engineer with W. J. Schafer Associates, Inc. (WJSA), Albuquerque, NM, where he is program manager and chief system engineer for a space systems technology assessment contract with the Air Force Phillips Laboratory at Kirtland Air Force Base. Recent accomplishments include a study of the application of advanced data and signal processors to space systems and initiation of a laboratory technical multimedia presentation project. He was also technical director on an environmental assessment expert system research project. His areas of professional interest, including workgroup computing, networking, multimedia, expert systems, and human-computer interaction, haved help WJSA diversify into several information systems technology areas.

Mr. Von Worley is Vice President of InfoConcepts, Inc., a New Mexico corporation, that develops and markets specialty database products. InfoConcept's first database application, a Civil War Regimental Information System, was commercially available in October 1994. In addition, Mr. Von Worley is on the faculty of the University of Phoenix.

This dissertation was submitted by Warren Von Worley under the direction of the Chairperson of the Dissertation Committee listed below. It was submitted to the School of Computer and Information Sciences and approved in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Information Systems and Sciences with specialization in Information Systems at Nova Southeastern University.

Approved:

1/16/95
Date

Harvey M. Deitel
Harvey M. Deitel, Ph.D.
Chairperson

Laurie P. Dringus
Laurie P. Dringus, Ph.D.
Committee Member

Marlyn Kemper Littman
Marlyn Kemper Littman, Ph.D.
Committee Member

George K. Fornshell
George K. Fornshell, Ph.D.
Program Director

3/3/95
Final Approval Date

Edward Lieblein, Ph.D.
Dean
School of Computer and
Information Sciences