# VIRTUAL GROUPS: A WEB BASED ELECTRONIC CONFERENCING SYSTEM FOR ONLINE EDUCATION

by

Ashish Pimplapure

B. Tech, Computer Science & Engineering
Indian Institute of Technology
Bombay, India, 1992

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science


SIMON FRASER UNIVERSITY
August 1996

ISBN 0-612-17054-3

SIMON FRASER UNIVERSITY

# PARTIAL COPYRIGHT LICENSE



Title of Thesis/Project/Extended Essay

**Virtual Groups: A Web Based Electronic Conferencing System for Online Education.**

Author:
(signature)

(name)

August 15, 1996
(date)

# APPROVAL

**Name:** Ashish Pimplapure

**Degree:** Master of Science

**Title of thesis:** Virtual Groups: A Web Based Electronic Conferencing System for Online Education

**Examining Committee:** Dr. Tiko Kameda
Chair

Dr. Thomas W. Calvert
Senior Supervisor

Dr. Jiawei Han
Supervisor

Dr. Linda Harasim
Supervisor

Dr. Ze-Nian Li
External Examiner

**Date Approved:** 13 August 1996

# Abstract

Collaborative learning and knowledge acquisition based on the needs of the learner have proven to be more effective than traditional learning which focuses on content and follows a rigid format. In the context of online education, collaborative learning requires a mechanism for enabling communication among the learners. Such a mechanism is usually established with the help of a conferencing system, which at a minimum supports textual messages and is preferably extendible to handle multimedia content. The conferencing system allows for the creation of "virtual spaces" which model the traditional classrooms, seminar rooms and even cafes for social interaction.

Previous attempts at providing support for group communication have either focused on using conventional tools (such as mailing-lists or newsgroups) or developing conferencing systems which need special client software or protocols for access (e.g., FirstClass). These tools however, are not suitable for use on the Internet whose recent growth has made it a promising medium for delivering online education. Using the Internet for online education is already common and will become pervasive as Internet accessibility continues to increase.

This thesis is an attempt to evaluate the feasibility of the world wide web as a medium for group communication in the context of online education. It presents the design and implementation of a web based conferencing system which uses the store-and-forward method for asynchronous communication. An important characteristic of online education systems is that they provide a mechanism for evaluating participation. Besides enabling group communication, the system also provides tools for quantitative analysis of communication (which allow the instructor to evaluate the

performance of the learner) and supports interoperability with other online education tools which are being developed concurrently as part of the Virtual-U project at Simon Fraser University.

# Acknowledgments

A thesis about collaborative work could not have been mere individual effort. Several people have contributed to the development of the system, and have helped it become a useful tool.

I would like to thank Linda Harasim for initiating the Virtual-U project, and establishing a vision for online learning environment using the state of the art networking technology. Her insights and rich experience in the field of online education have helped us identify and understand the needs of educators. I am thankful to my supervisor, Dr. Tom Calvert, who provided the guidance, the freedom, and more importantly, an unflagging support which has made this a most rewarding experience. Special thanks to Chris Groeneboer for all her help and assistance during the various stages of the project.

Thanks to Doug Ward, Esther Tiessen and other instructors for adopting the earlier version of the system despite its severe limitations in their courses and providing valuable suggestions. Thanks also to the students of CMNS 453 (96-1) for their patience and useful feedback during the course of the semester. Gordon Ross provided many suggestions and much needed support during the deployment of the earlier version of the system.

I would also like to thank Christian Pantel and Lyn Bartram for redesigning the user interface and Bonita Nga Yi Tse for implementing it. The user interface changes have significantly enhanced the system. The patience, technical support, and enthusiasm provided by the Virtual-U development team (Richard Rupp, Max Luk, and Bonita Nga Yi Tse) has been important in bringing this project to fruition.

# Contents

# List of Figures

# Chapter 1

# Introduction

Using computers and telecommunication technologies for delivering education has been a major goal for educational computing research. The field of online education, which uses networked computers for content delivery *and* interaction, is growing rapidly and has become a viable alternative to traditional modes such as classroom based learning and distance education.

One of the salient features of online education is that it is based on the social and collaborative nature of learning, and provides a more powerful environment than traditional classroom and distance education modes. In [Har90], Linda Harasim outlines the attributes of online education and compares it with traditional modes: the physical classroom and distance learning. In a physical classroom, the focus is usually on content and the course curriculum is structured around core topics of the subject. The topics are organized in a specific format and then taught according to a well defined lesson plan [NS96]. Classroom based learning allows participants to engage in interactive group communication. Harasim [Har90] notes that traditional distance education, on the other hand, emphasizes the independence of the learner [Moo86], and is characterized as a "transmittal model" [Bur88] where content is delivered via mass communication technologies, e.g., radio, TV, and mail. In this learner centered model, the focus is on the needs and interests of the learner. Instead of following a rigid format, learners adopt a self paced approach, which is suited to their skills and learning abilities. Unlike physical classrooms, distance education is time and place independent. There is however, no support for group communication. A rather weak form of learner–instructor interaction is occasionally supported via phone or mail. Online education combines the benefits of both classroom and distance learning and offers a unique environment. Harasim [Har89] characterizes the emergence of online education as a new mode for learning. Like distance education, it is time and place independent, and built around the learner centered model. Like the physical classroom, it supports interactive group communication. The field of online education thus offers novel opportunities for educational interactivity.

## 1.1 Computer Mediated Communication

A significant challenge in the delivery of online education is providing support for communication among participants – something which is implicit in a physical classroom [Har90]. Online courses have little or no face-to-face interaction associated with them and hence require some form of computer mediated communication. A conferencing system is a tool that provides the necessary services for conducting communication. Participants do not need to be physically present during the communication and can interact from homes, schools, or offices. An electronic conferencing system is a network based application and is characterized by:

**Client** which provides the basic user interface to the system and serves as a front end to the various services (navigation, reading, editing etc.).

**Server** which coordinates the activities between the clients, maintains information about users, and manages the conferences.

**Communication Protocol** which specifies how the data should be transferred between the client and the server.

The basic unit of interaction is usually called a *message*. Related messages can be grouped together into *topics* or *subjects*. For a high level structuring, related topics are further classified into *conferences.* Communication can be modeled using a hierarchical structure (where messages are organized into different levels) or a linear one (where the messages represent a flow of conversation in a period of time).

Applications of a conferencing tool can be broadly classified into two categories: interpersonal communication and collaborative work [HT94].

**Interpersonal Communication** :
In this case the conferencing system may serve as a medium for various activities like group discussions, exchange of ideas, polling, delivering public lectures, and even social interaction. Interaction between users is either on a person to person level (1–1) or in groups of a number of people. The conferencing system provides various facilities which include creation of private or public groups, support for restricted access, structuring of message threads, etc., and offers a flexible environment for interpersonal communication.

**Computer Supported Cooperative Work** :
Here the conferencing system is regarded as a type of groupware application. Though there is no formal definition of "groupware", it usually refers to the interaction among a group of people that is characterized by the following features [Sto95]:

1. An "environment" which consists of the required software and hardware that establish context for interaction.
2. "Interactive", which refers to time constraints imposed on interaction between participants.
3. "Shared", which indicates that a participant influences and is influenced by the activities of other group members.

The basic objective of groupware is to facilitate computer supported cooperative work, and includes activities such as task coordination, project management, collaborative design, collaborative writing, planning, etc. Unlike simple interpersonal communication, the messages here are associated with the objects that they refer to. Users therefore do not view the messages as separate entities, but a part of the data that is being manipulated. The conferencing system displays the messages in the appropriate context. For example, while collaborating on a shared document, "notes" or "comments" regarding the contents of the document may be placed in the margin or displayed using pop-up windows.

In the context of online learning, collaboration enables multiple perspectives and promotes sharing of expertise, thereby helping learners develop a greater understanding than that achieved by individual effort. The Collaborative Notebook of the CoVis project [EPG96] is an example of a tool which supports group learning.

## 1.2 Synchronous and Asynchronous Communication

Computer mediated communication may or may not require the participants to interact at the same time. In the case of synchronous communication or "real-time computer conferencing" (for example, CU-SeeMe, IRC), the users are connected to the system simultaneously and messages are delivered immediately without storage. This provides an instantaneous feedback to participants and can be useful in brainstorming, decision making etc. There is little or no support for recording the transcript of conversation. Furthermore, a dedicated connection needs to be setup for the period of communication.

Though synchronous communication eliminates the need for physical confluence, it still suffers from the serious limitation imposed by the requirement that all participants be simultaneously connected. This assumption is relaxed in asynchronous communication, where participants can interact with the system at different times. The conferencing system acts as a "store-and-forward" tool. Messages sent by a user are kept in a persistent store and may be retrieved at a later time. The asynchronous method provides a more flexible environment for communication in distance learning. Firstly, students can take their own time for providing response on a class topic.

Secondly, an instructor can "hide" the responses of other students until a student has submitted his or her own response. Also, regardless of when a person joins the discussion, it is always possible to retrieve the complete set of messages.

## 1.3 World Wide Web

With rapid developments in networking and proliferation of networked computer sites, global information access has recently received a lot of attention. The World Wide Web is perhaps the most popular service for accessing internet information. The Web merges the techniques of information retrieval and hypermedia to make a powerful yet easy to use global information system [BLCGP92]. Information is "published" in the form of documents written in a special mark-up language called HTML [BLC93] and is distributed via Web servers. Users employ clients (also known as "browsers") to access and display the documents. Data communication is carried out using a number of protocols: HTTP, Telnet, NNTP, FTP, etc. with HTTP [BL92] being the most common. The client server architecture of the Web is illustrated in Figure 1.1 (from [BL92]). Each document is identified by a URL or "uniform resource locator" that specifies the internet address of the document and the protocol for accessing it. Besides text, a Web document may also consist of objects of other media types, such as graphics, audio etc. Parts of the document can refer to other documents by *hyperlinks*. This ability to reference documents by simple links makes it a powerful medium for presenting information.

Recent developments in Web technology have further enhanced the Web capabilities. Documents have better features for formatting [Cor96c] and can also include "live content" [GM95]. Enhancements to Web servers now support transaction processing and electronic commerce, thus making the Web a basis for a much broader range of applications, many of which fall under the categories of communication and collaboration.

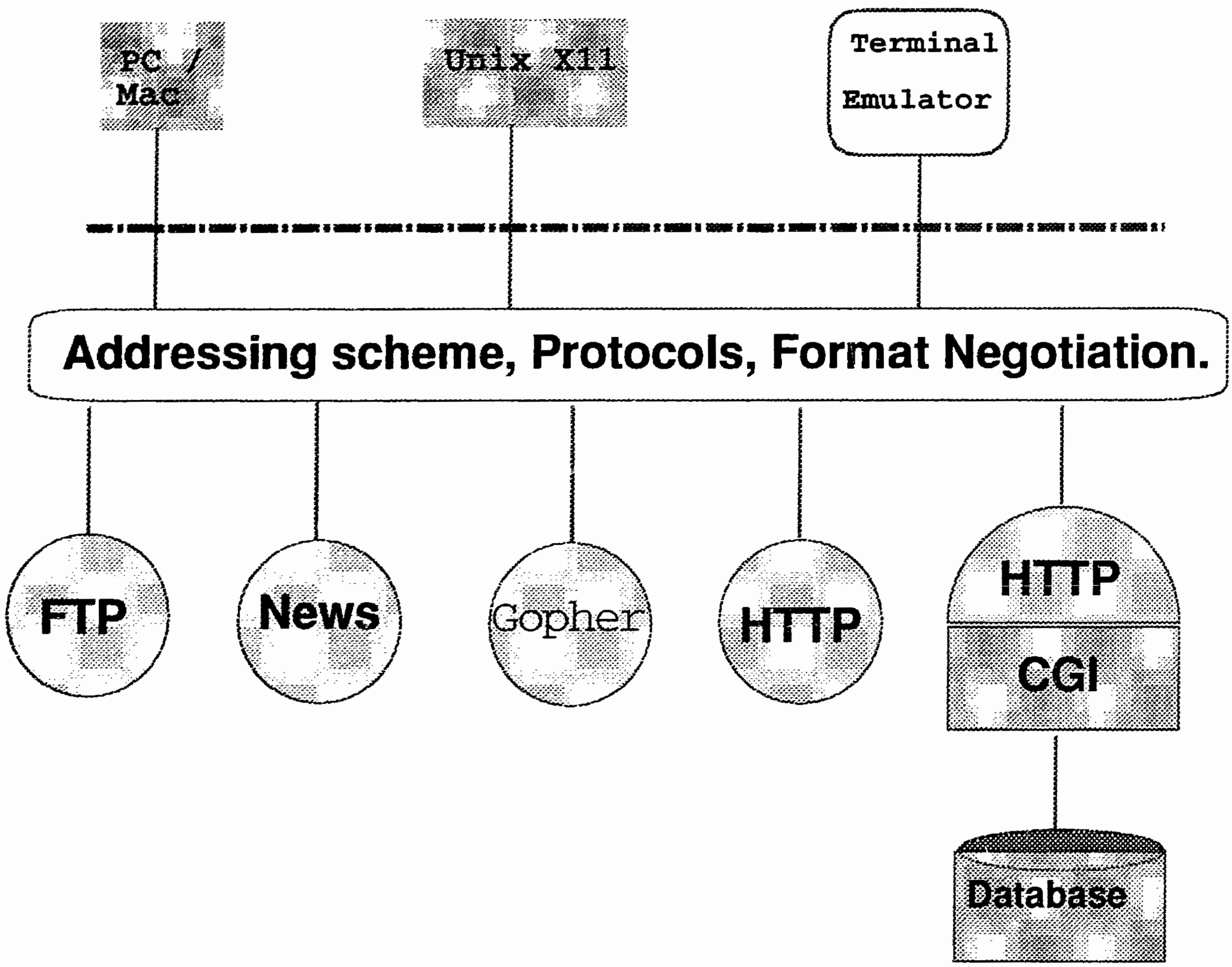


Figure 1.1: The World Wide Web client–server architecture

### 1.3.1 Online Education and the Web

The ease of information distribution, speed of retrieval and availability of a diverse set of information sources make the global network, or internet, an excellent medium for delivering online education. The Web offers unprecedented opportunities for deploying online education through the internet. Firstly, users can access the system easily without requiring any formal training. The user interface is uniform and independent of the underlying services. Both FTP and Gopher for example, can be accessed through the Web client without additional learning. Secondly, the Web is an extendible system and educational resources can be stored in different formats without having to change the access protocols. Moreover, an existing body of knowledge can be easily made available on the Web with little modifications.

One of the most important characteristics of information on the Web is that it can be presented in the form of Hypertext. Parts of text in a hypertext document can be associated with links and references to other documents. The advantages of Hypertext presented by Conklin in [Con87] imply several benefits for online education:

- Information structuring: both hierarchical and non-hierarchical organizations can be imposed on unstructured information; even multiple hierarchies can be used to organize the same information.

- Customized course-ware: course segments can be threaded together in different ways, allowing the content to be presented at varying levels of difficulty depending upon the skills of the learner.

- Ease of creating new references: users can easily construct their personal notes by making annotations or adding new links to the documents.

- Ease of tracing references: users can follow links to various documents in a uniform and consistent manner regardless of the location of the document.

- Modularity of information: since the same text can be referenced from several places, ideas can be expressed with less overlap and duplication.

Hyperlinks in a document can not only point to other documents, but also to objects which can be generated dynamically. This powerful feature enables interaction with legacy databases and information systems. Results of query processing for example can be displayed in the same format as a normal document. However, [Con87] also identifies two disadvantages of hypertext which can affect the learning process – *disorientation*, the problem of loosing the sense of "location" in the Web and *cognitive overhead*, the additional work needed to organize and classify linked documents. Hypertext, in general, tends to magnify the problems of comprehension. For example, it is not possible to quickly "skim" a set of interlinked documents without actually following each and every link.

## 1.4 Web based Conferencing

### 1.4.1 Motivation

Traditional conferencing systems serve as dedicated applications for exclusively supporting communication. For several reasons this model does not fit well with the needs of an open, internet based environment for online education:

1. Users are required to obtain special client software for accessing the system. This may also entail learning a totally new user interface.

2. The conferencing systems often use a proprietary protocol for client-server communication.

3. Since the conferencing system usually functions as a stand-alone application, there is no inherent support for interoperability with other applications. Upgrading and maintenance of the system is also independent of other applications.

4. The conferencing system does not provide any support for seamless integration with external resources. For example, while discussing a student term paper, it is not possible to directly refer to the documents of the term paper from within a message.

For these and several other reasons, conferencing for internet based distance learning requires a complete restructuring of the system, which includes enhancements of services of both the client and the server and a high level implementation which is platform independent.

### 1.4.2 Designing Web based Systems

Conferencing on the Web eliminates most of the problems faced by traditional systems described in section 1.4.1. Messages in a conference can be presented in the same style as normal Web documents. When messages are in hypertext format, they can also refer to other resources. Web based conferencing systems enable users to access and manipulate the conferences in the same way as other Web documents. There are two approaches for designing Web based systems: extending traditional conferencing systems with Web capabilities or building a system from scratch. In the former case, a mediating layer is provided between the Web server and the conferencing system which maps the Web requests to the commands of the conferencing system. Since the Web clients interpret HTML documents, messages are either replicated and stored in html format or translated into html on the fly [Inc95, Cor96b]. An advantage of such an extension is that users already familiar with the native system can now access it freely like other internet resources. A big disadvantage however is that the extended system is still constrained by the limitations of the native system. Moreover, since the Web itself is in an evolutionary phase, new developments on the Web cannot be easily incorporated into traditional systems by mere extensions.

A more promising but time consuming approach is to build a system from scratch. This allows re-implementation of the basic architecture so that it is optimized for Web services.

## 1.5 Thesis and Organization

The objective of this thesis is to evaluate the feasibility of the Web as a medium for group communication in the context of online education. Although there have been several efforts for building Web conferencing systems [Woo96], none seems to address the particular needs of an educational environment. Firstly, these systems are usually built for general purpose usage, and are intended for the open internet community. There is, therefore, little emphasis on activity tracking on a per-user basis, something which can be a useful metric while evaluating a student's participation. Secondly, these systems do not provide multiple views of a conference. For example, instructors may wish to see contributions by a specific student, or activity for a specific period of time, etc. Finally, since the primary (and often, the only) objective is to support group discussions, it is not possible to enable integration with other tools for online education, such as an annotation tool or the instructor's grade book etc. This thesis has attempted to address these and other issues related to conferencing in online education.

The next chapter discusses various issues related to electronic conferencing. This is followed by an analysis of some popular systems which are currently being used. A design overview of Virtual Groups is then presented. Issues which affect the performance and usability of the system are discussed and finally, some ideas for future work are outlined.

# Chapter 2

# Background

## 2.1 History

The earliest forms of communication on computer were modeled using electronic messages among users on the same machine. Computer conferencing, a system for many-to-many interaction among dispersed users, was invented in the early 1970s by a number of researchers. In 1971, Murray Turoff [TH77] designed EMISARI (Emergency Management Information System and Reference Index), a special purpose system to coordinate the Nixon administration's wage price freeze. This is generally recognized as the first computer conferencing system. Turoff regarded computer conferencing as a mechanism to enable the "collective intelligence" of a group of people. Turoff later developed EIES (Electronic Information Exchange System) , a testbed for experimenting with the nature of computer mediated communication. EIES served as an ancestor to several systems. Peter and Trudy Johnson-Lenz worked with Harry Stevens to create "Tegitec" using the scripting language of EIES. In 1979, Harry Stevens and others created Participate, a conferencing system to structure short discussions around questions and answers which could later be searched for specific information.

Application of computer technology to planning and other activities in the government was a potential growth industry in the 1970s. The US National agencies DARPA and NSF sponsored a group at IFTF (Institute For The Future) to develop a planning and forecast tool, which resulted in the creation of PLANET, or Planning Network.

PLANET later evolved into Notepad, a private but global conferencing system, which was adopted commercially and is still used by a number of large clients.

Besides efforts in computer conferencing, there were also other related developments which occurred in the late 1970s. In 1977, researchers at the Bell Laboratories created UUCP (Unix to Unix system copy), a utility to enable data communication between two Unix machines via a modem. This allowed different Unix hosts to exchange electronic mail. In 1979, graduate students Tom Truscott and James Ellis of Duke University, along with Steve Bellovin of UNC, developed a UUCP based tool to support structured interchange of messages. This later evolved into what is now known as USENET, perhaps the most wide reaching public discussion system. Computer conferencing, which began as an application for government activities, and was later adopted by corporate and scientific sectors, was now available to the general community.

### 2.1.1 Educational Conferencing

Although educational computer conferencing is a relatively new phenomenon, its roots had emerged almost a decade before the general conferencing systems. In early 1960s, Don Bitzer, a faculty member at the University of Illinois, Urbana, founded CERL (Computer based Education Research Laboratory). They developed PLATO [Woo94], a system for computer based education. Built as a time sharing system, PLATO allowed students and course-ware authors to interact via graphics display terminals connected to a central mainframe. In the early 1970s, PLATO was replaced by Plato Notes, a more general purpose system which supported a large number of users. The basic unit of interaction was a "note", and related notes were appended to a "notesfile". Due to hardware and other limitations, the size of a note was limited to 20 lines and each note had a fixed format. Subsequently, the developers added support for 1-1 communication, grouping notefiles by categories, synchronous communication, and access control.

Educational conferencing was generally supported in two ways: by using general purpose conferencing systems and by building systems which were tailored to the needs of an educational environment. The original motivation for using computer conferencing was to supplement classroom activity and support distance education. However, computer conferencing combined the benefits of both forms of education and resulted in the more powerful mode of online education. Harasim presents a comprehensive bibliography on educational conferencing in [Har90].

## 2.2 Networked Learning Environments

Akin to online education is the concept of networked learning environment which is based on computers and telecommunication technologies and supports a host of pedagogies and instructional models [Har93b]. Harasim [HR94] analyzes the research issues involved in the deployment of network learning environments. Recent advances in Web technologies make the internet a promising medium for enabling networked learning environments. Several efforts have been initiated to establish such environments [Inc96, Tur95]. The general goal is to enhance the effectiveness of postsecondary education using the internet. In 1995, a nationwide collaboration effort, "The Tele-Learning Networks of Centres of Excellence" , was initiated to link researchers and education communities in Canada [Har94]. Lead by Dr. Linda Harasim of the School of Communications, Simon Fraser University, this research program is an attempt to realize a knowledge based learning society. The objectives of this project [1] are (from [Har94]):

1. To develop new models of learning, learning environments, and pedagogies to better meet the needs of the workplace and the nation.
2. To develop and transfer new technologies that can manage, sustain and constructively direct networked learning.
3. To understand the nature of effective telelearning communities at all levels.

[1] An overview and general information about the TeleLearning project can be found at: http://www.telelearn.ca

4. To support a shift to these new methods in the educational practices in schools and the workplace by involving educators and trainers in the use of these technologies for their own knowledge building.

A number of *research themes* have been established to realize the efforts of the TeleLearning project. These themes have been organized into *Beacon Technologies* which will create the necessary software environments for deployment of networked learning environments. Harasim envisioned **Virtual–U** as one of the beacon technologies which will focus on the development of tools for "course design, electronic conferencing, course resource handling, and class management" [Har95b]. Based on the principles of active learning, collaboration and knowledge building, Virtual–U technology aims to provide a platform for advanced modes of teaching [HCG96]. Currently, tools are being developed for:

1. Course Design and Facilitation.
2. Class Discussion and Presentation.
3. System Administration and Resource Handling.
4. Class Management and Evaluation.

Central to the framework of Virtual–U is an electronic conferencing system. The objective of the system is not just to support group communication, but to also provide a flexible environment for collaborative learning [Har93a]. The latter requirement is an important characteristic of a conferencing system in the educational environment. In the following section, we describe the attributes of a conferencing system which make it useful in a learning environment [Har95a].

## 2.3 Electronic Conferencing: An Educator's Perspective

In this section we describe the essential elements that are desired in a conferencing system from an educator's perspective. While some of these form the basic requirements of a general conferencing system, others are unique to the context of online education.

### 2.3.1 Communication Models

Interaction among the users can range from "one-to-one", which involves exchange of messages between two individuals, to "many-to-many", where a group of users working together as a team can share their collective ideas with another group, for example, in a large scale design project. A conferencing system should be able to support the different types of communication requirements.

Another aspect of the interaction process is the **structure** of communication. A message may either initiate a new conversation or contribute to an ongoing discussion. In a 1–1 model, interaction is simple: a message may entail a reply which may generate a response from the original sender and so on, creating a chain of related messages. Group communication, however, usually entails more complex interactions. Firstly, a message may receive multiple responses, some of them generating further discussion, thus creating a tree of related messages. Secondly, a message may be in response to two or more messages (for example, in the discussion of a problem, a student's message may analyze the solutions presented by other students through previous messages), where the relationship no longer conforms to the tree structure but becomes an (acyclic) graph. Furthermore, in the case of multiple groups, a message may be relevant to more than one groups and may serve as a link between common topics. A conferencing system should support mechanisms for logical organization of messages which will preserve the structure of communication. While creating a message, users should be able to specify its relationship to other messages. While reading messages, users should be able to view them in an appropriate context.

### 2.3.2 Navigation and Browsing

Group interactions often require a number of conferences to be set up, depending upon the topic and nature of discussions. Conferences themselves may be related to each other in some manner. Users should be able to "move" from one conference to another. This creates the need for flexible navigation techniques.

At the browsing level, users should be able to obtain different views of the set of messages in a conference. Common viewing orders include chronological, sorted by author, organized by thread structure, etc. Since reading and browsing of messages is more frequent than creating new ones, the conferencing system should build and maintain index structures which can keep the "meta-information" about the messages.

### 2.3.3 Access Restrictions

Instructors often divide a class of students into a number of smaller groups and assign different tasks or activities. They may wish to assign different levels of privileges (ranging from read access to being able to moderate a given conference). A conferencing system should be able to provide flexible ways to specify various levels of privileges.

### 2.3.4 Administration

The conferencing system needs to provide support for various system administration activities which include creating new users, modifying privileges, purging old messages, removing expired conferences, etc.

### 2.3.5 Activity Tracking

Since interaction occurs asynchronously, different users view the system in different states. Each user may have his or her own set of subscribed conferences. Furthermore, depending on the activity, the set of unread articles may differ. To facilitate subsequent interaction, the system needs to track the activities of every user and maintain a user profile.

### 2.3.6 Evaluation Tools

Measuring participation is an important aspect during the analysis of communication in online education. An instructor would like to obtain the statistics for a given user (e.g., number of messages read, created etc.) or for a conference (e.g., number of active participants, total messages posted etc.). A conferencing system should be able to provide mechanisms for generating the various statistics.

### 2.3.7 Interoperability

In the context of distance learning, the conferencing system is one of the many online resources available to the student. Students should be able to access the various resources in a seamless manner – contents of a conference should be accessible from other course resources (such as lecture notes) and contents of a message should be able to refer to external objects.

### 2.3.8 Customizability

The set of requirements for conferences are diverse, depend upon the type of communication desired, and can be often conflicting. For example, some conferences may allow discussions at arbitrary levels, while others may limit the reply level to one. Some conferences may impose a limit on the size of the message posted or limit the activity to certain periods of time. The conferencing system should be able to support different types of conferences by providing customizable features.

## 2.4 System Design Issues

### 2.4.1 Storage

Asynchronous communication requires that the messages be kept in a persistent store until (and even after) they have been delivered. The system may either maintain its own storage or forward the messages to users' personal mailboxes. In the latter case however, the responsibility of maintaining the communication structure is delegated to the user. The storage strategy has important implications for *naming*. When the system maintains its own (logically) centralized store, each message can be identified by a unique id. and can be referred to easily in a thread of discussion. All the users get a consistent view of the system. However, when the messages are stored in personal mailboxes, users may have their own naming schemes and referencing becomes quite complex. Moreover, when messages are deleted, different users may obtain mutually inconsistent views of the system.

### 2.4.2 Security

Security issues in a conferencing system stem from three important requirements:

**Authentication** Allowing only authorized users to access the system services.

**Access Control** Limiting the access for a conference to a particular group of people as specified.

**Encryption** Encrypting data during transmission so that it is not "sniffed" during transmission over public networks.

The system needs to provide unique user ids to every user along with a password to correctly identify them. Users may be further classified into categories (for example, staff, instructor, student) which have different privileges. Each conference can have a set of access control lists associated with it which specify the type of privileges for different users. Some users for example may only be able to read messages in a conference, while others can also create new ones. The conferencing system should provide mechanisms for specifying and modifying access privileges. Since users access the system over a public network, it is desirable that the conferencing system provides some encryption mechanism during data transmission. In order to prevent unauthorized access to certain data from valid users, the system may also choose to store the data on the server in an encrypted format.

### 2.4.3 Network Protocol

Since the conferencing system is based on a client–server model, an important aspect of the system is the networking protocol used for transmission of data. The network services should be independent of the underlying technology. Users of the service should not be aware of the physical implementation of the network. Furthermore, since the system needs to be interoperable with other tools, there is a need for a uniform addressing scheme so that users are able to access various types of data in a consistent manner.

### 2.4.4 Human Factors

The primary objective of a conferencing system is to facilitate communication between end-users and nearly all the activity in the system is user driven. A conferencing system therefore is a user-centric application and the design should address the following issues:

**Skill Levels:** What should be the minimal skill level of the persons who use the system? Is mere familiarity with computers sufficient or is a stronger background assumed?

**Training:** What sort of training needs to be provided for effective use of the system? Are online tutorials sufficient or is hands-on training required? Can a short training session enable the users to start using the system in a rudimentary way?

**Interface:** Is the user interface intuitive? Does it mesh well (with regards "looks and feel") with the UIs of other tools or is it too application specific?

**Support:** Is online help provided? Is there support for easy and prompt interaction with "help-staff"?

**Transmission Cost:** Does the average user have access through fast links to support rapid transmission of multimedia documents? Is there an option for disabling multimedia content? If the costs of connection are prohibitive, is there support for off-line reading of a conference?

### 2.4.5 Other Issues

There are a number of other issues which do not form the "basic requirements" but can enhance the conferencing system features and are often desirable:

#### Message Editing

For creation of a message, conferencing systems usually provide a rudimentary editor with the client. While it works for simple text editing, creating messages with multimedia content or in mark-up language like HTML remains a problem. A conferencing system should either provide a WYSIWYG editor or be able to provide wrappers for accepting documents created with other editors. The latter approach though expensive, allows the people to use their favorite editor.

#### Search & Filtering

In a conferencing system with a large user base, the number of conferences and messages may grow arbitrarily. Even with navigation and browsing techniques, the handling of numerous messages becomes intractable. With the aid of search tools, the users can quickly find the desired messages by specifying the relevant keywords. With information filtering tools, users can specify the topics of their interest and ignore all irrelevant messages.[YH95].

#### Gateway Mechanisms

For general purpose communication needs, users also interact with other communication systems such as using email and subscribing to USENET newsgroups. A conferencing system can provide a gateway between itself and other services to facilitate exchange of messages across systems.

# Chapter 3

# Case Studies

In this chapter we look at some of the popular tools for group communication which have been used in the educational domain. Electronic mail is probably the most successful form of groupware developed and deployed to date. It is easy to use, fast, can be addressed to multiple receivers, and can be operated through a variety of user interfaces, ranging from simple, text based ones to those which can support multimedia documents. Electronic mail is used widely in various educational institutions. USENET is perhaps second only to electronic mail in terms of popularity and usage. Like electronic mail, it is easy to use and supports threaded discussions. FirstClass is a proprietary system which integrates personal communication with group conferencing. Finally, we describe Lotus Notes, another proprietary system which is perhaps the most popular groupware system and supports communication, collaboration and coordination of group activities. Notes also provides a rich application development environment for customization.

Each of these systems is described in the following manner. First, a general overview of the *system architecture* is presented. This is followed by the description of the *naming and storage* mechanisms. The *navigation and browsing* features are described, and finally, the pros and cons of the system as a group communication tool are discussed.

## 3.1 Electronic Mail

### 3.1.1 System Architecture

Electronic mail, or email, is one of the most widely used services on the Internet. In its simplest form, it allows a user to send a message to one or more users. An electronic mail system is characterized by two components: **Mail Transport Agents**, commonly referred to as MTAs, which are programs that handle the delivery of mail for all users on a computer and forward mail between machines and networks (e.g., sendmail on Unix), and **Mail User Agent**, or MUA which is a program users run to send and receive mail. Common examples for MUA include Elm, Pine, MH, etc. Separating transportation and presentation creates a level of abstraction and allows a greater flexibility for user preference. So users can choose their favorite MUA independently of the underlying MTA. Interface between MUA and MTA is defined by a **maildrop**, which acts like a warehouse. (On some Unix systems for example, "/usr/spool/mail" is the directory which contains a file per user for storing incoming mail.) Email is exchanged among various various MTAs over the network by using mail transfer protocols, such as SMTP [Pos82]. It may not be possible however, to maintain an MTA on every computer. Personal computers for example, may not have sufficient resources to host an SMTP server. ( Users are often charged on the basis of "connect time", so it may not be feasible to maintain a full-time internet connectivity.) It is still desirable to serve email on such machines and provide an MUA to access mail. This is generally achieved by using a protocol such as POP [MR94] which enables the small machine to contact a remote maildrop. To send a mail however, users do need to establish a temporary SMTP connection.

The basic unit of information exchange is **message**, which consists of two parts: a header, and a body. The header contains information *about* the message, such as the names and addresses of the sender and recipient(s), date, an optional subject of the message, etc. Information is organized in multiple lines, called header fields. Each header field comprises of a field-name and a field-body. A distinction is made between header fields which are *required* and the ones which are *allowed*. The former category contains mandatory information which is used for delivery of the message (for example, address of the recipient) while the latter may contain optional information which can allow for defining a rich and complex structure on the message content. RFC 822 [Cro82] discusses the standard for the format of message headers. The body contains the actual content. Since the MTAs use ASCII format to transfer data, messages cannot contain raw, non-textual data. Such data needs to be encoded before sending it by email.

### 3.1.2 Naming and Storage

Each message is identified by a message ID, which is usually an ASCII string containing the address of the machine from which the message originated, the date and time of creation, and an alphanumeric string. For example:
`<199603200840.AAA05443@indus.cs.sfu.ca>`
is the message ID for a message which originated from the machine `indus.cs.sfu.ca` on July 20, 1996, at 8:40am. The message ID is generated by the MUA and uniquely identifies the message on the internet. There is however, no specific format for message IDs and different MUAs generate message IDs in different formats. This has an important implication for naming and referencing of messages. It is not possible in general, to identify a message by its ID and messages are often referred to by their content. Such referencing is usually indicated in the body of the message, and therefore the MUAs are not always able to identify relationships between messages.

Since email was primararily designed to support 1–1 communication, the messages are not stored separately but delivered directly to the recipient(s). The MTAs deliver the message into the maildrop from where it is retrieved by the MUA at a later time. Each user has a separate maildrop and thus the storage is managed by the recipient of the message. On Unix systems, maildrops are usually text files kept in a spool directory. The MUAs provide options for deleting or relocating messages to user's personal folders. Storage management is thus completely delegated to the user.

### 3.1.3 Navigation and Browsing

The MUA provides an interface to the user for accessing email messages. Since presentation of email is separated from the delivery, a number of MUAs are available, ranging from Berkeley "mail" ( which provides a simple, text based interface), to NeXTMail, which can handle messages with multimedia rich content. Most MUAs however, provide a simple facility for navigation. Messages can be sorted by author or date. Threading is difficult or at times even impossible, since there is no standard method for naming and reference.

### 3.1.4 Communication Models

In its basic form, email supports 1–1 communication. Group communication can be implemented by creating **Mailing Lists**. A mailing list consists of the email addresses of all the group members and is itself identified by an address. Messages sent to the group are forwarded directly (without any processing) to the members of the group. Since storing of messages is done in users' personal mailboxes, there are a number of implications for the conferencing involved. Firstly, in the personal mailbox (the "maildrop"), there is no distinction between group's messages and the user's other mail. The responsibility of preserving the logical structure of the communication is delegated to the user. Since different users may adopt different maintenance policies, there is no consistent view of the communication across the users. Creation of new groups requires a greater overhead and there is no support for flexible access control. Nevertheless, mailing-lists offer a "quick-and-easy" way to perform group communication.

## 3.2 USENET

### 3.2.1 System Architecture

Based on a client/server paradigm, USENET provides a mechanism for one to many communication. Like email, the presentation of information is separated from the transport. A **news server** is a program which distributes news on various machines over the network, a **news reader** is a program which communicates with the news servers and allows the users to access and post news items. News is transferred among various news servers by the Network News Transfer Protocol, or NNTP [KL86]. This protocol basically broadcasts each article to all other sites, in contrast to email protocols which send messages to specific sites. A news server is configured by the system manager to know about one or more neighboring sites on the network which have active news servers, and there is an arrangement that these servers will communicate news to each other. (This arrangement may be limited to specific newsgroups.) Each server can receive and transmit news, and servers compare the peer's news to see if they need to fetch new news. The basic unit of information exchange is an **article**, which, like an email message, consists of a header and a body. Article belong to one or more **news groups**, which contain articles related to a certain topic. The news groups are organized into hierarchies so that large topics can be sub divided into finer areas of interest.

### 3.2.2 Naming and Storage

Articles are identified in two ways: by a "message ID" and by an article number. The message ID for a USENET article is similar to the message ID for an email message and is assigned at the machine on which the article was created. Every article has a unique message ID. The article number is the number assigned to the article in its newsgroup and is local to every news server. Articles in different newsgroups can thus have the same article number. Also, a given article may be assigned a different number on different news servers. In general, message IDs are used when it is necessary to refer to an article globally – when one article references another article, or when an article is posted to more than one newsgroup. The servers identify articles by their message IDs, so that cross posted articles are not retrieved multiple times. Article numbers are used to obtain information about articles sequentially – for example, article numbers would be used to list all the articles in a given newsgroup. Unlike email, the news articles are stored in a common repository called news spool on a machine which acts as the *news server*. The news readers maintain a **user profile** for every user which keeps track of the newsgroups which the user has subscribed to, and the articles which the user has read so far. Storage management is done by system administrators at the site level. One of the major problems with USENET storage mechanism is redundancy. Current news mechanisms do not support sharing of articles among servers. For a user to access an article, it needs to be available on the user's news server. Consequently, each widely distributed newsgroup is replicated on countless servers around the globe, consuming large amount of disc space. Articles are assigned an expiration date based on the newsgroup they are in and need to be purged for accommodating new ones. This, results in a truncated view of the system and many discussion topics are rehashed. Most newsgroups address this issue by building lists of Frequently Asked Questions, or FAQs.

### 3.2.3 Navigation and Browsing

Like email, users have the flexibility of choosing a news reading software. The news reader provides an interface for navigation and browsing among the various newsgroups. Most of the news readers allow the users to "subscribe" to desired newsgroups. Articles in a group can be sorted in various ways, such as by date, author, or subject of the article. Through user profiles, the news readers maintain a state of user's activity and can distinguish between read and unread articles.

### 3.2.4 Communication Models

The newsgroups on USENET provide a convenient way to conduct threaded discussions. Unlike mailing lists, there is no well defined set of recipients, so the discussions are freely distributed in the domain of the newsgroup. Anyone with an access to a news-server can participate in the discussions. By means of gateways, users can also interact through email. Access control for posting articles can be established by using a moderator, but reading privileges are available to all. Using newsgroups for courses is a popular means to supplement interaction in the physical classroom but is rather inadequate for full-fledged communication medium that distance learning demands.

## 3.3 FirstClass

### 3.3.1 System Architecture

FirstClass is a proprietary system which provides support for email and explicit group communication. Like USENET, it is based on a client/server paradigm but the server acts as a central repository for all information. Clients are available for Mac and Windows platforms. FirstClass uses a its own protocol called "First Class Protocol" or FCP for client/server communication. Like email, the basic unit of interaction is a message. However, FirstCiass distinguishes email messages (which correspond to 1–1 communication) from conference messages.

### 3.3.2 Naming and Storage

The server maintains a centralized repository for storing both private and group messages. Since all messages are stored logically at a single location, naming scheme is much simpler than that of email or news. Personal messages are simply identified by a number which is unique to a user, and conference messages are identified by a tuple comprising of the conference name and a message number. Users can be assigned special privileges for managing conferences which includes deleting old messages, purging conferences etc. On a systemwide level, storage management is carried out by "administrators".

### 3.3.3 Navigation and Browsing

FirstClass provides a graphical user interface for navigation and browsing. Users interact with the system through a set of *menus.* Conferences are represented by folders. Contents of a conference are displayed in a separate window. Each message is displayed in a separate window. As a consequence, a large number of windows may be created while reading a conference. (Since there is a limit on the maximum number of windows that can be created in FirstClass, users have to manually manage the number of windows on screen.) FirstClass supports activity tracking and previously read messages are marked with a red flag. FirstClass also supports a search mechanism across conferences.

### 3.3.4 Communication Models

Conventional email does not support group communication well, and systems for groupware address private mail poorly, if at all. FirstClass was an attempt to integrate personal communication and workgroup conferencing. It provides three different mechanisms for communication: *mail* for 1–1 communication, *conferencing* for group communication, and *private chat* for synchronous communication. FirstClass has been a very popular tool for group communication in educational environments. However, because of proprietary communication protocols and data formats, it does not easily lend itself to internet based learning environments.

## 3.4 Lotus Notes

### 3.4.1 System Architecture

Lotus Notes is an industry-standard client/server platform for developing and deploying groupware applications. It provides a number of solutions for various activities, ranging from messaging to group collaboration.

The basic element of interaction is called a *document*. The structure of a Notes document is defined by a *form*, which comprises of a number of *fields*. A field can contain both structured and unstructured content. (The text of a message for example, is unstructured.) Notes supports rich text and multimedia content in the documents. A full text search engine is also integrated to support search mechanisms.

Besides general solutions for group communication, Notes also provides a fairly extensive environment for application development. At the programmer level, Notes provides an API which allows application developers to access system level services. Notes also provides high level support for languages like VisualBasic and C++, which reduces the complexity of working directly with the API. In recent release (R4), Notes has also provided a new tool called LotusScript, which is a BASIC compatible programming environment for accessing Notes services. Other programming features include *agent builders* which allow developers to automate application tasks and *actions*, which allow developers to present users with an "action-oriented" user interface. A powerful development environment allows programmers to create third-party Notes applications and customize the Notes system for enterprise specific needs.

### 3.4.2 Naming and Storage

Like FirstClass, Notes uses a logically centralized repository for storing messages. There may exist more than one cooperating Notes servers. Documents are stored in databases called *message stores.* Notes also has support for accessing external databases. A document is created by filling in information on its form which the database contains. Documents in a database can be categorized into *folders.* A document may be included in more than one folder. From the user perspective, documents are identified by a tuple which consists of the name of the database and the document. For supporting high availability, Notes also provides a fairly flexible mechanism for replicating parts of a database.

### 3.4.3 Navigation and Browsing

Notes provides a proprietary client for accessing the databases. Information presentation is built on the concept of a *workspace.* Users may categorize the databases into various workspaces. Navigation across workspaces is done through workspace tabs. Each database is represented by an icon in the workspace. Shortcuts for common commands are provided by a set of *smart icons.* While displaying the contents of a database, Notes uses *panes*, which are akin to frames in Netscape.

Documents are presented to the user in the form of *views.* A document view consists of a list of all the field names of the document. For example, if the user wishes to look through documents by date, Notes presents a view of the documents sorted by the values entered into the "date" field. Other field information is listed in columns to the right. Notes views are flexible, employing an outline-like, expand-and-collapse metaphor. For example, if a parent document has multiple child documents, the user can elect to view just the parent, or the parent and all next-generation documents, or all the generations of documents related to the original parent.

### 3.4.4 Communication Models

Notes provides a powerful environment for communication and collaboration. Support for 1-1 communication is provided by an integrated messaging system which incorporates the `cc:Mail` [Cor96a] interface. Group communication is supported by *discussion groups*.

One of the major drawbacks however is that the system is not based on open standards, and thus, is not suitable for internet applications. Users are required to obtain proprietary clients for accessing the system. Information is also stored in Notes' database format and cannot be easily integrated with other formats. (Third-party tools for database integration are available at extra cost.) Recently, Notes began providing support for web documents. Notes documents can be converted to HTML for presentation on web clients and the Notes server can also accept HTTP requests [Cor96b]. Efforts are being made to provide all the web services through Notes server. However, since the web itself is in a state of constant flux, it may not be possible to incorporate the new developments by merely extending the basic architecture of Notes.

## 3.5 Web based systems

Conferencing systems built on web technology began appearing in early 1994. The basic idea was to develop conferencing systems as web applications, and delegate the user interaction to the web browser. Web conferencing systems act like *gateway programs*, and interface with the client through the HTTP server, which also handles requests for other documents. The interaction between the HTTP server and such gateway programs is defined by Common Gateway Interface, or CGI [Tea94]. CGI specifies how data are sent to the gateway program, and how data can be returned by the gateway program, through the HTTP server and back to the client. The CGI specification is independent of the underlying HTTP server and any programming language.

In this section we describe some of the popular web based conferencing systems. Most of these systems however are still in the process of development and we provide condensed descriptions of their distinctive features.

### 3.5.1 WIT: W3 Interactive Talk

WIT [Luo94] is a forms based discussion system which was developed following the WWW94 conference in order to allow discussions on W3 technical matters to be stored in a structured format. WIT is a highly structured system and discussions are modeled using a hierarchical tree. The tree may have arbitrary depth, but the top three levels of the hierarchy have specific purposes and are named as:

- **Topic** – an issue to be resolved
- **Proposal** – a statement proposed for discussion,related to the topic
- **Argument** – an argument (also called "article") for / against a proposal

Arguments may be nested arbitrarily. Each argument is tagged by either a check-mark or an "X" to indicate whether it is for or against the proposal. Arguments are thus constrained to be either for or against the proposal and cannot simply put forward a new idea. Secondly, once a topic is set, it is not possible to split the discussion into subtopics. WIT was implemented rather quickly, and a number of required features were omitted. Activity tracking for users was not supported and navigation between messages was rudimentary. Furthermore, the basic design of WIT did not make it suitable for general purpose conferencing.

### 3.5.2 HyperNews

HyperNews [LaL95] was developed along with WIT and was one of the first systems to appear. HyperNews also uses a hierarchical tree structure like WIT, and consists of two types of messages: base articles and responses. A discussion is initiated with a base article and replies to the base article form a sub-tree of responses. The base article is a normal HTML document and can reside anywhere on the web. The hierarchy of responses is mapped directly onto the physical storage. HyperNews supports messages in plain text as well as HTML. Users can associate an icon with the message title to indicate its "role" (idea, feedback, question, note, etc.) in the discussion.

### 3.5.3 Threads

Developed in late 1994 by HotWired [Hot96], this is the first system built for supporting a commercial site. Unlike other systems, Thread presents each discussion in a single document instead of a collection of documents. Each discussion consists of a *topic* followed by a number of responses. Unlike previous systems though, Threads does not organize the discussion into a hierarchy. Responses are listed sequentially after the topic. Threads is one of the first web based conferencing systems to present a discussion as a continuous flow. Since the entire discussion is served in a single document, reading a set of message becomes much more efficient. However, it is not possible to obtain different views (such as sorting by author, subject, etc.) of the discussion. Since it displays the entire messages, obtaining an overview of the discussion can be difficult. Furthermore, Threads does not keep track of users' reading activities but supports filtering of messages on the basis of their time of creation.

## 3.6 Summary

We have described a number of systems which are currently being used for group communication in online education and other fields. While some of these systems were specifically designed for group communication (or contain group communication as one of the components), others were extended to support it. In the following table we compare the various systems in terms of the desired elements of conferencing described in section 2.3 [1].

[1] A comparison of the Web based systems is presented in Chapter 6.

| Features | **E-mail** | **USENET** | **FirstClass** | **Lotus Notes** |
|---|---|---|---|---|
| **Original target application** | Point to point communication | Group discussions | Interpersonal and group communication | Groupware (Communication, collaboration, and coordination) |
| **Communication Model** | 1–1, 1–n (*mailing-lists*) | 1–n | 1–1, 1–n | 1–n |
| **Structuring of communication** (threading, etc.) | NO | YES | YES | YES |
| **Flexible access control** | NO | Unrestricted read access, posting can be controlled by moderator | YES | YES |
| **Activity tracking** | Depends on MUA | Depends on newsreader | Limited | Limited |
| **Support for evaluation tools** | NO | NO | NO | Support for development |
| **Interoperability** | Depends on MUA | NO | NO (Standalone system) | Limited |

The effectiveness of electronic mail as a group communication tool mostly depends on the MUA which is being used. For example, there is no implicit support for structuring of messages. However, the MUA can generate cross references by analyzing the set of messages in the user's mailbox. Similarly, the MUA can track which messages have been already read, etc. USENET also separates presentation from transport. However, while a large number of MUAs are available for electronic mail, the number of USENET clients is limited and less varied. Recently, Web clients have begun providing support for presenting electronic mail and accessing USENET news. Nonetheless, most of these systems do not provide support for evaluation of messaging activity, which is an important aspect of online learning. While third party applications can be built for providing evaluation in the case of Notes, storing data in proprietary formats and using proprietary protocols make it too limiting for an open, internet based environment. In the next chapter we describe the design of our system, which attempts to address the specific needs of educational conferencing and eliminate some of the problems associated with these systems.

# Chapter 4

# Design of Virtual Groups

## 4.1 Evolution

The earliest version, known as **ClassTalk**, was developed in the summer of 95 as the first prototype of the Virtual–U conferencing tool (see Figure 4.1).

The basic unit of interaction in ClassTalk was a *message*. Users created messages to either initiate a new conversation or to participate in an ongoing conversation. Messages related to each other were grouped into discussions. Discussions could be further grouped into directories. Directories were hierarchical in nature and could contain sub directories. They provided a convenient way to organize a large number of discussions. The hierarchy of directories was directly mapped onto the physical store (Unix file system). Names of discussions and directories corresponded to file system directories. Messages of a discussion were stored as text files in the directory corresponding to that discussion. Messages which initiated new conversation were assigned numbers sequentially. Other messages, i.e., which contributed to an ongoing conversation were assigned numbers using the dot-notation [1] based on their position in the thread of conversation. Thus, on a system wide level, a message could be identified by a tuple consisting of the file-path relative to top level and its number. Some of the popular web based conferencing systems, like HyperNews [LaL95] for

[1] Message numbered 1.2.5 for example, was the fifth reply to the second reply of the first message.

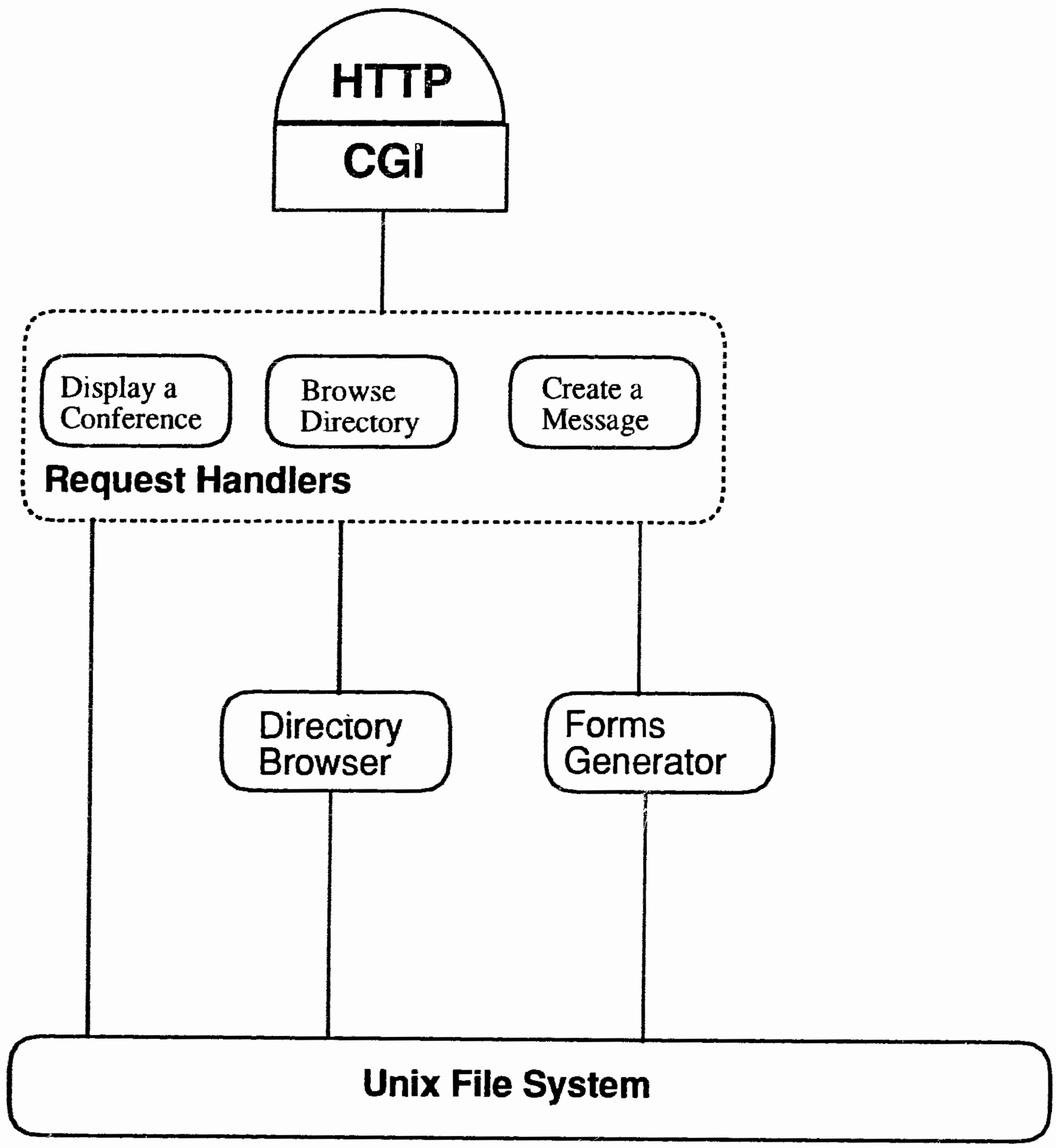


Figure 4.1: ClassTalk Server Architecture

example, still continue to use a similar naming scheme.

Access control in ClassTalk was based on the underlying HTTP server's access control mechanism. The set of people who could access a discussion was specified in the `.htaccess` file for the corresponding Unix directory. ClassTalk did not provide multiple levels of access for a discussion and so there was no provision for example to allow only read privileges. The system administrators organized the users into groups and there was no support for creating personal groups. ClassTalk supported activity tracking on a per-user basis and user profiles were maintained on the server. Although inconvenient and limited in functionality, ClassTalk served as a useful mechanism for communication in the early days of Virtual-U's web based conferencing.

A major step in the evolution of the conferencing system was the recognition that users would like to set up their own groups for conversation. This implied a need for greater flexibility with respect to access control. While direct mapping of discussions to the physical storage reduced the access time, it also imposed several limitations on naming and organization. Furthermore, to analyze a discussion, instructors wished to obtain different views of the set of messages, such as sorting by author, date, etc. These, and several other considerations led to the redesign of the conferencing system and the development of Virtual Groups.

## 4.2 Design of Virtual Groups

Based on the client-server parad'ɜm, Virtual Groups, like its predecessor, uses the web server and the HTTP protocol [BL92] for communication with clients. Figure 4.2 illustrates the layered structure of the complete system. The server includes a conference manager, an authenticator, a group manager, a profile manager and set of tools for activity logging (see Figure 4.3). The front end accepts requests through links or forms provided by the web client [2], and, after pre-processing (which includes access authorization, etc.) invokes an appropriate request handler. The request handler executes the request and ships the resulting documents (or fill-out-forms) back to the client. All activities are logged, both by the Web server and the log manager.

[2]The current version has been customized for Netscape Navigator

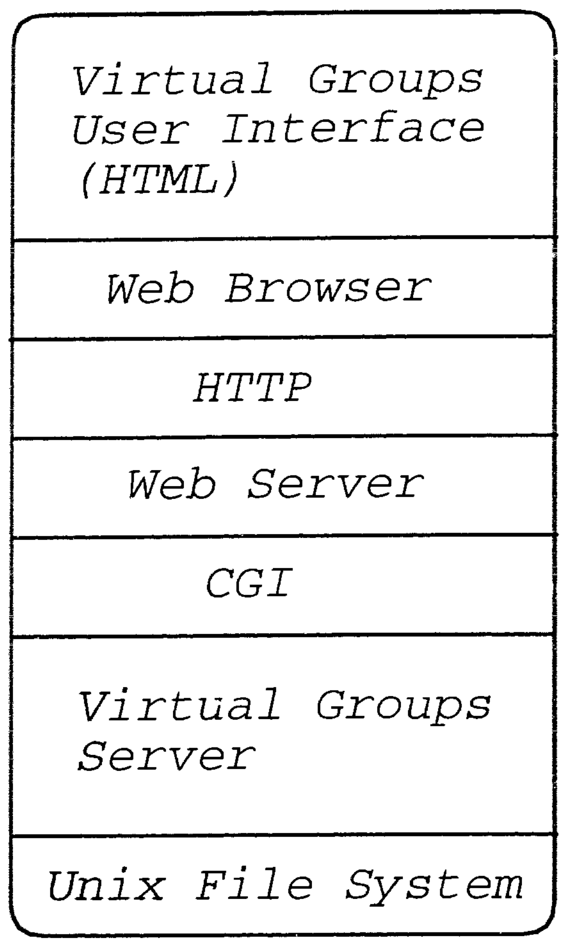


Figure 4.2: Virtual Groups layer structure

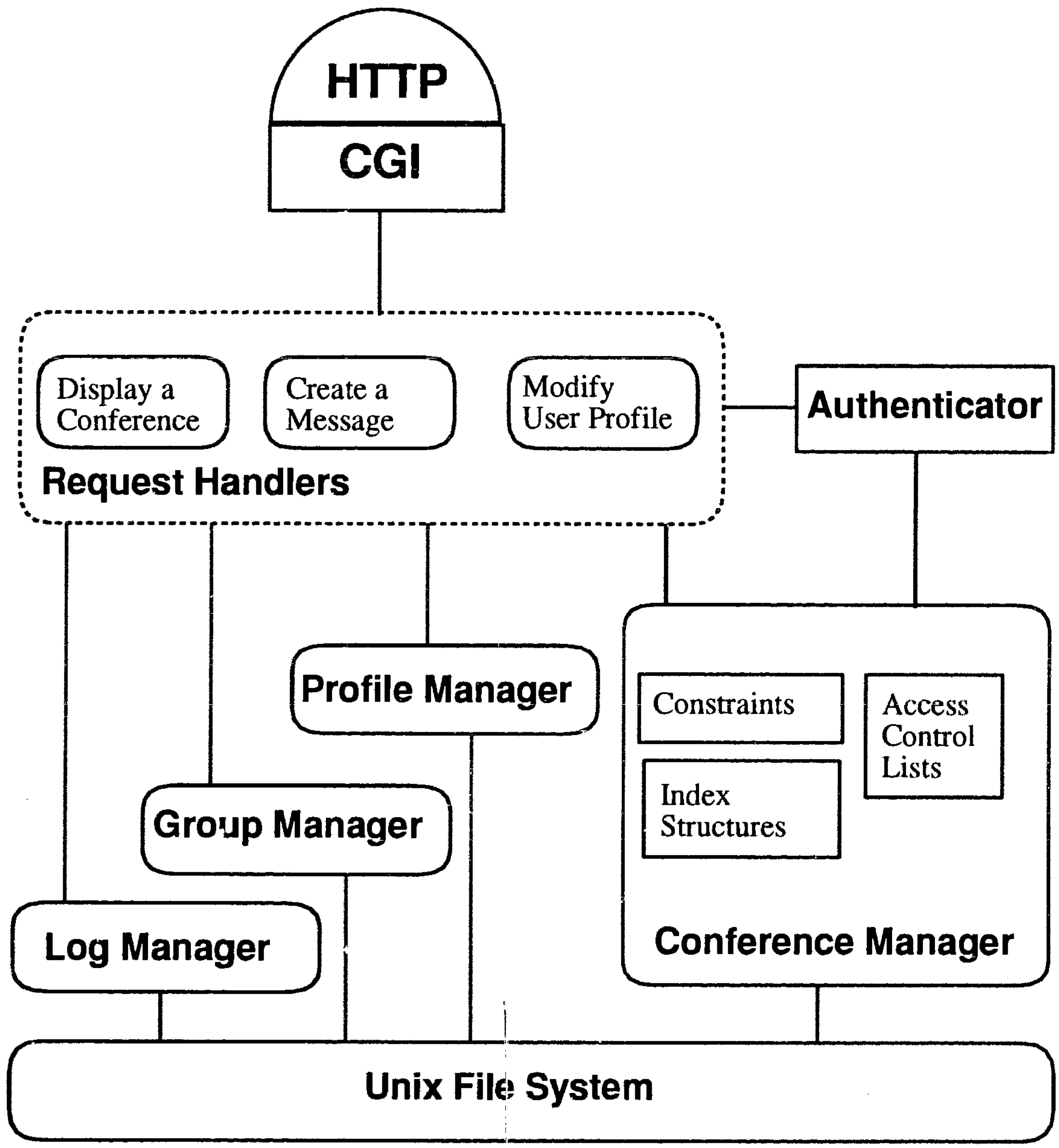


Figure 4.3: Architecture of Virtual Groups Server

As in ClassTalk, the basic unit of interaction is still a message. However, discussions and directories are merged into a single concept called *conference.* Conferences can nest and so a conference usually contains sub conferences and messages. the system allows the users to create their own *groups.* See Figure A.1 for an illustration of abstractions in ClassTalk and Virtual Groups.

### 4.2.1 User Interface

The user interface is an important part of the conferencing system. Rather than being confronted with a new user interface, users prefer to access the system by using popular and familiar services, such as e-mail and Web clients. Therefore, instead of creating yet another graphical interface for the HTTP protocol, we have chosen to support user interaction through Netscape, a commonly used Web client. This also facilitates seamless integration of the conferencing system with other course resources which are accessible through the Web. After "logging in", every user gets a customized view of the conferencing system. Interaction with the system occurs in two basic ways:

1. By following a hyperlink, which contains the encoded request in the URL (for example, name of a conference may be a link to the request for displaying the conference)
2. By filling out a form (for example, while submitting a new message).

The interface is organized in a number of *screens* Since the content of a screen (e.g., list of messages, body of a message) could be larger than screen size, it is possible that the navigation links could scroll off the screen while viewing. To prevent this, the screens are divided into a number of frames (See Appendix A for the various screen shots).

**Main Screen:** This is the top level screen in Virtual Groups and displays a list of conferences which the user has joined (see Figure A.2). The number of unseen messages and total number of messages are also shown with the conference.

**Conference Screen:** This screen displays the contents of a conference: i.e., the list of messages and sub conferences if any (see Figure A.3). Users can choose to view either all messages or only the unseen messages, and this can be set on a per-conference basis. The list of messages can be sorted by author, date or thread, thereby providing different views of the conference. The system retains the choices of the user in a profile and will display the appropriate view when the conference is visited the next time. An interface to a search engine is also provided and users can perform a keyword search on the full text of messages.

**Message Screen:** This screen shows the actual content of a message (see Figure A.4). Currently, Virtual Groups supports three different formats of message type: plain text: when the message composed by the user is displayed verbatim (URLs are automatically converted to hyperlinks), HTML: where the message is assumed to be marked up with HTML tags, and URL: where the actual message can be located at a different site.

**Other Screens:** For creation of conferences and messages, the system provides a number of form based screen interfaces.

### 4.2.2 Conference Manager

Virtual Groups uses the store-and-forward technique for supporting asynchronous communication. All messages are stored on the server. The conference manager provides a mapping between the logical organization of the messages and the physical structure of the persistent store. Since contents of the messages can be HTML documents, users may include URLs to external objects. In such cases the system does not copy the external resource on the server but only keeps the referring link. In the present implementation, Unix file system is being used as the persistent store. Messages are stored in files and conferences are modeled using directories. The system also maintains indexes which are used to provide different views during browsing and tables which contain the relationships between messages.

One of the drawbacks of current Web based systems is that they tend to provide a single communication model. In an educational environment however, different instructors may wish to structure the communication in different ways. The instructors who used the previous version (ClassTalk) for example, wanted to organize the communication in terms of directories and discussions. Other instructors preferred to organize communication as nested 'conferences'. To support multiple metaphors, the conference manager provides the notion of **constraints**. By adjusting the appropriate constraint values, users can set up desired structures of communication. Constraints are also used for imposing other restrictions on a conference, such as limiting the depth of discussions, size or type of messages permitted. Furthermore, users can also control access to a conference by specifying the times when the conference will be 'active'.

### 4.2.3 Authenticator

Access validation is carried out in two steps: 1. general user identification and 2. request specific authorization. (Currently, the system does not support data encryption.) Unlike the Telnet and FTP protocols, HTTP is not session based. Hence, every request from the client requires validation. Moreover, other course resources also require the general user identification procedure. To provide a consistent authorization scheme for the users, the conferencing system (and other course tools) delegate the responsibility of user identification to the HTTP server. Each user is assigned a user id and a password. The browser encodes these using the 64 bit encoding [BL92] and sends to the server for validation. Once a request is validated, the browser caches the encoded string for possible future requests.

For request specific authorization, the system uses access control lists on a per conference basis. The moderators of a conference can associate different lists of users (or groups) with each type of request (READ, POST, etc.). Before processing, the privileges of the user are evaluated from the specifications in these access control lists.

### 4.2.4 Group Manager

To support flexible group communication, the conferencing system provides mechanisms for easy creation and maintenance of groups. A group may be defined in terms of users and other existing groups.

*For example, an instructor may divide a class into six lab-groups : L1 - L6 and later create three project groups, each consisting of two lab-groups: P1 (L1+L2), P2 (L2+L3) and P3 (L4+L5).*

However, since each request is subjected to the validation process, maintaining group definitions in terms of other groups increases the overhead in request validation and reduces the response time. Hence, group membership is determined at the creation stage only and subsequent changes to other groups will not affect the definition of the new group.

*In the previous example, subsequent changes to L1 will not cause any changes in P1.*

Users have their own name space for defining groups. Thus different users may use the same group name. On the system level, a group is identified by a tuple, which consists of the group name and user-id of the creator, thus disambiguating any name conflicts.

### 4.2.5 Profile Manager

Since each user sees the same copy of the message, users cannot delete the messages after reading. Instead, the system maintains a profile for every user and keeps track of the reading activity. This information is used in presenting subsequent views of the conferences, where the status of each message (read/ unread-old/ new) is displayed.

### 4.2.6 Request handlers

These are the set of routines that actually execute the user's request. Requests fall in two categories:

1. Hyperlink based requests, which fetch information from the system: navigation, display a conference, read messages view statistics, etc.

2. Form based requests, which add or modify the information in the system: create conferences or messages, modify conference properties, privileges, etc.

The request handlers interface the conferencing system with the HTTP server and are implemented as CGI programs [Tea94].

### 4.2.7 Log Manager

Although the Web server logs every request, it is not possible to derive statistics for conferencing activity, as the logs only keep information about "physical file accesses". This means that when documents are generated on the fly, the web server will log the name of the script that generated the document rather than the document itself. This can be ambiguous in the case of scripts that generate fill-out-forms (for example, create a new message). Secondly, logging by the Web server does not reflect the success or failure of the client's request. This necessitates the need for a high level logging tool that can record requests in terms of conferencing activities, such as number of messages read, posted etc.

## 4.3 Implementation

Presentation is delegated to the web browser. However, no changes to the browser are needed as data is marked up in HTML. The browser communicates with the web server which in turn interacts with the request handlers. Virtual Groups uses standard CGI instead of server-side includes or proprietary server APIs and works with most servers. The entire system is written in Perl v4.36 and can work with little modifications on any Unix system.

# Chapter 5

# Evaluation

Though the conferencing software has only been available for use at Simon Fraser University classrooms for eight months at the time of writing, some of our expectations about the feasibility of the web have been given significant testing. Five SFU instructors have used VGroups for various activities in their courses, ranging from supplementing a face-to-face course with some online discussions, to offering a fully online course (with face-to-face introductions in the first week). Cognitive and motivational factors play an important role in the adoption of the system [Ori92]. Using Virtual Groups was made mandatory for the students, and part of their course grade was based on participation in online communication. The Virtual–U research team also used the system for its group discussions over development issues. During this time, a total of about 210 users worked with 170 conferences ( an additional 54 conferences were created while testing the system) and generated 3107 messages. Figure 5.1 shows the distribution of messaging activity by hours of the day. Since the system is available through the Web, students were able to connect at all times and the messaging activity is uniform during the "work hours" for students. (The peak at 3:00pm represents activity during special lab sessions conducted by the instructor). Although significant part of the course was conducted online, the pattern of regular courses was still followed. Figure 5.2 shows the distribution of messaging activity during various weeks of the semester, which confirms with the pattern for a regular course. The maximum number of messages for a single conference was 187, in the social conference *Café*. Of the 3107 messages, 1847 were written in plain text, while 1238 were marked up using HTML (22 messages were references to objects on other sites).

Special conferences were set up to obtain feedback about the system from the users. A number of useful suggestions, bug reports, etc. were received through these conferences. Our expectations that a web based system will help students participate more effectively and in an unconstrained manner were partially validated. While web technology is promising, there are a number of issues which need to be addressed before it becomes an effective medium. These issues can be broadly classified into two areas: performance, and HCI. Networking protocols currently used were not originally designed for the type of traffic which is generated in conferencing activities and so affect the performance of the system. The user interface provided by the web browser, though simple and easy to use, is not powerful enough to replace the rich interactivity of a traditional client like FirstClass or Lotus Notes. In the following sections, we outline some of these issues and describe how they affect the overall system usage. Pantel et al [PWWW96] present an alternate approach to evaluation of the system based on user observation and other evaluation strategies.

## 5.1 Performance Issues

### 5.1.1 HTTP Performance Problems

The HTTP protocol was originally designed for rapid and effective transfer of digital documents. While efficient for simple document transfer, this protocol does not serve well the needs of a conferencing system. Firstly, HTTP has an extremely simple model; the client establishes a connection to the remote server and issues a request. The server then processes the request, returns a response, and closes the connection. Thus each request entails a separate connection between the client and the server. So if a screen has multiple frames, each frame is accessed through a separate connection.

Secondly, HTTP uses TCP [Pos81] as a transport layer, and some design features

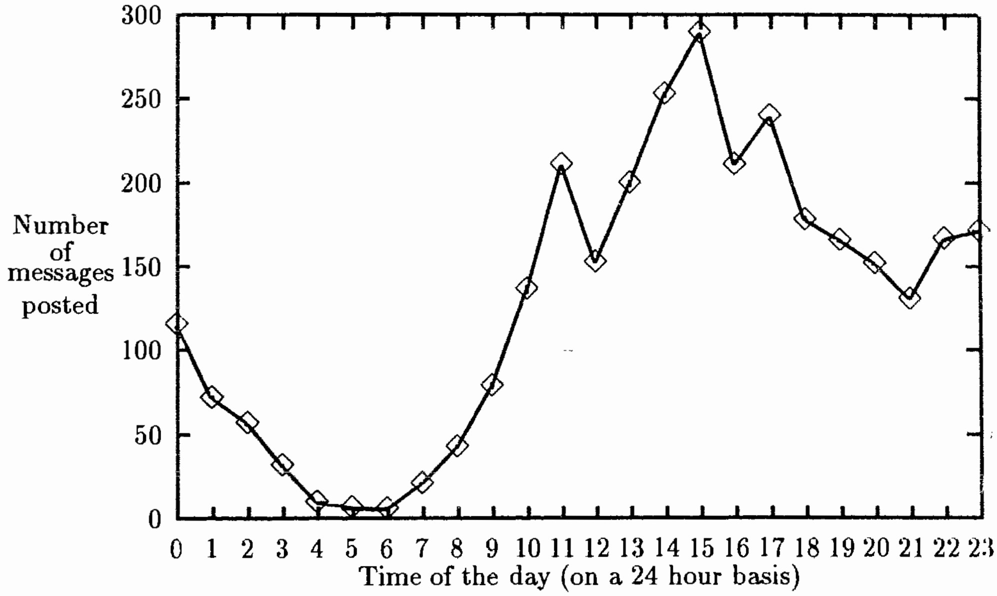


Figure 5.1: Distribution of messaging activity by hours of the day.

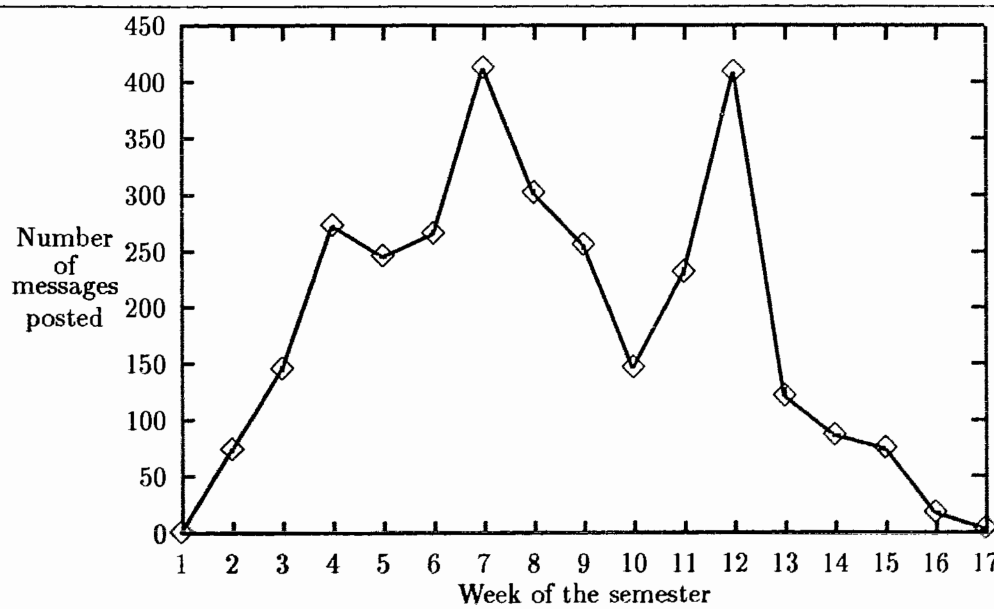


Figure 5.2: Distribution of messaging activity during different weeks of the semester.

of HTTP interact badly with TCP, resulting in low performance and reduced throughput. While initiating a connection, TCP adopts a more conservative approach and uses a *Slow Start* [Jac88] for transmitting data. While this approach may be ideal for long lived connections like Telnet and FTP, it has an adverse effect on short lived connections like those used in HTTP. Furthermore, when a server closes a TCP connection, it is required to keep information about that connection for a period of time, to accommodate any delayed packets. This extra wait time also reduces the throughput of the system. Spero [Spe96] discusses in detail the various problems associated with the current version of the HTTP protocol.

### 5.1.2 CGI Mechanism

Traditional systems like FirstClass and Notes have a server process which is dedicated to and interfaces directly with the clients. The web server forks a CGI process for every request. This mechanism of handling client requests has some significant drawbacks. Although users tend to interact with the system in long "sessions", requests are processed independently of each other, and a new CGI process needs to be created for handling each request. If a user for example, reads 10 messages in a conference, every request will require authentication, access evaluation, opening and closing of relevant databases, and writes on the disk, leading to a significant drop in performance.

### 5.1.3 Network Bandwidth

While ability to access the system from anywhere is certainly an appealing, scarcity of bandwidth does not allow the users to take full advantage of this feature yet. Users who dial in from homes for example, are often unable to access the system properly.

### 5.1.4 Access Authentication

Currently, the conferencing system (and other Virtual-U tools) delegate user validation to the underlying HTTP server. The server [1] stores the user name, and password

[1] NCSA HTTPD, version 1.4

tuples in a text file and does a sequential search for every request. With a large user base, this causes significant delay in validating a request. Since each request needs to be validated, the authentication process has an impact on the overall performance of the system.

## 5.2 HCI Issues

### 5.2.1 System Feedback

Currently, the system does not support any computing on the client side. This implies that client's every action needs to be transmitted to the server for evaluation. Delays across the network compounded with statelessness of the server can result in unacceptable response times for simple actions, like previewing a message before submission.

### 5.2.2 Navigation

The conferencing system provides a set of hyperlinks on every screen to support navigation. By following an appropriate link, users can not only access various parts of the conferencing system but also other resources in Virtual-U. The web client also provides a set of buttons (called *Back* and *Forward*) and a "Go" menu for navigating among the recently visited pages. Thus, while the conferencing system provides options for "logical" navigation of the system space, the web client provides options for "physical" navigation of the recently accessed documents. These options do not blend well and inexperienced users are often confused and unable to navigate in a consistent manner.

### 5.2.3 Message Formatting

One of the salient features of the web, availability across all platforms, is also a cause for certain deficiencies in the system. Need for multi-platform presentation implies that documents be either composed in plain text format (the lowest common denominator across platforms), or encoded in a language which is supported by all clients, e.g., HTML. The original design goal of HTML was to specify content, and presentation was delegated to the client and so it does not provide support for rich formatting of documents. Users are therefore constrained to compose their messages with limited formatting. Unfortunately, even the limited formatting was available only to those who were familiar with HTML.

### 5.2.4 Interoperability

There are several occasions where users wish to process the messages with other tools. Instructors for example, would like to extract a subset of messages for analysis. Currently, web browsers provide a limited support for text editing. Furthermore, for security reasons, it is generally not possible for the web browser to inter-operate with other tools (such as editors, other applications, file systems, OS) on the user's machine. The user is thus unable to manipulate messages with native tools without incurring significant overheads. For example, using a WYSIWYG HTML editor for composing messages required a number of steps before users could actually submit the message.

### 5.2.5 Browser Stability

The web browser itself is under development and is not a robust tool. Several users had problems with the browser that resulted in unexpected crashes, errors, and other system faults. Although this was due to the problems in the browser, it affected the overall usability of the system.

### 5.2.6 User's Resources

Although systems built on web technology are platform and architecture independent, limitations of the user's hardware do affect the presentation of the interface. Our current design which uses multiple frames in screens has met with mixed success. Frames were intended to provide appropriate context while reading a message and enhanced the presentation on large screened monitors. On low resolution screens and smaller monitors however, frames based interface adversely affected the usability of the system and required the user to do extra work of scrolling and locating relevant links in individual frames.

# Chapter 6

# Conclusions and Future Work

The work described in this thesis has presented the development and implementation of the early version of the conferencing system for Virtual–U. Unlike other web based systems, this system addresses issues specific to electronic conferencing for online education. A prototype was implemented and used for two courses in Fall of 95. Feedback from the users and analysis of its performance lead to the development of Virtual Groups. The system was built as a gateway application to the web server and supported interaction via the Netscape browser. Group communication has been modeled using conferences. The system supports threaded discussions and provides multiple views. Users are allowed to create their own groups for discussion. A flexible mechanism for access control has been implemented which supports multiple levels of privileges for users. Tracking of user activity has been supported on a per-user basis. The following table compares Virtual Groups with other Web based conferencing systems.

| Features | **WIT** | **HyperNews** | **Threads** | **Virtual Groups** |
|---|---|---|---|---|
| **Communication Model** | 1–n | 1–1, 1–n | 1–n | 1–1, 1–n |
| **Structuring of communication** (threading, etc.) | Fixed format of "arguments". | YES | Fixed (linear string) | YES |
| **Flexible access control** | NO | NO | NO | YES |
| **Activity tracking** | NO | Limited | NO | YES |
| **Multiple Views** | NO | YES | NO | YES |
| **Support for evaluation tools** | NO | NO | NO | Yes |
| **Interoperability** | NO | Under construction (E–mail and USENET) | NO | Limited (With other Virtual–U tools) |

Early results obtained through special feedback conferences are encouraging and have partially validated our expectations of an open, web based system. The system was available at all times and users could access it from any location. Users found it much easier to interact through a web browser than via special client software. Referencing of messages via hyperlinks was more intuitive than traditional methods. Students were able to integrate other course material and even external references easily in the conferences. The social conferences provided an excellent place for non academic interaction and the volume of messages overwhelmingly confirmed the importance of such spaces in the educational framework. Continued usage of the system after the end of the 13 week semester (see Figure 5.2) strongly suggests that after a sustained use of the system, users were able to adopt is for their regular, informal communication.

However, since the web is itself under development, the system had to cope with the limitations of the available technology and did not compare well with proprietary conferencing systems. The limitations affected the performance and overall usability of the system. Some user actions, which were trivial in traditional conferencing systems, turned out to be awkward or even impossible. Users were not able to fully exploit the multimedia nature of hypertext because of severe bandwidth constraints. Due to lack of an appropriate editing environment, message composing was rather restrictive. Use of HTML, for example, was limited to people who were familiar with the language.

## 6.1 Future Work

This thesis has helped researchers and developers understand the issues involved in building a web conferencing system and has identified a number of challenges in web based design. However, web technology is still in a state of flux and new concepts, features are being continuously developed which not only relax some of the limitations, but also provide better ways of doing current work. In this section we list some of the enhancements that are being planned for the next version.

**Persistent Service** Request handlers which interface with the web server could be made persistent processes so that startup cost could be eliminated for multiple requests by a client. This will allow the system to maintain state for various requests and process long sessions efficiently.

**Offline operations** Most users are yet to obtain high bandwidth connectivity for remote locations for large periods of time. In some cases, the users are also charged for connectivity on the basis of usage. It is therefore desirable that the system should provide a method for offline reading of conferences. This may entail development of supporting tools which could run on client's machine.

**E-mail Gateway** Currently, the conferencing system serves as a "closed" environment for discussions. However, it is often desirable to integrate electronic mail messages with the messages of the conferencing system.

**Multiple views** Currently, the system provides a limited number of ways in which a conference can be viewed. Often, users like to focus on certain set of messages in a conference (for example, messages created in a specific period of time, or by certain users, etc.). In the case of weak connectivity, users may prefer to download all the messages in a single document instead of accessing them one by one. The system should allow the users to specify how the set of messages in a conference be presented.

**Database Support** All the data is currently stored in standard text files. However, as the number of users increases, file access is likely to affect the overall performance of the system. The next version will use databases for storing frequently accessed information, such as user profiles, access control specifications for a conference, etc.

## 6.2 Summary of Contributions

One of the goals of the thesis has been to provide an understanding for educators of the various issues in using the Web as a medium for online education. The objective was not to develop a full fledged solution, but to expose the challenges and pitfalls of adopting Web as the medium. The major contributions of this work can be identified as:

- Analyzed issues which affect the viability of Web.
- Developed a working system through iterative design which supports:
  - Flexible structuring of communication
  - Easy creation of groups
  - comprehensive activity tracking

The Web seems to offer many exciting, new possibilities for deploying networked learning environments. By developing the system and using it for offering actual courses, we have tried to compare how the capabilities of current state of the art in Web technology match up with the expectations of educators. While the results seem promising, it is clear that there is still a need for improvement before Web based applications become comparable to traditional applications in terms of robustness and quality of service. We hope that this thesis will serve as the basic groundwork for developing future systems.

# Appendix A

# UI Screens

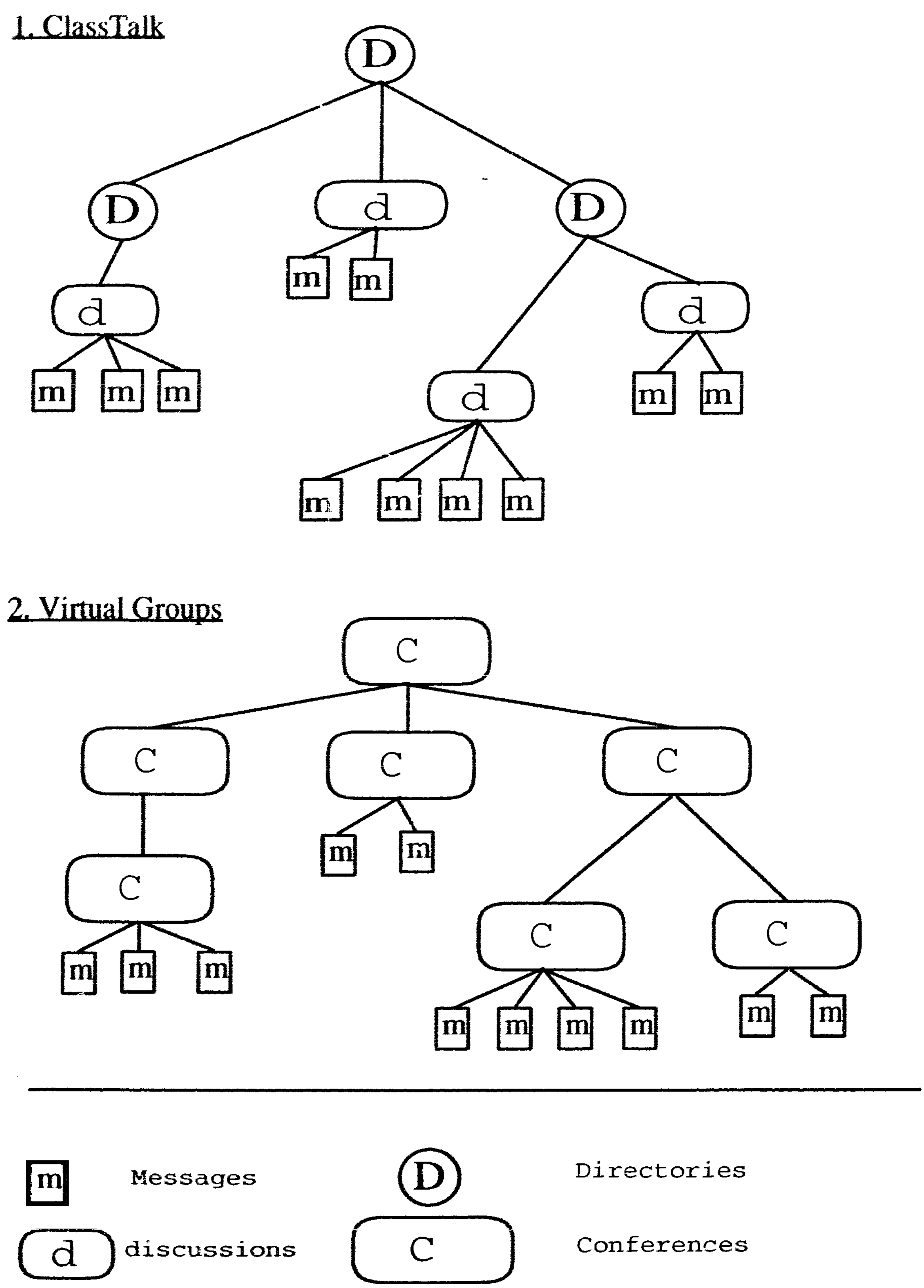


Figure A.1: Communication Structures in ClassTalk and Virtual Groups

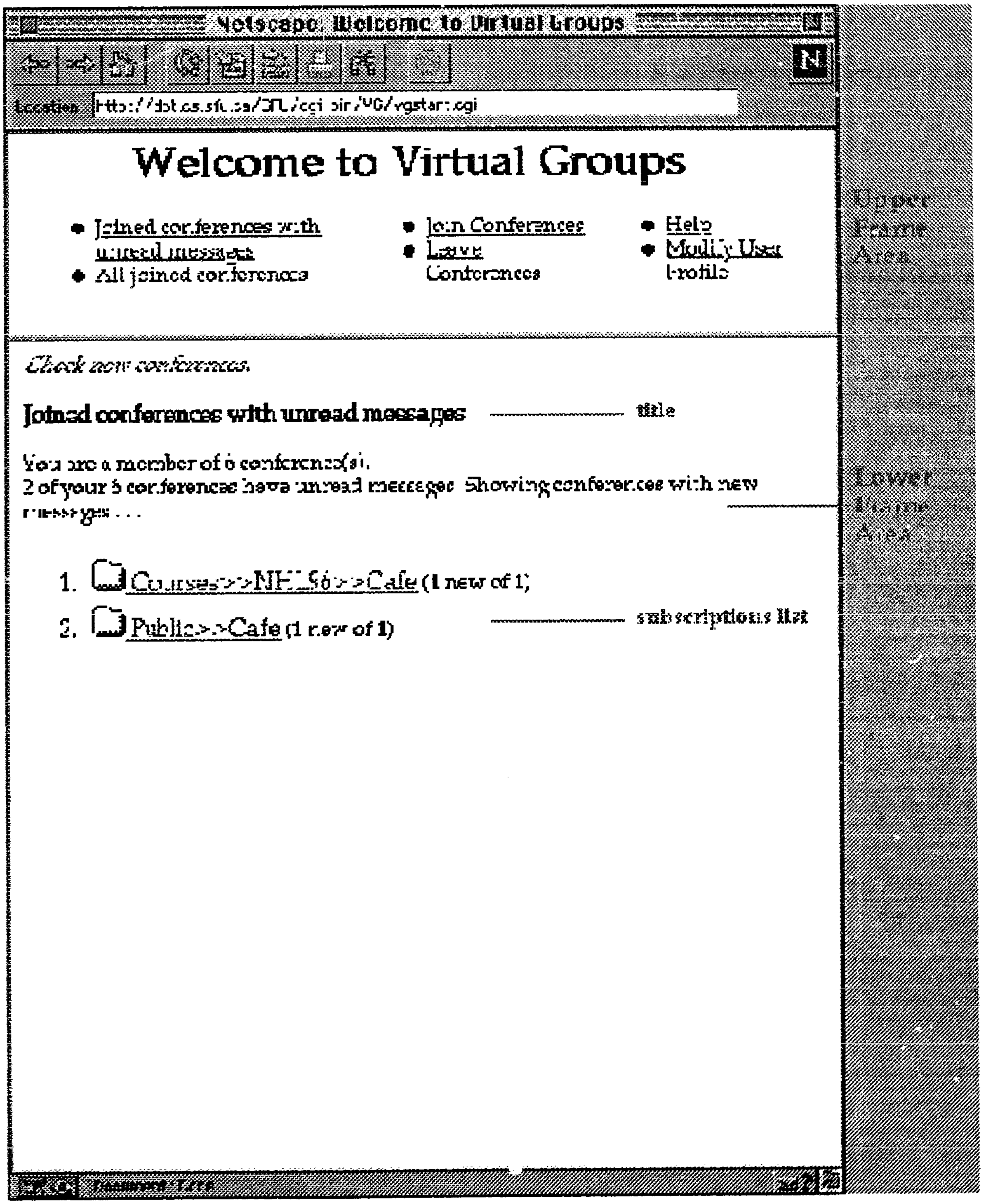


Figure A.2: Top Level Screen

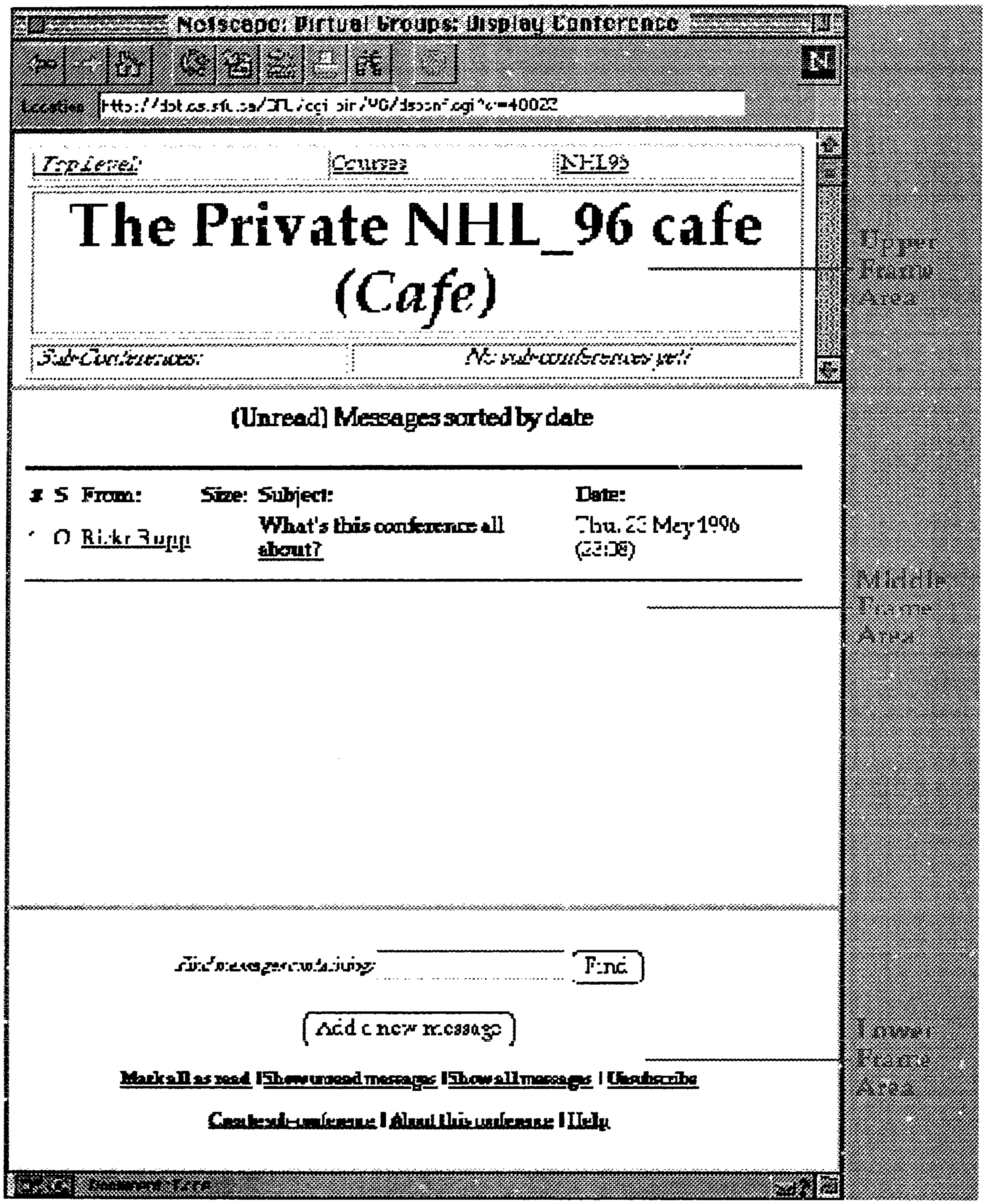


Figure A.3: Conference Screen

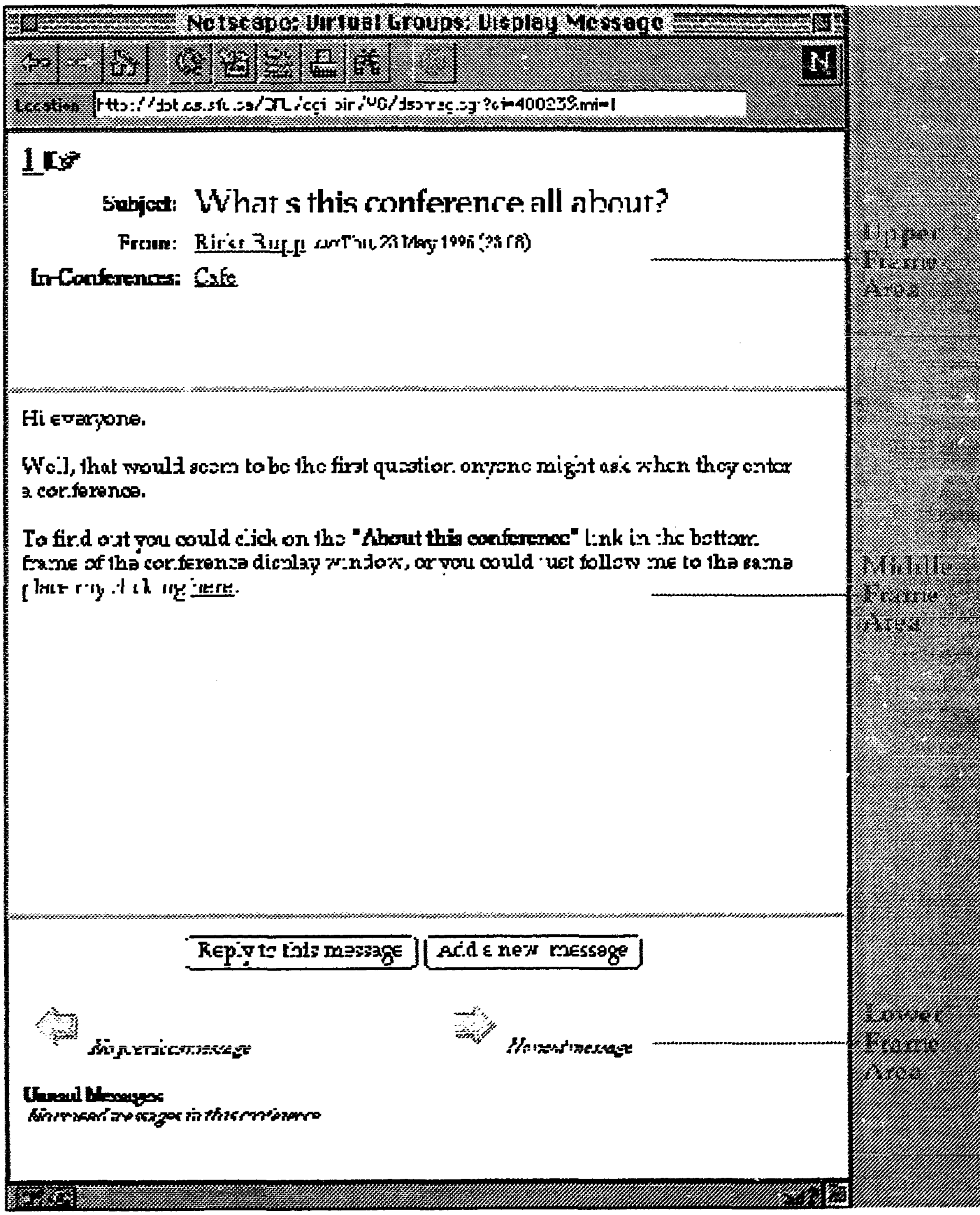


Figure A.4: Message Screen

# Bibliography


[BL92] T. Berners-Lee. Hypertext protocol requirements, internet working draft, w3 consortium. (Available on the world wide web as http://www.w3.org/hypertext/WWW/Protocols/HTTP.html), 1992.

[BLC93] T. Berners-Lee and D. Connolly. Hypertext markup language: A representation of textual information and metainformation for retrieval and interchange. (Internet Working Draft, W3 Consortium. Available on the world wide web as http://www.w3.org/hypertext/WWW/MarkUp/HTML.html), 1993.

[BLCGP92] T. Berners-Lee, R. Cailliau, J. Groff, and B. Pollermann. World wide web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1(2), 1992.

[Bur88] L. Burge. Beyond andragogy: Some explorations for distance learning design. *Journal of Distance Education*, 3(1):5–23, 1988.

[Con87] J. Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 2(9):17–41, 1987.

[Cor96a] Lotus Development Corporation. cc:mail. (Available on the world wide web as http://www.lotus.com/comms/ccmail.htm), 1996.

[Cor96b] Lotus Development Corporation. Lotus internet strategy. (Available on the world wide web as http://www.lotus.com/corpcomm/340a.htm), 1996.

[Cor96c] Netscape Communications Corporation. Extensions to html. (Available on the world wide web as http://www.netscape.com/assist/net_sites/html_extensions.html), 1996.

[Cro82] D. M. Crocker. Standard for the format of arpa internet text messages. (Network Working Group, RFC 822. Available on the world wide web as ftp://ds.internic.net/rfc/rfc822.txt), 1982.

[EPG96] D. C. Edelson, R. D. Pea, and L. M. Gomez. The collaboratory notebook. *Communications of the ACM*, 39(4):32–33, 1996.

[GM95] J. Gosling and H. McGilton. The java language environment: A white paper. (Available on the world wide web as http://www.javasoft.com/whitePaper/java-whitepaper-1.html), 1995.

[Har89] L. M. Harasim. Online education: A new domain. In *Mindweave: Communication, computers, and distance education*, pages 50–62. Oxford: Pergamon Press, 1989.

[Har90] L. M. Harasim. Online education: An environment for collaboration and intellectual amplification. In *Online Education: Perspectives on a New Environment.* Praeger, New York, 1990.

[Har93a] L. M. Harasim. Collaborating in cyberspace: Using computer conferences as a group learning environment. *Interactive Learning Environments*, 3(2):119–130, 1993.

[Har93b] L. M. Harasim. Networlds: Networks as social space. In *Global Networks: Computers and International Communication.* MIT Press, Cambridge, 1993.

[Har94] L. M. Harasim. The canadian telelearning nce. In *New Media Learning Technologies: Perspectives on Developing an International Collaboration for Flexible Distance Learning.* 1994.

[Har95a] L. M. Harasim. From minutes of virtual–u planning meeting in may 1995, 1995.

[Har95b] L. M. Harasim. The virtual university: New approaches to higher education in the 21st century. Keynote Address to the Australian Society for Computers in Learning in Tertiary Education Conference (ASCILITE'95). Melbourne, Australia., 1995.

[HCG96] L. M. Harasim, T. Calvert, and C Groeneboer. Virtual–u: A web–based environment customized to support collaborative learning and knowledge building in post secondary courses. *Proceedings from International Conference on the Learning Sciences (ILCS) '96, Northwestern University, Chicago, Illinois, U.S.A.*, 1996.

[Hot96] HotWired. Hotwired network. (Available on the world wide web as http://www.hotwired.com/frontdoor/), 1996.

[HR94] L. M. Harasim and M. Riel. Research perspectives on network learning environments. *Journal of Machine–mediated learning*, 4(2,3), 1994.

[HT94] L. M. Harasim and L. Teles. Developing the virtual interactive environment for workgroups. In *New Media Learning Technologies: Perspectives on Developing an International Collaboration for Flexible Distance Learning*. 1994.

[Inc95] Coral Technologies Inc. Firstweb. (Available on the world wide web as http://www.dive.com/coral/fcweb/fcweb.html), 1995.

[Inc96] Yahoo! Incorporated. Education: Online teaching and learning. (Available on the world wide web as http://www.yahoo.com/Education/On_line_Teaching_and_Learning/), 1996.

[Jac88] V. Jacobson. Congestion avoidance and control. *ACM SIGCOMM*, 1988.

[KL86] B. Kantor and P. Lapsley. Network news transfer protocol: A proposed standard for the stream-based transmission of news. (Network Working Group, RFC 977Available on the world wide web as ftp://ds.internic.net/rfc/rfc977.txt), 1986.

[LaL95] D. LaLiberte. Hypernews. (Available on the world wide web as http://union.ncsa.uiuc.edu/HyperNews/get/hypernews.html), 1995.

[Luo94] A. Luotonen. Wit: W3 interactive talk. (Available on the world wide web as http://www.w3.org/pub/WWW/WIT/), 1994.

[Moo86] M Moore. Self-directed learning and distance education. *Journal of Distance Education*, 1(1):7–24, 1986.

[MR94] J. Myers and M. Rose. Post office protocol – version 3. (Network Working Group, RFC 1725. Available on the world wide web as ftp://ds.internic.net/rfc/rfc1725.txt), 1994.

[NS96] D. A. Norman and J. C. Spohrer. Learner centered education. *Communications of the ACM*, pages 24–27, 1996.

[Ori92] W. Orilkowski. Learning from notes: Organizational issues in groupware implementation. *Proceedings of CSCW '92, ACM*, pages 362–369, 1992.

[Pos81] J. B. Postel. Transmission control protocol. (Network Working Group, RFC 793. Available on the world wide web as ftp://ds.internic.net/rfc/rfc793.txt), 1981.

[Pos82] J. B. Postel. Simple mail transfer protocol. (Network Working Group, RFC 821. Available on the world wide web as ftp://ds.internic.net/rfc/rfc821.txt), 1982.

[PWWW96] C. Pantel, C. Wick, K. Wong, and L. Winstone. Report on the evaluation of vgroups. Unpublished manuscript. Project report for CMPT 882, 96-1., 1996.

[Spe96] S. Spero. Analysis of http performance problems. (Available on the world wide web as http://sunsite.unc.edu/mdma-release/http-prob.html), 1996.

[Sto95] D. Stodolsky. Frequently asked questions list for comp.groupware[usenet]. (Available by anonymous FTP from rtfm.mit.edu in the directory pub/usenet/news.answers/comp-groupware-faq), 1995.

[Tea94] NCSA HTTPD Development Team. The common gateway interface. (Available on the world wide web as http://hoohoo.ncsa.uiuc.edu/cgi/), 1994.

[TH77] M. Turoff and S. R. Hiltz. Meeting through your computer. *IEEE Spectrum*, pages 58–64, 1977.

[Tur95] M. Turoff. The design of the virtual classroom. *Proceedings of the International Conference on Computer Assisted Instruction (ICCAI'95)*, 1995.

[Woo94] D. R. Woolley. Plato: The emergence of on–line community. (Available on the world wide web as http://freenet.msp.mn.us/ drwool/dwplato.html), 1994.

[Woo96] D. R. Woolley. Conferencing on the world wide web. (Available on the world wide web as http://freenet.msp.mn.us/ drwool/webconf.html), 1996.

[YH95] T. Yan and Garcia-Molina H. Sift - a tool for wide area information dissemination. *Proceedings of the USENIX 1995*, pages 177–86, 1995.