Classifying Groupware

Jonathan D. Fouss, Kai H. Chang
Department of Computer Science and Software Engineering
Auburn University
Auburn, AL 36849-5347
{foussjd,kchang}@eng.auburn.edu

Abstract - The definition of what groupware is can be a topic of great debate and is often very broad. This allows many types of software to earn the name groupware but makes it very difficult to compare applications and to do any kind of background research in the field. This paper attempts to bring some organization to the chaos and serve as introduction to groupware.

1. Introduction

Since the beginning of the human race, man has had tasks to complete and has sought ways to make those tasks easier to accomplish. This led to the forming of groups to lessen the load of an individual and the continual development of tools to aid in the work. In that regard, little has changed in the modern workplace. It is quite common for groups of individuals to work together, or collaborate, to complete some project. The tool of choice is the computer, which is ceaselessly being given more power and capabilities.

The computer's purpose is the same as any tool: to help users complete tasks more quickly and more easily. This may involve mathematically simulating or modeling some phenomena or simply allowing a user to type a paper. Regardless of the size of the task, computers have traditionally focused on increasing an individual's productivity and have somewhat ignored the fact that many projects are worked on by groups. The development of local area networks has allowed users to more readily share information and offered new ways of communication. This has been extended further with extraordinary growth of the Internet. Yet, the computer is still largely seen and used as a tool to help an individual complete work.

2. CSCW

Computer Supported Cooperative Work (*CSCW*) is a relatively new area of research that deals with issues involved in group work and how technology can aid the process. Irene Grief and Paul Cashman coined the phrase "computer supported cooperative work" in 1984 [1, 2]. There has been some debate that this term is an inappropriate title for the research field. The typical complaint is that the phrase does not adequately define the field, and the words themselves are too restrictive [1, 3, 4]. *Computer* is typically used to mean all technology and not just specifically computers. *Supported* is essentially anything technology does to help the group. *Cooperative* is used to refer to all group interactions such as competition, conflict, and cooperation. *Work* typically refers to the task or product itself but normally includes social interaction and communication. Strict adherence to the actual meaning of the phrase does not encompass the whole area of research that has come under the umbrella term CSCW [4]. The term has now come to represent a multidisciplinary field that focuses on two areas of study: one, the human aspect of groups (cooperative work) and two, the technological aspect of groups (computer support) where a group is a set of individuals attempting to accomplish a task. The various disciplines involved in this area of research include computer science, sociology, psychology, and management science to name just a few. [2, 3, 4, 5]

Another common term in the field, *groupware*, has had similar problems. There has been much discussion, though less than that about CSCW, about what exactly is groupware and how to classify it [4, 6, 7, 8]. A generally accepted definition would be any computer application (software or hardware) that in some way supports group activities [4, 8]. The drawback to this definition is that it is very broad and encompasses many different types of applications. For example, a tool that allows individuals to simultaneously contribute to a single written document would be considered groupware but so would a network operating system and an e-mail system.

Due to the broad definition often given to groupware, it is nearly impossible to develop a way to contrast and compare various groupware tools. In literature there are generally two approaches to creating a taxonomy for groupware. The first approach defines characteristics that can be used to distinguish between different groupware applications. The second approach gives some common categories of groupware applications.

3. Groupware Characteristics

The term groupware describes an extraordinarily large class of software applications. These applications vary greatly in scope and purpose but are all linked because they in some form or fashion facilitate group work. To help bring some organization to the chaos and give some means of

comparison, characteristics of groupware have been developed.

By far the most common characteristic found in literature describes the time interaction between group members. This may refer to how the software handles communication between the members, how the members work on the project, or both. An application that is *asynchronous* allows a single user to work on the project at a time. When a user completes his work, the project is passed on to the next member. In a *synchronous* application multiple users are allowed to work on the project simultaneously. A synchronous system would be real-time. [2, 3, 5, 8, 9]

Location of group members in relation to each other is another characteristic used to classify groupware applications. Location is typically specified to be same place or different place. *Same place* means the group members are actually in the same location. For example, all group work may be done in a conference room and the application may be designed to help out during the meetings. *Different place* simply means that face to face communication is unavailable. This could mean team are in vastly different geographic location or simply not in the same room with each other. [2, 3, 5, 8, 9]

The characteristics describing the timing and location of groupware are often used to create a two by two grid used to classify groupware [2, 5, 8, 9]. Figure 1 shows an example of this and some examples of what fits into each category. It is important to note that many groupware applications may actually overlap and fit into multiple locations in this grid.

	Same Place	Different Place
Synchronous	traditional face to face meeting	video conferencing
Asynchronous	assembly line	e-mail system

Figure 1

Groupware can also be characterized by looking at the degree to which the technology supports group members. Groupware applications can be classified as f-, k-, or w- groupware. In an f-groupware system, there is a single workstation controlled by the person in charge of organizing the meeting, the *f*acilitator. The facilitator uses traditional methods to run the meeting and gather information from the other group members and then inputs the results into the computer. The idea is to use a software application to interpret and organize the group's brainstorming. K-groupware systems also call for a single coordinating workstation, but each group member has some way to input information directly into the system such as a *k*eypad. The software application would likely be some sort of voting system. A w-groupware system allows each group member to have a *w*orkstation. It perhaps allows the greatest flexibility of the three types and allows for greater efficiency in capturing data. [9]

Another characteristic used to classify groupware is the method the system uses to coordinate user activity. Groupware coordination can be sequential, parallel, or reciprocal. *Sequential* indicates that only one member is actively working on the project at any time. When one member completes his work on the project, it is passed on to the next. In *parallel* coordination, each team member is working independently on a part of the project, but the work is being done simultaneously. And *reciprocal* coordination means the multiple group members are working together on a part of the project. A groupware system may support a single type of coordination or some combination. [10, 11, 12]

Another characteristic, *closeness of collaboration* [10], is very similar to the coordination characteristic. The closeness characteristic attempts classify groupware by defining a spectrum to measure how closely together team members work. On one end of the spectrum is *division of labor.* This indicates that individual group members work independently on their assigned part. Collaboration takes place when the individual parts are combined. The other end of the spectrum is *shared mind.* Shared mind means the project is not divided for individual tasks but rather the group works in unison to complete the project.

Common task dimension forms another spectrum that can help to classify groupware. *Common task dimension* is a measure of how tightly coupled individuals are in completing work. The low end of the spectrum would be something like users on a mainframe time sharing systems. Each user is working independently on different tasks but is still a group in a sense because they are all working on the same machine. In this case the users are loosely coupled. Some sort of software review system that allowed multiple users to simultaneously review a piece of code would represent a tightly coupled system on the other end of the spectrum. [8]

A *shared environment dimension* spectrum is an indication of the level of information the groupware systems provides about other users and the environment. The low end of this spectrum would be a basic e-mail system since it provides very little information about the environment. An electronic classroom system would be an example of the high end of this spectrum. To simulate the experience of classroom electronically, the system would need to

somehow provide many environmental cues about the users and the state of the class. [8]

The type of output supported by the groupware system is another method of classification [9]. This does not refer to physical results generated by the system such as a document but rather what awareness each group member has after using the system. This classification system can be modeled as a three-tiered pyramid. At the lowest level, each member is simply aware of the other members' opinions about the project. This is referred to as *share opinions*. The next level is *shared opinions*. This means that the group has come to a consensus about the project's goals and priorities. The top level indicates that everyone in the group has a clear understanding about what must be done to complete the project and the path that will be followed. It is called a *shared mental model.*

There are numerous characteristics, spectrums, and terms that have been proposed to help describe and classify groupware, but no single method seems to be adequate. However, by using a combination of the techniques, it is possible to get a reasonably clear picture of a particular application's capabilities and give some means for comparison. A fairly complete picture can be described by looking at how an application handles three areas:

Communication - how ideas are shared between group members,
Collaboration - how data is shared between group members, and
Coordination - how work is shared between group members. [8]

These areas are easily described using the previously discussed classification methods.

4. Groupware Taxonomy

A common way to classify single-user applications is to describe types of applications based on the functionality they provide. For example, some common single-user application types would be productivity applications, development applications, and entertainment applications. These types can be further divided. Spreadsheets, word processors, and presentation software are examples of subtypes of productivity applications. It is possible to classify groupware in a similar manner. This approach also seems to be more useful than the first. However, the groupware taxonomy tends to have more overlap than in single-user applications and varies greatly in literature. The following groups have been developed from [2], [5], [6], [8], and [13].

Message-based Systems

Message-based systems are the oldest and most primitive type of groupware but are still the most common. The idea is simple: messages can be passed from one user to another. Messages are generally text-based, but any type of data can be included. The primary use of such systems is communications, but many do provide simple mechanisms to share files. The most common example of a message-based system is electronic mail, more commonly referred to as e-mail. [8, 13]

There have been case studies that show e-mail can be a useful tool for collaboration. However, studies have also shown that there are deficiencies in using e-mail collaboration. E-mail does not provide social or cognitive cues about the other group members and is highly subject to misinterpretation. It is ineffective when it comes to negotiation and resolving conflicts, and the success of such collaborations seems to be strongly dependent on the personalities of the group members. There is very limited means to monitor progress of individual work or of the project as a whole. Studies have shown that face-to-face meetings are still very helpful in such situations [2, 11, 12, 13, 14]. Another problem with e-mail is that typically one address receives all of a user's mail and not just the e-mail related to the collaboration. Depending on the user, the collaboration e-mail may be a small part of his regular e-mail which means there is a danger it may be overlooked or even deleted and hinder the collaboration process. This phenomenon is sometimes referred to as "information overload." [8]

Several specialized variants of e-mail have been developed in an attempt to address the groupware limitations of e-mail. Coordinator [3, 6, 8, 10, 11] appears to be one of the most mentioned groupware applications in literature. However, this is not because the product's merit but rather because it seems to be one of the first commercial attempts at groupware. The basic idea is to make e-mail much more structured by adding a few additional fields and incorporating 'speech-act' rules into the e-mail tool. The hope was that the speech-act methodology would help pass along the intent of the sender and allow the whole dialogue to be viewed in a logical manner. In practice however, users tended not to use the system's features and reduced it back to common e-mail.

COSMOS [15] and Information Lens [3, 8] also attempt to enhance e-mail by adding additional fields to the traditional e-mail message. Again, the idea was that requesting users to provide more information, e-mail would be more structured and provide better collaborative environment. One noteworthy feature of COSMOS was that it allowed group members to be assigned specific roles and tasks in the project. However, neither product was widely accepted because users did not see the benefits justifying the extra work and time it took to create an e-mail message.

Some of the ideas of these three applications can be seen in more modern e-mail tools. While the idea of adding compulsory fieldsdid not flourish, the idea of filtering e-mail did. Most contemporary e-mail tools such as Microsoft's Outlook Express [16], Microsoft's web based e-mail tool – Hotmail [17], Eudora [18], and Netscape's Communicator [19] allow users to setup a reasonably

sophisticated method of filtering e-mail. This typically involves creating rules that instruct the e-mail tool to watch for certain keywords in the header of the e-mail message. Upon identifying a message, the rule may instruct the tool to delete the message, forward it to another address, or place it in a specified folder.

Another method of enhancing e-mail is to utilize a scripting language inside the e-mail message. In this way the message is still text, but the tool is programmed with added intelligence to interpret special commands inside the message. Imail [8] and MESSE [13] are examples of systems that have implemented this type of strategy. Newer e-mail tools such as Microsoft's Outlook Express [16], Microsoft's web based e-mail tool – Hotmail [17], Eudora [18], and Netscape's Communicator [19] permit the use of the hypertext markup language (HTML) inside of messages. However, the primary reason for this is to allow formatting in e-mail and not to enhance collaboration abilities.

Filtering, adding compulsory fields, and scripting languages all enhance e-mail and can be found to some degree in most contemporary e-mail systems. They do tend to solve the 'information overload' problem but do not truly enhance the collaborative abilities of e-mail. While certainly useful, e-mail by itself is not enough for teams to consistently collaborate on a project.

E-mail is primarily a person-to-person communication medium. Most tools do provide the capability to send or copy messages to multiple individuals, but this is not very effective for large groups. In order to share information with larger groups and to simplify the maintenance of the list of members, some modified forms of e-mail have been developed. [5]

Mailing list servers and newsgroups are the most known modifications. Both provide the means to broadcast messages to a group but function in slightly different manners. A list server maintains a roster of e-mail addresses. When an e-mail message is sent to the list, it is forwarded to all the e-mail addresses in the roster. The only work required of the user is to find a list server that deals with topics of interest and subscribe to it. Subscribing typically involves sending an e-mail message to the server with a specific message in the body (e.g. subscribe gamers.) [5, 20]

Newsgroups work in a similar manner, but require the user to occasionally check for new messages. When this is done all new messages that have been posted to the group are downloaded to the user's tool. Newgroups also allow the user to see the progression of a conversation by grouping response messages with the original message. [5, 20]

In the commercial world, the most known groupware applications are Lotus Notes [6, 20, 21, 22, 23] and its competitors (Microsoft Exchange [21, 22, 24], Novell Groupwise [21, 22, 25], and Netscape Collabra [22, 26].) The major function of these applications is to be a message, or discussion, server. Essentially, these applications provide the necessary functionality to provide standard e-mail, list servers, and newsgroups to the local area network on which they are installed. They may also be configured to be accessible via the Internet. These applications do have other collaborative features that will be discussed later, but are primarily message-based systems.

Due to the boom of Internet use, several applications based on web technology and standards have been developed and offer services similar to Lotus Notes. The major benefit of these applications is that only a web browser is required on the client machines whereas special client software is required for Lotus Notes and competing products. They also tend to be much cheaper. However, most of the web applications offer only a single feature and require the purchase of several products. For example products like NetThread, RadNet, Allaire Forums, and WebForum support only the newsgroup functionality found in Notes. [27, 28]

Coordination Systems

This type of groupware refers to systems that help group members coordinate their activities [5, 8]. Coordination systems can be thought of from two perspectives. First, there are group calendar and scheduling applications. These applications are designed for use by anyone in the group. In these applications, the user is typically presented with some sort of calendar/appointment book combination in which he can schedule in own activities. Depending on the user's privileges, the user may be able to view other users' calendars and make appointments in them. Users are also typically allowed to schedule a meeting and invite other users. The application will then automatically check the other users' schedules for conflicts. If none are found, the meeting is scheduled and the invitees are notified automatically. Some scheduling applications will even help users share resources, such as conference rooms, and avoid conflicts. There are several products of this nature commercially available including Meeting Maker [6, 29], OnTime [6, 30, 31], Calendar Manger [31], CyberScheduler [31], and TimeCruiser [31]. There are also applications such as personal information managers (PIMs) that include scheduling among several other features. Consultant [32], Organizer [33], and Outlook [34] are examples of PIMs of this type. Word Perfect Office comes with a utility called Central that gives scheduling capabilities to that package [35].

The second type of coordination system is from the manager's viewpoint. The basic idea behind such systems is to provide project managers the capability to track a project's progress. This allows the manager to coordinate the project team's efforts and resources to assure the project is completed on time. A typical system of this type allows a manager to create a plan to complete the project and delegate particular tasks to team. Team members receive their assignments on-line and make status reports the same way. Systems typically have the capability to generate

various charts, such and Gantt and PERT, and make time and cost estimates. Microsoft's Project is an example of this type of application as is CA-SuperProject, TurboProject, and Time Line Solutions. [36]

A slight variation of the management type of coordination system is the workflow system [5, 20]. The basic principle of workflow systems is that most business processes are accomplished by completing a series of tasks in a specific order. The process of completing these sub-tasks can be completed and tracked by routing a message between individuals. These systems typically allow the route to be predefined and allow a manager to see the current status of the project being run through the system. Like scheduling systems, workflow systems may be stand-alone products like Parametric Corporation's WindChill [37] and ActionWorks Metro [38] or integrated into a larger system like FileNET's Visual WorkFlo module in Panagon [39].

Again Lotus Notes [6, 20, 21, 22, 23], Microsoft Exchange [21, 22, 24], Novell Groupwise [21, 22, 25], and Netscape Collabra [22, 26] are among the most popular scheduling applications. All four either have a calendar and scheduling built into the system or provide an add-on product to give that capability. The same is true for management coordination systems. In the very least, these applications provide simple workflow management.

Conferencing systems

Whereas the primary function of message-based systems is to provide asynchronous communication, the main purpose of conferencing systems is to provide or enhance synchronous communication. The application typically thought of in this area is video conferencing where users can actually have a face-to-face communication, but there are others. Conferencing systems can be categorized into three basic areas: conferencing tools, conferencing aids, and contact tools. [2, 6, 22]

Conferencing tools are applications that actually provide a means for users in different locations to communicate in real-time. The medium for this communication may be video, audio, text, or some combination. Furthermore, the application may be a specialized hardware device, such as products offered by the C-Phone Corporation [40]. However, most researchers and developers tend to focus on *desktop conferencing* [8, 22]. In desktop conferencing, the conferencing tool is a part of the user's computer system, which gives the user the convenience of real-time communication while still having access to his workstation [6, 22]. Text-based chat programs such as IRC [20] are still the most common applications, but audio and video applications are becoming more prevalent. Most advance conferencing tools such as Internet Phone [41], CU-SeeMe [42], and NetMeeting [42] allow users to use text, audio only, or audio and video for real-time communication.

Conferencing aids are applications that enhance an existing conferencing system and are often part of a desktop conferencing application. The two most common examples of such applications are whiteboards and application sharing programs. In whiteboard applications [5] all users participating in the conference are presented with window in which they can place text and graphic information. As changes are made all user views of this window are updated to reflect these changes. The window basically becomes a shared scratch pad that allows users to help sort out ideas. Visual Rendezvous [43] is an example of this type of application, and CU-SeeMe [42] and NetMeeting [42] have whiteboard utilities as components.

Application sharing programs are simply utilities that allow a user to activate an application on his desktop and share it with the people in the conference. Depending on the software this may mean other users are simply allowed to view the work or that they can actually be given control of the application. Some sharing utilities only allow certain applications to be shared. Visual Rendezvous [43] allows shared web browsing, while Netmeeting [42] allows any application to be shared.

One of the drawbacks of using conferencing tools is that the meeting must typically setup in advance using channels outside the conferencing tool. *Contact tools* help alleviate this problem by keeping track of who is currently available for a conference. This allows spontaneous conferencing to take place. An application of this type usually allows a user to specify people that he wants to be aware of when they are online. Once the user has been made aware of a person's presence online, the user can contact the person to set up a conference through the contact tool. ICQ [44], Yahoo!Messenger [45], and Netscape AOL Instant Messenger [46] are contact tools that works in this way.

Group decision support systems (GDSS)

Groupware applications of this type attempt to take into account the views of everyone in the group and present some sort of summary to aid in the decision making process [5, 8, 20, 22]. The goal is to either speed up the process, improve the decision quality, or both. GDSS can be categorized according to the level of decision-making they support [20, 22]. Level 1 systems are very similar to the conferencing systems described above but provide additional functionality to help meeting organization, to encourage participation, and to allow voting to take place. These systems typically provide summary information about the ideas presented and voting results. Level 2 systems have the same functionality as level 1 systems but also add features to help alleviate uncertainty and clarify a group's position on the topic. This is typically accomplished by applying more structure to the process. For example, the system may attempt to put the group's ideas into a particular decision making model such as planning and financial models or resource allocation models. Finally, there are

level 3 systems that add even more structure. Typically such systems attempt to actually structure the meeting process. For example, the system may use rules of order to determine the order in which group members speak.

The first GDSS were implemented as electronic meeting rooms. Electronic meeting rooms are specially built conference rooms that have technology added to implement GDSS. The Decision Support Laboratory at the University of Arizona [8, 20, 22] is one of the better known examples of this type of GDSS. However, this is an expensive endeavor, and most GDSS are now designed for use on local area networks or the Internet. GroupSystems [20, 47], VisionQuest [20], and TCBWorks [20, 48] are all GDSS, but provide varying support levels.

Multi-user editors

As the name suggests, groupware in this category allow multiple users to edit a single document. A document may be text-based, graphic-based, or some combination so this category would include collaborative drawing tools and collaborative writing tools. Tools in this category have similar characteristics to the conferencing aids described in the conferencing section. The distinction between the two is that conferencing aids tend only to be useful during a conference and are not designed for the creation of documents. For example, a whiteboard application would not be very useful to a single user and would not be typically used to create a memo for the office.

Though there have been many approaches to multi-user editing, they can be divided into two basic categories: shared mind and division of labor [11]. In a *shared mind* system, all group members can edit the same file and submit their changes. A coordinator will then integrate the changes into a unified document. For example, GROVE [8] is a real-time text editor that allows shared items to be viewed and edited through the *outline window*. The window not only provides the shared contents in an outline but also group information such as session members and item read/write permissions. SEPIA [49] is a shared mind hypertext system that provides various modes, or levels, of collaboration. In the independent mode, users may work on their own tasks without interfering with each other. In the loosely-coupled mode, users may share certain public information while working on their own tasks. In the tightly-coupled mode, users will share the same view, but resources, e.g., mouse and file, are strictly controlled to avoid conflicts.

In a *division of labor* system, the document is divided into parts on which different members can work. For example, MILO [50] supports distributed asynchronous collaborative writing by creating structured documents in a user-defined graphical tree-like representation. The nodes in the tree are referred to as *notes*, and each note stores social information such as creator, creation time, and creator along with its document section. Notes also have permissions set to control user access. To generate a completed document, the notes of the tree are merged. Alliance [51] and Duplex [52] provide for a similar scheme of structuring documents; however, the structure is more formalized.

Collaborative Programming

There are two distinct types of collaborative programming tools. First, there are tools that are designed to help groups collaborate on a programming project. These tools may simply provide basic version control to help keep members from destroying each other's work or take an active role in helping the group build the program. Although version control [53] can be applied to any type of file that undergoes revisions, it is commonly associated with application development due to the numerous iterations involved in that process. It is often associated with a group collaborating on a programming project. The more advanced systems are sometimes classified as Software Configuration Management (SCM) systems, but their primary function is still version control.

The main purpose of a version control system [53] is to track the history of revisions made to files. Files are normally stored in a central location or a repository. From this location, users can check out a file, make revisions to it, and then put it back. While the file is out, the file is locked so other users who try to check it out will only get a read only version. When the file is returned, it is assigned a new identifier. In this way, the revision history is maintained and any version of the file can be retrieved. Most systems also have some method to compare versions to see what changes have been made.

Most modern systems take advantage of new technologies to increase their functionality. For example, MKS Source Integrity and StarTeam can both use the World Wide Web to distribute files [53]. ClearCase [54] is classified as an SCM system. Although its primary function is version control, it also helps developers manage the various builds of their applications. It is available on most major operating systems and integrates with popular development tools. Add-on products provide additional functionality. ClearCase Multisite allows geographically separated teams to share files by controlled replication of the repository. ClearGuide adds project management to the SCM system and allows a manager to assign tasks to individuals and track the progress.

There are many computer-aided software engineering (CASE) tools available commercially. All are designed to support part or all of a software life cycle, and many have limited collaborative capabilities. In most CASE tools, this simply means all group members have access to all files and that the CASE tool provides some version control method. Simply Objects [55], QASE [56], and ICONIX [57] are examples of this type of tool.

Development kits that can be used to create groupware applications are the second type of collaborative programming tool. There are numerous kits available for a

variety of languages, but most simply provide three services: a framework for supporting groupware applications, some pre-built groupware applications, and the necessary libraries to create new applications. DistView [58] is a Java-based tool kit that fits this description and is being developed at the University at Michigan. Tango [59] is a very similar product being developed at Syracuse University. GroupKit [60] from the University of Calgary is Tcl/Tk-based, and Egret [61] from the University of Hawaii uses Emacs as its foundation.

5. Conclusion

Research in CSCW in general and groupware in particular can be a rather arduous because of the rather hazy boundaries that define groupware. By solidifying these boundaries, the research into this area can be more focused. It is hoped that the information collated here can be a solid starting point for new research into the area of groupware and serve as a guide to the development of newer and better groupware systems.

6. References

[1] Liam J Bannon and Kjeld Schmidt, "CSCW: Four Characters in Search of a Context", *Readings in Groupware and Computer-Supported Cooperative Work*, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993, pp. 50-56.

[2] Andree Woodcock, "Guide to Computer Supported Co-operative Work", dougal.derby.ac.uk/andree/cscw6.html, 1997.

[3] Richard Power and Lorella Carminati, "Computer Supported Cooperative Work", *Cooperation Among Organizations: The Potential of Computer Supported Cooperative Work*, Springer-Verlag, Germany, 1993, pp. 13-25.

[4] Saul Greenberg, "Computer-Supported Cooperative Work and Groupware", *Computer Supported Cooperative Work and Groupware*, Academic Press Inc., San Diego, CA, 1991, pp. 1-7.

[5] "Groupware", www.usabilityfirst.com/groupware/, 1998.

[6] Clive Holtham, "Groupware: It's Past and Future", *Groupware in the 21st Century*, Praeger Publishers, Westport, CT, 1994, pp. 3-14.

[7] Rose Lockwood, "The Groupware Market", *Computer Support for Co-operative Work*, John Wiley & Sons Ltd., England, 1994, pp. 3-18.

[8] C A Ellis, S J Gibbs, and G L Rein, "Groupware: Some Issues and Experiences", *Readings in Groupware and Computer-Supported Cooperative Work*, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993, pp.9-28.

[9] Richard T Watson, Robert P Bostrom, and Alan R Dennis, "Fragmentation to Integration", *Groupware in the 21st Century*, Praeger Publishers, Westport, CT, 1994, pp. 29-39.

[10] M Sharples, J S Goodlet, E E Beck, C C Wood, S M Easterbrook, and L Plowman, "Research Issues in the Study of Computer Supported Collaborative Writing", *Computer Supported Collaborative Writing*, Springer-Verlag, Germany, 1993, pp. 9-28.

[11] M Sharples, "Adding a Little Structure to Collaborative Writing", *CSCW in Practice: An Introduction and Case Studies*, Spring-Verlag, Berlin, 1993, pp.51-68.

[12] D Diaper, "Small-Scale Collaborative Writing Using Electronic Mail", *CSCW in Practice: An Introduction and Case Studies*, Spring-Verlag, Berlin, 1993, pp.69-92.

[13] Kai H Chang, Liam Murphy, Jonathan D Fouss, Timothy D Dollar II, Byong G Lee, and Yifang Chang, "Software Development and Integration in a Computer Supported Cooperative Work Environment", *Software – Practice and Experience*, Vol. 28(6), 1998, pp. 657-679.

[14] G N Gilbert, "CSCW For Real: Reflections on Experience", *CSCW in Practice: An Introduction and Case Studies*, Spring-Verlag, Berlin, 1993, pp.39-50.

[15] A Kirkwood, S Furner, W Ablard, B Clark, K Dickerson, A Mercer, S O'Donnell, Y Siu, and O Williams, "Usability trailling for CSCW techology: lessons from a structured messaging task", *CSCW in Practice: An Introduction and Case Studies*, Spring-Verlag, Berlin, 1993, pp. 163-176.

[16] Microsoft, "Outlook Express Home Page", www.microsoft.com/windows/oe, 1999.

[17] Microsoft, "Hotmail – The World's FREE Web-based E-mail", www.hotmail.com, 1999.

[18] Qualcomm, "Qualcomm's Eudora Place", www.eudora.com,1999.

[19] Netscape, "Netscape Communicator", home.netscape.com/cummunicator/v4.5, 1999.

[20] Efraim Turban and Jay E Aronson, *Decision Support Systems and Intelligent Systems* Fifth Edition, Prentice Hall, New Jersey, 1998.

[21] Matt Kramer, "Groupware: Unleashing the Power", PC Week Online, www.zdnet.com/pcweek/, ZDNet, 1999.

[22] George M Marakas, *Decision Support Systems in the 21st Century*, Prentice Hall, New Jersey, 1999.

[23] "Lotus Notes", www.lotus.com/home.nsf/welcom/notes, Lotus, 1999.

[24] "Microsoft BackOffice – Exchange Server", www.microsoft.com/exchange, Microsoft Corporation, 1999.

[25] "Novell: GroupWise", www.novell.com/groupwise, Novell, Inc., 1999.

[26] "Netscape Collabra Server Datasheet", www.iplanet.com/products/infrastructure/messaging/n_collabra/index.html, Sun-Netscape Alliance, 1999.

[27] Bill Roberts, "Still Miles to Go: Web-based collaboration is improving, but Notes rules", **Web Week**, Vol. 2, Issue 6, Mecklermedia Corp, 5/20/1996.
[28] Daniel LaLiberte, "WWW Collaboration Projects", www.hypernews.org/Hypernews/get/www/collaboration.html, 1997.
[29] "Meeting Maker Group Scheduling", www.meetingmaker6.com, On Technology, Corp., 1999.
[30] "Open Text – OnTime", www.opentext.com/ontime, Open Text Corporation, 1999.
[31] William Wong, "Meeting Planning Made Easy", Internet Business, www.zdnet.com/enterprise/e-business, ZD Inc, 1998.
[32] "Consultant", www.chronosnet.com/consultant, Chronos LC., 1998.
[33] "Lotus Organizer", www.lotus.com/home.nsf/welcome/organizer, Lotus, 1999.
[34] "Microsoft Office – Microsoft Outlook", www.microsoft.com/office/outlook, Microsoft, 1999.
[35] "WordPerfect Office 2000: Home Page", www.corel.com/Office2000, Corel Corporation, 1999.
[36] Nelson H. King, "According to Plan", **PC Magazine**, ZDNet, 8/21/1998.
[37] "Windchill Product Information Homepage", www.ptc.com/windchill, Parametric Technology Corporation, 1999.
[38] "ActionWorks Metro", www.actiontech.com, Action Technologies, Inc., 1999.
[39] "FileNET: Panagon Integrated Document Management", www.filenet.com/solutionsProducts/panagon, FileNET Corporation, 1999.
[40] "C-Phone Video Communication Systems", www.cphone.com, C-Phone Corporation.
[41] "VocalTec Communications – Products", www.vocaltec.com/products/products.htm, VocalTec Communications Ltd., 1999.
[42] Gregg Keizer, "Meeting of the Minds", www.cnet.com/content/reviews/compare/netconference, CNET, 1997.
[43] "Visual Rendezvous Real-Time Web Collaboration and Conferencing", rendezvous.visualtek.com, VisualTek Solutions, Inc.
[44] "ICQ – World's Largest Internet Online Communication Network", www.icq.com, ICQ Inc., 1999.
[45] "Yahoo! Messenger", messenger.yahoo.com, Yahoo! Inc., 1999.
[46] "Instant Messenger", www.netscape.com/aim/, Netscape Communications Corporation, 1999.
[47] "Overview of Groupsystems Products", www.ventana.com/html/overview.html, Ventana Corporation, 1999.
[48] "TCBWorks: Webware for Teams", tcbworks.cba.uga.edu, The University of Georgia Research Foundation, 1996.
[49] H Ishii and N Miyake, "Toward an open shared workspace: computer and video fusion approach of TeamWorkStation", *Comm.* ACM, 34(12), 1991, pp. 37-49.
[50] Jones, S., "MILO: A Computer-Based Tool for (Co-)Authoring Structured Documents", Mike Sharples(Ed.), Springer-Verlag, 1993, pp. 185--202.
[51] Decouchant, D., V. Quint and S. Kiesler, "Structured and Distributed Cooperative Editing in a Large Scale Network", Groupware and Authoring, Roy Rada (Eds), Academic Press, 1996, pp. 265--295.
[52] Pacull, F., A. Sandoz, and A. Schiper, "Duplex: A Distributed Collaborative Editing Environment in Large Scale", Proceedings of CSCW 94, October 1994, pp. 165--173.
[53] Robert DelRossi, Version Control Hits the World Wide Web, Software Development, Dec. 1996, Vol. 4, No. 12, Miller Freeman Inc, pp. 64-70.
[54] "Rational Software: The e-development Company", www.rational.com, Rational Software Corporation, 1999.
[55] "Adaptive Arts Home", www.adaptive-arts.com, Adaptive Arts Pty Ltd, 1999.
[56] "AES – Computer Systems Performance Modeling", www.astes.com, AST Engineering Services Inc., 1999.
[57] "Iconix Home Page", www.iconixsw.com, ICONIX Software Engineering, Inc., 1999.
[58] "DistView Collaboratory Toolkit", www.eecs.umich.edu/distview/, University of Michigan, 1998.
[59] "Collaboratory on the Web: TANGO", trurl.npac.syr.edu.tango, TANGO Interactive.
[60] "GroupKit Home Page", www.cpsc.ucalgary.ca/projects/grouplab/groupkit, University of Calgary, 1998.
[61] "Collaborative Software Development Laboratory", csdl.ics.hawaii.edu, University of Hawaii, 1999.