

REMOTE COLLABORATION IN THE DESIGN STUDIO

A Thesis

by

ABEY MATHEW GEORGE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2004

Major Subject: Architecture

REMOTE COLLABORATION IN THE DESIGN STUDIO

A Thesis

by

ABEY MATHEW GEORGE

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Valerian Miranda
(Chair of Committee)

Julie Rogers
(Member)

Lloyd Korhonen
(Member)

Phil Tabb
(Head of Department)

May 2004

Major Subject: Architecture

ABSTRACT

Remote Collaboration in the Design Studio. (May 2004)

Abey Mathew George, B.Arch., Kerala University

Chair of Advisory Committee: Dr. Valerian Miranda

Information technology offers many tools for promoting collaboration and communication in architectural design. A growing number of companies and individuals are adopting computer-based techniques to facilitate remote collaboration between geographically distributed teams. Thus, it is important to investigate the use of technology in developing collaborative tools for architects, especially as required training in architectural education.

This research explores the feasibility of augmenting communication in the design studio using a web-based collaboration tool. A prototype was developed for an integrated system that allows for streaming media, real-time collaboration, and multi-way video, audio and text messaging, tailored specifically to the needs of a distributed architectural design studio. The Collaborative Online Architectural Design Studio (COADS) is based on a three-tier client-server structure consisting of an interface tier, an application-logic tier and a data tier. COADS allows role-based participation for students and teachers, facilitating collaboration over design sketches and presentations using personal computers equipped with a microphone and a web-cam.

The system was developed and subjected to usability testing in a design studio consisting of graduate-level students of architecture. The participants were required to use COADS for conducting peer evaluations of designs for their class project and subsequently, to answer a questionnaire assessing the usability of the system.

The analysis showed that COADS has definite advantages as a tool to augment communication in the design studio. The biggest advantage was that participants could get immediate feedback about their designs from their peers, irrespective of their location. COADS was also relatively easy to set up on end-user machines and provided an integrated point for accessing relevant studio resources from a single location.

The disadvantages were mostly due to the limitations of the hardware on end-user machines such as small screen sizes, low quality microphones and web-cams. Further, the collaborative whiteboard within COADS lacked essential tools, such as pan/zoom and erase/undo tools, which reduced its usability.

In conclusion, systems such as COADS can effectively augment communication within the architectural design studio. However, they need to be integrated closely with the course structure, right from the introductory stage of the project to the final presentation stage.

DEDICATION

To my family

ACKNOWLEDGEMENTS

Special thanks to Dr. Miranda for chairing my committee and for his tremendous support and patience in guiding me through this endeavour.

I wish to thank Dr. Rogers and Dr. Korhonen for their guidance and enthusiastic support.

Thanks also to the students of ARCH 607 – Fall'03 for helping me test and refine COADS.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	v
ACKNOWLEDGEMENTS	vi
TABLE OF CONTENTS.....	vii
LIST OF FIGURES.....	x
 CHAPTER	
I INTRODUCTION.....	1
Background.....	1
Problem Statement	2
Hypothesis	3
Research Objectives	3
Research Methodology.....	3
Scope & Limitations	6
 II LITERATURE REVIEW	 7
Design Communication.....	7
Design Studio	9
Computer Aided Design (CAD)	10
Collaboration in Design Teams	11
Collaborative Design Tools.....	13
Interface Design.....	16
 III SYSTEM DESIGN AND SPECIFICATION	 20
Identification of System Requirements.....	20
Conceptual Model.....	21
User Model	25

CHAPTER	Page
IV IDENTIFICATION OF TECHNOLOGY	28
Interface Design Tools	28
Application Server	30
Database Server	30
V IMPLEMENTATION	31
User Interface Design.....	32
Application Server	42
Database Server	45
VI EVALUATION	47
Evaluation Process	48
Questionnaire.....	51
Domain Survey	52
Features Survey.....	56
Usability Survey	60
VII CONCLUSION	76
Future Work.....	78
REFERENCES	80
APPENDIX A SCRIPTS	83
Main Interface scripts.....	83
Common Interface scripts	90
Subscriber Interface scripts	95
Moderator Interface scripts.....	101
Login Interface scripts.....	105
Resources Interface scripts	110

	Page
COADS Communication Server Application scripts.....	113
User authentication script.....	115
Subscriber file upload script.....	117
Moderator file upload script.....	121
Retrieve file list script.....	125
 VITA.....	 127

LIST OF FIGURES

	Page
Figure 1: Conceptual model for COADS.....	22
Figure 2: Conceptual model for uploading a file for discussion.	22
Figure 3: Conceptual model for participating in a discussion.....	23
Figure 4: Conceptual model for initiating a discussion.	24
Figure 5: User model.	25
Figure 6: Three-Tier structure.....	31
Figure 7: UI structure.....	33
Figure 8: SWF structure.....	34
Figure 9: Screenshot [COADS Main interface].	36
Figure 10: Screenshot -Detail [COADS Login interface].....	37
Figure 11: Screenshot - COADS Subscriber interface.	38
Figure 12: Screenshot – Detail [Control Panel and People List tools].....	39
Figure 13: Screenshot – Detail [Presentation board].....	40
Figure 14: Screenshot - Upload files interface.....	41
Figure 15: Screenshot - Moderator Control Panel.....	42
Figure 16: Application server design.....	43
Figure 17: Communication Server application design	44
Figure 18: Screenshot one – COADS session.....	49
Figure 19: Screenshot two – COADS session.....	49
Figure 20: Screenshot three – COADS session.....	50

	Page
Figure 21: Using the prototype could significantly improve communication with other students.....	52
Figure 22: Using the prototype could significantly improve communication with faculty	53
Figure 23: Centrally located source of studio resources can allow for easy access of design information.....	54
Figure 24: Using the prototype can result in significant feedback on my design.....	54
Figure 25: The prototype can integrate very well with my existing design studio workflow	55
Figure 26: The collaborative drawing tool can help significantly with design communication.....	56
Figure 27: Face to face contact using web cameras can help facilitate better communication than text or audio only	57
Figure 28: Audio support for hands free conversation can help provide better focus for collaborative sketching	58
Figure 29: The studio resources feature can allow for easy access of relevant studio material.	59
Figure 30: The instructions to use the application were clearly specified online.	61
Figure 31: Navigation within the application was easy and intuitive.....	61
Figure 32: Finding information within the application was easy and intuitive.	62
Figure 33: Uploading files/information to the application was easy and intuitive.	63
Figure 34: Moving between my design programs and the prototype was easy and trouble free.....	64

	Page
Figure 35: Material presented in the application was easy to read (colors, graphics, fonts etc.)	64
Figure 36: Use of menus (draggable menus, drop down menus etc.) was intuitive and user friendly.	65
Figure 37: Use of color (e.g., supports material; provides interest, appeal) was appropriate and consistent.	66
Figure 38: The application status was clearly visible at all times (the current state of the application, such as loading interface, loading images etc.).....	66
Figure 39: The symbols used in the application matched existing practices in the design studio (eg.: icons, nomenclature etc.)	67
Figure 40: User control & rights were clearly specified and adequate for my role as a student.....	68
Figure 41: The user-interface is consistent with standard applications that I use (eg.: Internet Browsers, graphic applications etc.).....	68
Figure 42: The user-interface is appealing & aesthetically pleasing to use.....	69
Figure 43: The user-interface design is minimalistic and does not interfere with the use of the application	70
Figure 44: The user-interface is easy to use.....	70
Figure 45: The system and error messages are clear in their meaning.....	71
Figure 46: The system and error messages are adequately provided.	72
Figure 47: Online help and documentation are easy to use.	72
Figure 48: Online help and documentation provide adequate information about all areas of the application.....	73
Figure 49: Overall, how would you rate this application as an effective communication solution for the architectural design studio.....	74

CHAPTER I

INTRODUCTION

Background

The process of design is fundamentally collaborative with many participants, wherein ideas are continually being shaped and refined by input from all participating members. Design communication is an effective means to negotiate with other participants about the design product and the design process itself. Furthermore, studies indicate that this need for collaboration and decision-making in a group setting reflects a larger need among designers for incorporating the social skills required to interact with clients, users, specialists, consultants and others [1]. The lack of adequate communication and collaboration among design and engineering teams can adversely affect the quality of the process and product. Further, the failure within design organizations, to learn from past experiences has been linked to the inadequate and ineffective communication between teams, resulting in repeated and costly errors [2].

Information technology offers many effective tools for promoting collaboration and communication in the design process. Recent advances in information technology have made it possible to seamlessly and effectively integrate design collaboration activities into the everyday work processes of professional and academic design teams.

This thesis follows the style and format of Computer Networks.

Today, a growing number of companies and individuals are adopting computer-based collaborative techniques to facilitate remote collaboration between geographically distributed teams for effective decision-making [3].

Given the overwhelming need, in the professional environment, for effective group communication within and between design teams, it is imperative that we understand its role, significance and importance within design education. According to the Burton Report [4], architects are generally not perceived to be good listeners, communicators or team players. The tendency of architectural education to emphasize “the primacy of the individual” [5] leads to an inward focus within the design studio inculcating a design culture that isolates itself from the influences of the outside world. This is in striking contrast to the reality of the profession wherein architects are called upon to be very good team players.

Thus, it is important to investigate further the role of technology in developing collaborative tools for architects, as a means for facilitating communication between the growing numbers of globally dispersed teams. More important perhaps for architectural education, is the need to augment the shortcomings of the established modes of communication in design studios and to prepare students to be effective users of the emerging technologies of collaborative design development.

Problem Statement

This research is an investigation into the application of remote collaboration as a means for augmenting communication and collaboration in architectural education by developing and testing a prototype for an online design studio.

Hypothesis

This research hypothesizes that if a web-based remote collaboration solution is intergrated into the architectural design studio curriculum, it may augment design communication among participants in the studio.

Research Objectives

The research investigates remote collaboration in the design studio by addressing the following objectives.

- Investigate the application of current web-based technology to augment communication in the architectural design studio
- Determine and demonstrate the feasibility of developing web-based remote collaboration solutions by studying the requirements of a design studio and implementing a prototype for a system called Collaborative Online Architectural Design Studio (COADS).

Research Methodology

The research methodology adopts commonly utilized strategies from software engineering and human computer interaction disciplines, based on the spiral model of software development and enhancement [6]. These procedures combined with the five-stage process summarized by Rubenstein [7] leads to a five-stage research methodology to develop the proposed application.

Information gathering

The information gathering stage helped to identify the desired characteristics for the prototype and formulate requirements for the system design. This was achieved by:

- Talking to architecture students and faculty to understand the subtleties of the design studio and the possible potentials, limitations and character of a remote collaborative design studio session.
- Drawing on personal experiences in the design studio and explore the possibilities and limitations of remote collaboration.
- Studying existing methods for collaborative design activities (both digital and non-digital).
- Analyzing tasks of the potentials users of COADS.

System design & specification

The system design & specification stage, used the data from the information gathering stage to formulate design specifications for COADS. This process involved

- Identification of system requirements, goals and specifications.
- Specification of the "conceptual model" through which users perceive the system.
- Specification of the "use model" describing system use, documentation and error handling.

Evaluate & identify appropriate technology

After identifying the system specifications for COADS, an evaluation of current web-based technology was conducted to identify the appropriate tools for implementing COADS. This stage identified tools for the following development areas:

- Interface design tools
- Authoring application
- Remote communication server technology
- Dynamic web applications server technology

Design & implementation of the prototype

This stage involved developing the prototype, based on the specifications identified in the system design and specification stage, using the tools identified in the previous stage. The design and development process addressed the four main components of COADS, namely:

- Graphic user interface
- Server side application logic and communication server logic
- Database server and shared object implementation
- System setup

Testing, evaluation & iterative design

The evaluation process tested the usability of the prototype with respect to the stated goals of the study. Studies of usability inspections methods by Mack & Nielsen [8], indicate that four to five users are sufficient in most cases to identify 80 percent of the

total usability problems that are likely to be identified. Accordingly a combination of heuristic evaluation techniques and formal usability inspections [8] with end-users was used to evaluate the prototype by

- Testing the user interface on a subset of the sample user population for feedback.
- Analyzing feedback and evaluating the results.

Scope & Limitations

This research is limited to the development of a prototypical system that augments collaboration in architectural educational design studios. It is not meant to replace the physical collaborative activities that currently exist in the design studio.

The system users will be limited to students, faculty and external participants in a single design studio. Currently available computing technology will be utilized and as such the system features will be restricted by the limitations of the technology used. However, it is expected that technology will not affect the general research concepts.

The proposal is limited to evaluating the usability of the system and consequently does not include in its scope, issues such as performance, cost, flexibility, social issues of remote collaboration and design quality.

CHAPTER II

LITERATURE REVIEW

The literature review process was a part of the information gathering stage and covered a wide range of publications and articles covering area relevant to the research namely:

- Design Communication
- The Design Studio
- Computer Aided Design
- Collaboration in Design Teams
- Collaborative Design Tools
- Interface Design

The material covered issues relevant to communication, design , collaboration and technology. It also compared collaborative tools and distance learning solutions currently in use and their relevance to the architectural design studio

Design Communication

The role of communication in the process of design as a means to negotiate with other participants about the design has been clearly emphasized in design literature [9]. Furthermore architectural design by its very nature tends to be a team effort, where ideas are shaped and refined by input from all participating members. This need for decision making in a group setting is reflective of a larger need for social skills desired in

designers who in real life have to interact with clients, users, specialists, consultants etc. An examination of work schedules of professional architects, clearly show that most of the design time is spent interacting with specialist consultants and other architects, a fact not reflected in the curricula of most schools of architecture [9]. Experiments with Gambit, a group dynamics game that explores architectural teams, show that teams that resolve group tensions well, outplay teams that may have highly talented designers but do not collaborate well [10]. Successful group dynamics involve a proper understanding of group norms, leadership roles as well as membership roles and consequently results in design practices that are intensely social. This results in a common vocabulary and a sharing of concepts that facilitate understanding within the group.

The lack of adequate communication among engineering design teams has also been shown to adversely affect the quality of the design itself. The study by Busby suggests that design organizations fail to learn from their past experiences resulting in repeated design errors [2].

The Burton Report – findings and recommendations of a Royal Institute of British Architects (RIBA), steering group of architectural education - finds that architects were generally not seen as good listeners, communicators or team players [4]. A review of architectural education in the UK suggests that the tendency to encourage the idea that architectural discourse is esoteric by nature and thereby limited use for communication purposes, isolates architecture from its public and its procurers, thus diminishing the value of the discourse itself. In her work, Dana Cuff proposes that the inward focusing of the design studio results in students becoming more isolated from the

outside world [5]. The tendency of current architectural education to emphasize what Cuff terms the “primacy of the individual” is in striking contrast to the real-world need in the profession for architects to be very good team players.

Unquestionably, there is a need not only to provide exposure to the students in existing group collaboration techniques but also prepare them for the future modes of group communication.

Design Studio

The Design Studio is the wellspring of design education for the architecture student. It is here that the students’ thoughts and creative ideals receive direction from the Studio Mentor through design critiques. Within the Design Studio itself, Design Critiques have always been the core of educational activity in the studio. The communication involved in design studio critiques has been shown to highly critical in shaping the overall quality of the student architect. Studies by Belkis Ulouglu [11], indicate that, the educational role of the Studio Mentor is considered the most important, followed by his/her personality traits and finally his/her characteristics as an architect/designer. The design studio critique has been shown to be the most powerful tool in architectural education. Critiques in the studio normally fall into three categories :- individual desk crits, group crits, juries [11]. The main purpose of the critiques, irrespective of their category, however remains the same, i.e.,

- to communicate with the individual student and live in his/her world
- to bring the student face-to-face with others ideas to see each others worlds

The paper by Ulouglu [11] goes on to further analyze the design studio critique and explain its nature of functioning and concludes with certain findings. Striking among these conclusions are that

- emphasis will shift from predesign to the design process stage to the development of design solutions
- discussion on supporting knowledge will take place within a more personalized course, where the student takes control of his/her own development.

Therefore, an increasing amount of interaction, throughout the design process is called for, that allows a free flow of ideas and critiques both between students and with faculty.

Computer Aided Design (CAD)

CAD systems have evolved over the years to provide efficient mechanisms for producing drawings. While very efficient in addressing the issues related to production drawings or the end products of design, they have been inadequate in addressing the conceptual design development phase. This failure in providing successful design development aids is especially true in the field of design collaboration. The exponential growth in computer networks, and its potential for design collaboration has however sparked considerable research in this area [12].

The most pertinent of these questions however relates to the nature of communication that is desirable during design development and has been shown to be both visual and aural. Studies have clearly shown the importance of the conjunction between drawings and talking in design groups [13]. Both graphics and words have their advantages but combined together they offer a very powerful design language.

Design drawings do more than merely communicate the design. They foster the understanding of the problem and in some cases can even take over and become the focus of attention for the designer [9]. Even with the wide spread use of advanced computer graphics technology and CAD systems, hand drawings remain the central activity in the process of design [14]. At the design development stage, the drawings serve to represent a hypothesis for gradual refinement of the design solution and seem to do so by representing the limited amount of information being considered by the designer at that moment to the level of certainty in his or her mind [9]. Studies also show that designers prefer working with smaller drawings usually from A4 to A5 sizes [1], confirming the need to visualize their sketches at once. The value of drawings in communication seem to be clear, however their interpretation by different users can be problematic and needs to be carefully considered to avoid misinterpreted and confusing drawings.

Collaboration in Design Teams

Studies of the cognitive processes of design teams in dealing with design problems, based on four postulated cognitive processes of generation, exploration, comparison and selection have revealed a number of interesting results [15]. The collective design process is a combination of both content-oriented and process-oriented sequences, with teams spending as much attention on the structuring of the group processes as on the design problem itself.

For design problems of lesser complexity, a process that allows an immediate evaluation of solution ideas before analysis, results in considerable savings in time and

cognitive effort spent on a problem and it has been argued that design teams naturally tend to employ this process [15].

The authors [15] also emphasize the need for a larger focus on design methodology in addition to the end-product or the solution concept and suggest the adoption of a practitioner-centered design methodology as opposed to the prevalent normative view. The practitioner-centered approach takes into account the everyday constraints of the practitioner such as, time constraints, financial constraints, cognitive overload through simultaneous multiple projects etc.

David Craig's [16] study of unstructured online collaboration in the context of architectural design uses an asynchronous online environment, to provide contexts for communication that are autonomous from offline contexts to potentially support independent communities on its own. The premise behind this study being that, a well-guided online environment used in conjunction with an existing studio might be able to provide an alternative context for supporting open interaction. The study isolates three factors that make online collaboration in architectural studios unique, namely,

- Unlike most online unstructured systems, a large amount of time is required to represent an idea in architecture, spacing out interactions thus reducing continuity between exchanges.
- The representation of ideas as, traditionally presented in design studios are cumbersome, making it hard to reduce the boundaries between one person's ideas and another persons'

- Educational studios bring about groups of individuals who are very apart in age and experience; hence existing social boundaries between participants may be particularly strong.

Collaborative Design Tools

Various tools are currently used for asynchronous and synchronous distance education. The report by Jon Sigle [17], studies the benefits and limitations of these tools and advocates the training of faculty, development of higher performance hardware and software and the development of qualified technical support personnel. A strategy of experimenting with the technologies and gradually implementing a workable solution is advised.

The study of the adoption of collaborative technology in Boeing by Gloria Mark et al. [3] indicates a growing number of people adopting computer based collaborative techniques due to various organizational and geographical factors brought about by the necessity for geographically displaced teams to remotely collaborate over design decisions.

In a field study of NetMeeting, in a geographically distributed organization, Thomas Finholt et al.[18], found that the adoption of collaborative technology is extremely sensitive to small changes in the amount of effort required to utilize these applications, “champions” or enthusiastic proponents of the technology are vital for successful dissemination and non technical barriers such as shared behavioral protocols need to be resolved. Finally, perhaps the most critical findings relate to the easy availability of technical support and training programs to the users.

WebCT is currently one of the most popular applications available for distance learning. In their initial report on the project, Muray Goldberg et al. [19], describe the system and its implementation. Since its launch WebCT has been hugely popular and is currently used in a large number of educational institutions around the world.

None of the current distance learning solutions, however address the needs specific to the architectural design studio. As studies indicate, architectural design collaboration requires synchronous & interactive environments that provide effective tools for quick & intuitive design discussion and development [9]. While these solutions can be adapted to an architectural studio, there are inherent limitations in the systems that would prove the solution ineffective. The most noticeable shortcomings in WebCT, as observed by using WebCT and from the WebCT documentation, are in the following areas:

- Collaborative Drawing Environment

Though WebCT has a basic collaborative whiteboard, it does not support instantaneous real-time data updates thus resulting in delayed updates to the drawing board. It also launches the whiteboard as a secondary window thus forcing the user to deal with multiple windows and increasing the complexity of the system. It also does not seem to perform reliably when more than one user tries to upload images for collaboration. This limitation renders it inefficient for use as a remote collaboration tool in the architectural design studio as drawing capabilities are a critical requirement for design discussions [9]

- Control Structure

The WebCT drawing board does not support a user based control mechanism that allows moderating users to control the collaborative sessions. Any participating user may overwrite the ongoing session by uploading their images thus interrupting the discussion. However, within physical design studios, the presentation of information is usually more structured and directed by the moderator [11].

- Real-Time Audio Video

WebCT does not currently support real-time audio video collaborative tools. As the evaluation and analysis of COADS indicates, audio and video support for design collaboration was felt to be superior to text based communication and an important part of the collaborative experience.

The most noticable shortcomings in NetMeeting, as observed by using NetMeeting and from the NetMeeting documentation, are in the following areas:

- Control Structure

Though NetMeeting provides an effective whiteboard, the lack of a user based control mechanism that allows moderating users to control the collaborative sessions, renders it inefficient for collaborative design discussions.

- Integrated Environment

NetMeeting is currently implemented as a stand-alone application, that runs independently to handle communication functionality. It does not provide a single integrated environment with support resources that are designed for the

architectural studio. Consequently it forces the user to have multiple applications running and increases the complexity of the system

- Media Types

NetMeeting limits the type of presentation media to images only. There is currently no means to share multimedia presentations using slides and audio that are typically created in presentation software such as Microsoft Powerpoint and Macromedia Flash.

- Compatibility

NetMeeting requires the users to be running the Windows operating system and have the latest version of the Windows Messenger instant messaging software installed on their desktops. This excludes a wide range of users using alternate operating systems or alternate instant messaging software.

Interface Design

Recent research has shed significant light on the design of interactive systems. Researchers have developed predictive and descriptive theories while professional have attempted to compile useful guidelines for practitioners. Ben Shneiderman's taxonomy and rule based for interaction styles [20] provides guidelines for five different interaction styles, two of which are particularly relevant to this research

- Menu selection:

A menu selection and its variant the list selection, provides a list of items from which the user makes his/her selection. Menu selection is primarily advantageous because it decomposes a complex interaction into a series of smaller steps and

provides a structure for decision making. The other advantages of using a menu selection are shortens training, reduces keystrokes, permits use of dialog management tools and ease of supporting error handling. The disadvantages include slowing down frequent users, screen space requirements and requirements for rapid display rates [20].

- Form fill-in

In a form fill-in the users' main task is to provide data in labelled fields clustered on one or more screens. The form fill-in method make use of the keyboard, which is a very effective way to input length data fields. The advantages of using form fill-in are simplified data entry procedure, faster process for frequent users and the ability to use form management tools. The main disadvantages is the requirement for the user to have typing skills [20].

Shneiderman's guidelines recommend a blending of various interaction styles to suit a particular task with form fill-in for high data entry tasks and menu selection for multiple decision tasks.

Aaron Marcus's work on visual communication in user interface design [21] establishes principles for an effective user interface. The article uses principles from information-oriented systematic design to assist in providing guidance for simple, clear, consistent solutions for the design of menus, window, icons, dialogue boxes and control panels. A Graphic User Interface (GUI) design should account for the following:

- A comprehensive mental image
- Appropriate organization of data, functions, tasks and roles

- Efficient navigation schema among these data, functions, tasks and roles
- Quality appearance characteristics
- Effective interaction sequencing

Marcus proposes a set of three principles for GUI design [21]. These principles are suggested as a guide to professional practice research and development and include the following:

- Organize – provide the user with a clear and consistent conceptual structure.

There are four concepts within the major principle organization namely :

- Consistency – observe the same conventions and rules for all elements of the GUI and deviate from existing conventions only when doing so provides a clear benefit to the user.
 - Screen Layout – organize visual displays by designing their spatial layout or composition through the use of grid structures, standardized screen layouts or grouping related items.
 - Relationships – establish clear relationships by linking related elements and disassociating unrelated elements
 - Navigability – organize visual elements by using navigability techniques, such as providing initial focus for the viewer's attention, directing attention to secondary or peripheral information and assisting in navigation throughout the application
- Economize – designing with the least amount of elements required. Economy can be achieved by using the following techniques:

- Simplicity – include only those elements that are essential for communication
- Clarity – design all components so that their meaning is not ambiguous
- Distinctiveness – distinguish important properties of essential elements
- Emphasis – make the most important elements most easily perceived
- Communicate – to communicate effectively a GUI must keep in balance the following factors
 - Legibility – individual characters, symbols and graphic elements should be designed so that they are easily noticeable and distinguishable
 - Readability – the display should be easy to identify and interpret
 - Typography – typefaces should be small in number, of suitable legibility, clarity and distinctiveness to distinguish between different classes of information.
 - Symbolism – the icons, symbols and graphical elements used in the displays must be designed to efficiently convey their content
 - Color/texture – appropriate highlighting and deemphasis techniques should be used to convey meaningful semantic distinctions.

The principles outlined above [21] help to develop better visual communication in GUI's that allow users to benefit from the capability of high performance graphics systems to communicate more efficiently and effectively.

CHAPTER III

SYSTEM DESIGN AND SPECIFICATION

Identification of System Requirements

The information gathering process mentioned in Chapter I, utilized literature reviews, informal sessions with users, both students and faculty, and review of existing remote collaboration and distance education solutions, namely Microsoft NetMeeting and WebCT. These sessions allowed for the identification of specific features that would be desired in COADS, essentially:

- Capability for collaborative sketching overlaid on design studies.
- Real – time audio/video communications capabilities
- The capability for students to selectively interact with each other.
- The capability for students to interact with faculty.
- The capability for faculty to moderate interaction during faculty presentations.
- The capability to easily access other online resources relevant to the design studio session.
- The capability to host multiple student - faculty sessions.

The requirements, clearly indicated a need for role based user interaction with two primary roles, the Moderator and the Subscriber. This is derived naturally from the

physical design studio structure wherein participants are allowed a free hand in contributing ideas, while a Moderator directs or manages the overall structure of the studio. The Moderator user would typically be a faculty member, but in some situations could also be a student, who is elected to moderate the respective session. A third role would be that of the Presenter, i.e., a Subscriber who has been selected by the Moderator to be the current speaker or presenter.

Conceptual Model

The conceptual model is the end-users notion of the system model that he/she develops and relies on while using the system. Forming a conceptual model involves acquiring knowledge about the system and forming a theory of how it works by using the system, watching others use the system or reading the documentation. The conceptual model is then relied on while using the application to predict the behavior of the system. Thus it is imperative that the application match as closely as possible the conceptual model of the user in order for a more intuitive design. Figure 1 shows the conceptual model for the entire system.

The requirements analysis for COADS allowed the design of conceptual models for COADS based on some of the most common tasks. Figure 2 illustrates the conceptual model for a Subscriber to upload a file for discussion. The Subscriber logs into the system using his email id and pre-assigned password which directs them to the student interface.

After the upload completes, the Subscriber checks his list of uploaded files and if a Moderator is present in the current session, submits the file for discussion into the

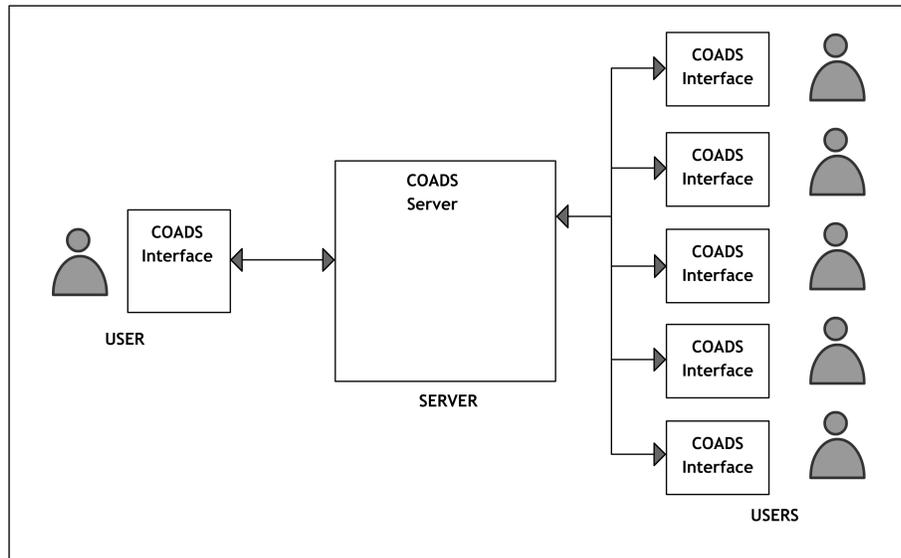


Figure 1 : Conceptual model for COADS.

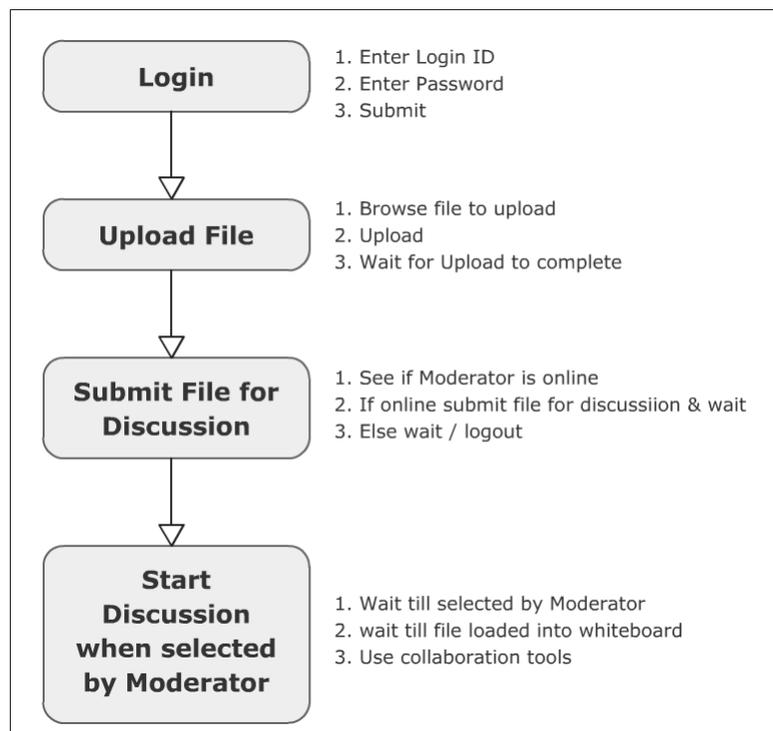


Figure 2: Conceptual model for uploading a file for discussion.

common queue. The Subscriber then waits until selected by the Moderator to discuss the file. On being selected the Subscriber proceeds to discuss the uploaded file with other participants using the collaborative tools.

Figure 3 illustrates the conceptual model for a Subscriber to participate in a discussion. The Subscriber logs into the system using his email-id and pre-assigned password and is directed to the Subscriber interface.

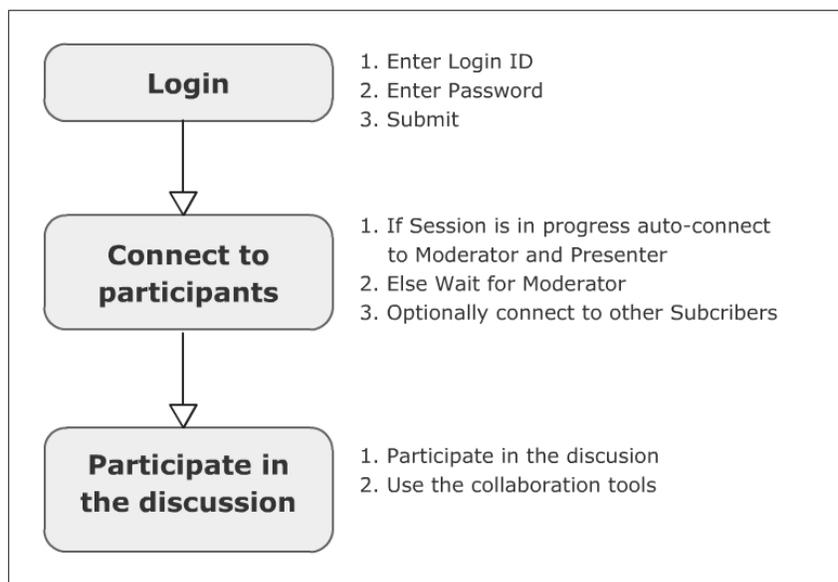


Figure 3: Conceptual model for participating in a discussion.

If a session is currently underway with a Moderator and/or a Presenter, the Subscriber automatically connects to them. If a session is not in progress, the Subscriber waits for the Moderator to initiate a new session. While the Subscriber is logged on, he/she may optionally connect to any other users that are online.

Figure 4 illustrates the conceptual model for a Moderator to initiate a discussion. The Moderator logs into the system using his/her email id and password and is directed to the Moderator interface. The Moderator then selects a Subscriber to be the Presenter, selects any one of his/her previously submitted files from the queue to the whiteboard on stage. The uploaded image and the Presenter's audio/video feed are then displayed on all connected users' screens. While the Moderator is logged on, he/she may optionally connect to any other users that are online.

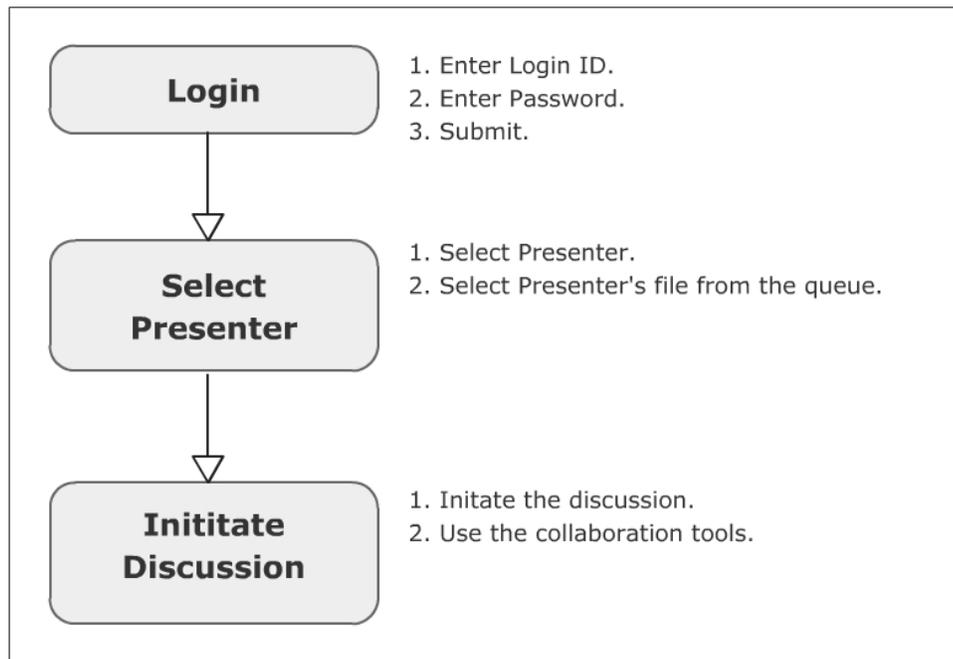


Figure 4: Conceptual model for initiating a discussion.

User Model

The user model for the system describes the interaction model for the entire system from the perspective of the end – user. As mentioned earlier, there are two primary user roles for any COADS session i.e. the Subscriber and the Moderator. A third role, the Presenter, is essentially a Subscriber who has been selected by the Moderator to present his/her files for discussion and occupy center stage. The

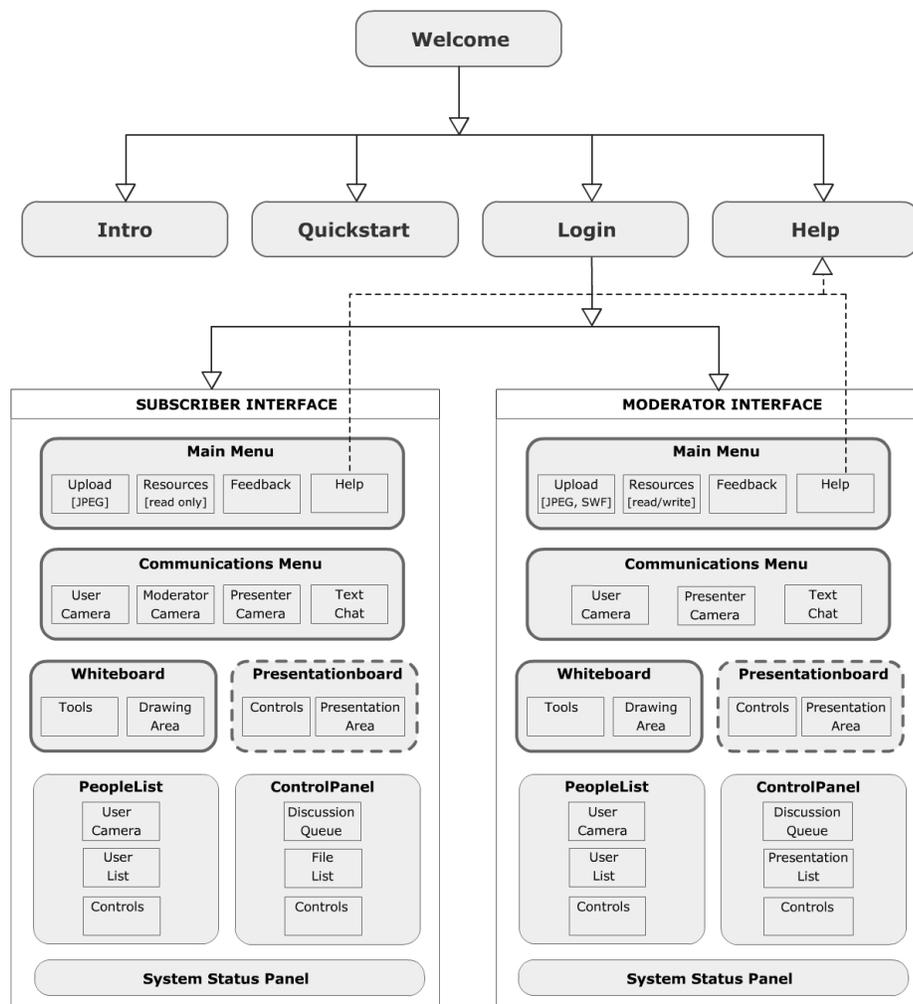


Figure 5: User model.

organization of user types thus leads to two specialized interfaces, the Subscriber Interface and the Moderator Interface. Figure 5 illustrates the user model for COADS.

The entry point to the COADS application consists of links to the introduction, quickstart, login and the help sections. The introduction provides a brief overview of the entire application. The quickstart section helps users understand the bare essentials needed to start using the application. The login section authenticates the user and directs them to the appropriate interface. The help section links to detailed documentation about COADS. After a successful login, the user is directed to either the Subscriber interface or the Moderator interface, depending on the user type.

The Subscriber interface consists of the main menu, the communications section, the whiteboard, the presentation board, the people list and the control panel. The main menu consists of action required by the user occasionally, for e.g. uploading files, accessing studio resources, submitting feedback and accessing the help files. The communication section consists of all elements related to communicating with other participants e.g. Moderator camera panel, Presenter camera panel, user camera panel and text chat panel. The white board occupies the largest area of the available screen space to provide a convenient drawing area comparable to letter sized paper. The presentation board remains hidden in the interface until a presentation session is initiated by the Moderator. The people list panel and the controller panel provide resources to connect to other people and controls to join discussions respectively.

The Moderator interface is similar to the Subscriber interface except for the additional controls available. The Moderator has control options over adding resources, initiating discussions, managing the file queue, initiating presentations, and managing chat sessions.

CHAPTER IV

IDENTIFICATION OF TECHNOLOGY

The identification of appropriate technology for developing COADS was dictated by its three tier client –server architecture and addresses three primary areas:

- Client (interface design & development)
- Application Server
- Database Server

Interface Design Tools

The interface was perceived to be one of the critical components of COADS, requiring capabilities to support audio-video, graphics, text and a flexible drawing Application Programming Interface (API). A further constraint was to make the client application operate across a wide variety of operating systems and browsers in a consistent manner. Various current technologies were considered including HTML, Java, and Macromedia Flash.

Macromedia Flash was selected for the client application technology because of its ubiquity on end-user machines across diverse operating systems and browsers as well as its capabilities as a rich client for developing Rich Internet Applications (RIA's). RIA's take advantage of interactive technology to offer intuitive, responsive and effective end –user experiences on the web that closely match desktop application experiences.

Rich client technology makes it possible to build RIA's by providing a runtime environment that can host compiled client-side applications delivered as files using Hypertext Transfer Protocol (HTTP). The client-side applications connect to existing application server back ends using an asynchronous client/server architecture. It offers a significant benefit over HTML pages by allowing a statefull environment throughout the duration of the application use. Some of the key benefits of using Macromedia Flash for developing COADS were:

- Client-side scripting : the availability of Actionscript, a client-side scripting environment based on the industry-standard European Computer Manufacturers Association (ECMA) Script, with an object-oriented object and event model as well as API's for client/server application development.
- Data connection through components: the availability of components that support data connectivity with external sources such as Extensible Markup Language (XML) and Simple Object Access Protocol (SOAP) based web services.
- Real-Time server communications: the integration with the Flash Communication Server that allows for real-time data exchange including two-way audio and video.
- Ubiquity: the availability of the Flash Player for a wide range of operating systems and browsers, thus allowing a consistent performance for a larger user base.

Application Server

The choice of the application server was decided by the availability of Windows 2000 (Internet Information Service) IIS servers in the Dept. of Architecture, Texas A&M University. All application logic was written in VBScript and hosted on the IIS server.

The selection of the communications server was limited to Macromedia Flash Communication Server (FCS). While, there are a few open-source solutions to address this problem, none of them provide the flexibility or power as FCS. Furthermore its integration with the Flash Player, through the use of communication components provide an easy framework to develop communication applications. FCS support for multi-way, real-time audio and video as well as one-to-many, broadcast-style streaming audio and video, was ideal for the stated goals of COADS. The Developer Edition of FCS was installed on an IIS server for testing COADS and provided support for a maximum of 5 concurrent users.

Database Server

The data associated with user logins is stored in a Microsoft Access database, while the data associated with COADS was stored in Remote Shared Objects (RSO's) on the Flash Communication Server. Due to the relatively low number of end-users testing the application, an MS Access database located on the IIS server was considered adequate for the job.

CHAPTER V

IMPLEMENTATION

The Collaborative Online Architectural Design Studio (COADS) was designed based on a three tier client server structure. Figure 6 shows the three –tier structure that consists of the interface client, the application server and the database server. The Client consists of the Macromedia Flash Player embedded in an HTML wrapper page which allows for a lightweight client that is available to the end – user free of charge. The Application Server consists of the Web Server, IIS using ASP for the application logic

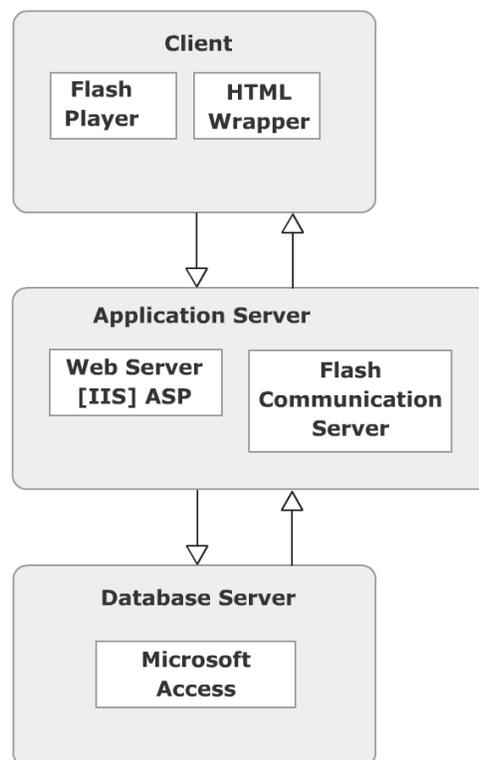


Figure 6: Three-Tier structure.

and the Flash Communication Server that handles the shared communication. The database server is an MS Access database that was also located on the IIS server. The three-tier architecture offers several advantages namely:

- The computational work can be distributed among different machines
- The clients can access the server's functionality from a distance
- The client and server can be designed separately, so they can both be simpler than a single program that does everything
- All the data can be kept centrally at the server thus making it easier to assure its reliability.
- Conversely, distributing data among many different geographically-distributed clients or servers can mean that if a disaster occurs in one place, the loss of data is minimized
- The server can be accessed simultaneously by many clients
- Competing clients can be written to communicate with the same server, and vice-versa

User Interface Design

As mentioned previously, the users of COADS were resolved into Moderators, Subscribers and Presenters. This structure called for two specialized interfaces, the Moderator interface and the Subscriber interface. In addition to these are the common interfaces required for all types of users. Figure 7 shows the structure of the user interface implementation of COADS. The interfaces were developed in Macromedia

Flash and compiled into Shockwave Files (SWF) that were embedded in an HTML wrapper page. The User Interface (UI) was developed as a central shell movie (coads.swf) that loads the secondary interfaces based on the user input. For eg. When the user selects the “login” option from the main shell, the login UI (login.swf) is loaded into the main shell.

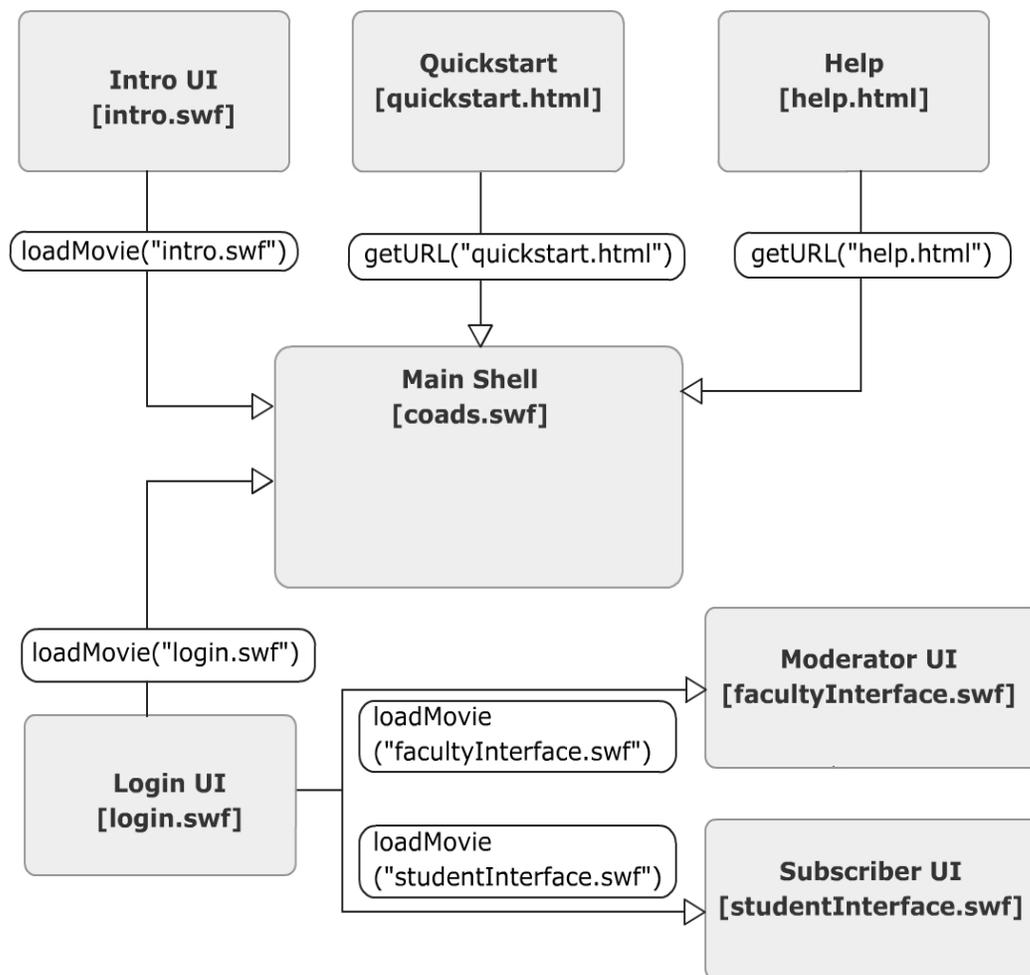


Figure 7: UI structure.

The information sections such as the help and quickstart were designed as HTML pages as it is a better medium for searching and displaying large text material. Figure 8 shows the components of a typical SWF file describing the UI. Each UI SWF file consists of on screen assets that represent the graphical parts of the UI and the Actionscript code that drives the logic of the UI. The communication related assets and code manage the parts of the UI that control the communications e.g. the camera panels, text chat panels, whiteboard etc.

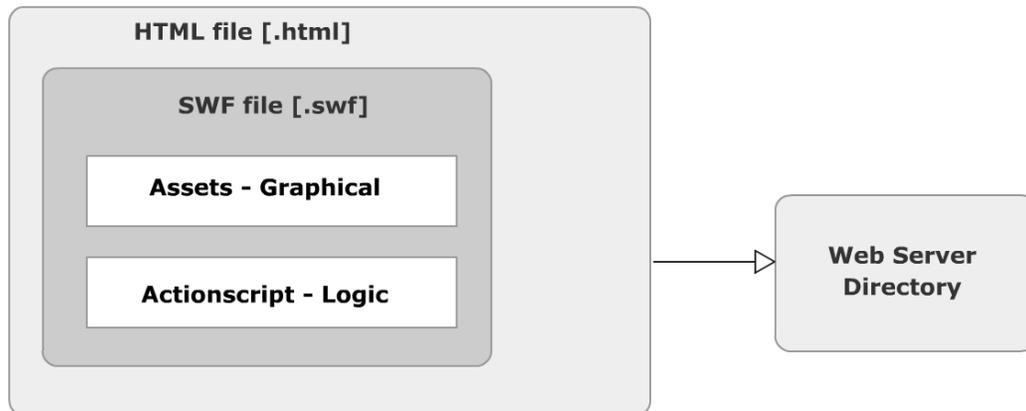


Figure 8: SWF structure.

The design for the interfaces follows the three common usability guidelines namely:

- Consistency - The color scheme, menus and buttons are kept internally consistent. Typical web based components, such as the login forms are kept consistent with existing trends, with deviations only in visual validation techniques, which needed to be more user friendly. The standard Flash library

components were re-skinned to be consistent with the look and feel of the application.

- **Layout** - The layout of the screen follows conventional guidelines for locating screen elements with hierarchical menu arrangement and intelligent screen resizing. The Moderator & Subscriber Interfaces are organized with a main menu at the top of the screen which allows the user to manage the application and access relevant resources. A secondary communications menu that allows access to the communication components is located immediately below the main menu. The initial feedback from the conceptual design suggested that the whiteboard component was too restricted to allow for effective graphical communication. Hence the whiteboard was designed to occupy maximum screen space available with the rest of the components designed as drop-down panels that could be relocated and minimized by the user.
- **Economy** - The interfaces were kept as simple as possible, especially the start screen, which was restricted to four controls that allowed the user to quickly ascertain his intended course of action. A limited color palette was utilized to allow important interface feedback to stand out.

Figure 9 demonstrates a screenshot of the COADS main shell, indicating the start point for the user. The start page consists of four main buttons that link to an introduction, a quick-start tutorial, a login interface and a help system. A system status menu is located at the bottom of all the interfaces to inform the user of the system state during the application use. Figure 10 displays the Login interface that is used to

authenticate the user. The design for the Login interface follows the common layout of web forms to allow the users to easily identify the actions required to access the system.

The Login interface provides users with three options:

- Login to the system
- Retrieve their password via email
- Register first time users of the system

A successful authentication directs the user to either the Subscriber interface or the Moderator interface, depending on the user's rights.

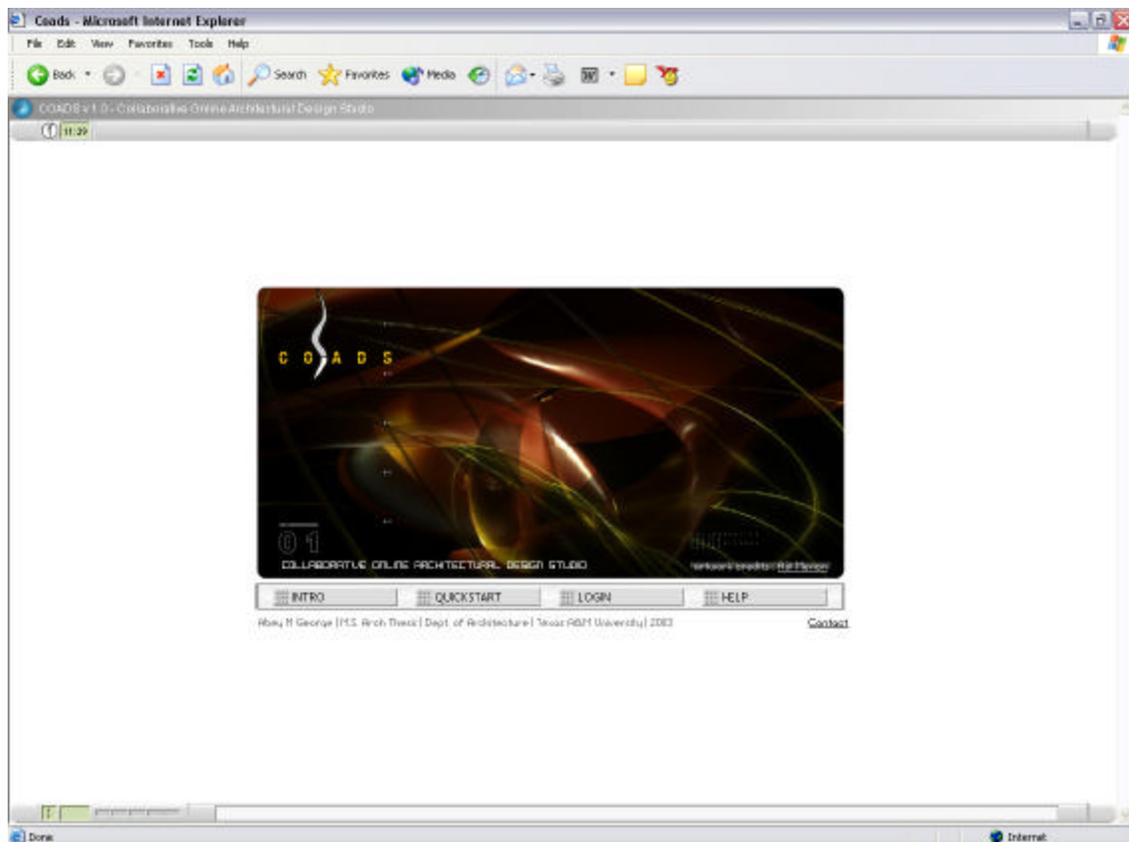


Figure 9: Screenshot [COADS Main interface].

LOGIN

close (X)

1. Email ID :

2. Password :

3. Click here to submit -->

Forgot your password ?

First Time User ?

INTRO QUICKSTART LOGIN HELP

Abeq M George | M.S. Arch Thesis | Dept. of Architecture | Texas A&M University | 2003 [Contact](#)

Figure 10: Screenshot -Detail [COADS Login interface].

Figure 11 displays the Subscriber interface. The user controls in the Subscriber interface are arranged in a hierarchy based on their functions. The top-most menu consists of actions used less frequently by the user such as file uploads, resource access, feedback and help. They are fixed menus that spawn secondary windows either internally in COADS or as external browser windows.

The second level of menus, address communication-related actions for the user including text chat, Presenter camera, Moderator camera and user camera. These menus are fixed collapsible menus that can be minimized to increase the available drawing area. These menus additionally, allow the user to set preferences for communication components such as the bandwidth selector, shared cursor and the text chat panel. The third category of menus, host the Control Panel and People List tools that are likely to be

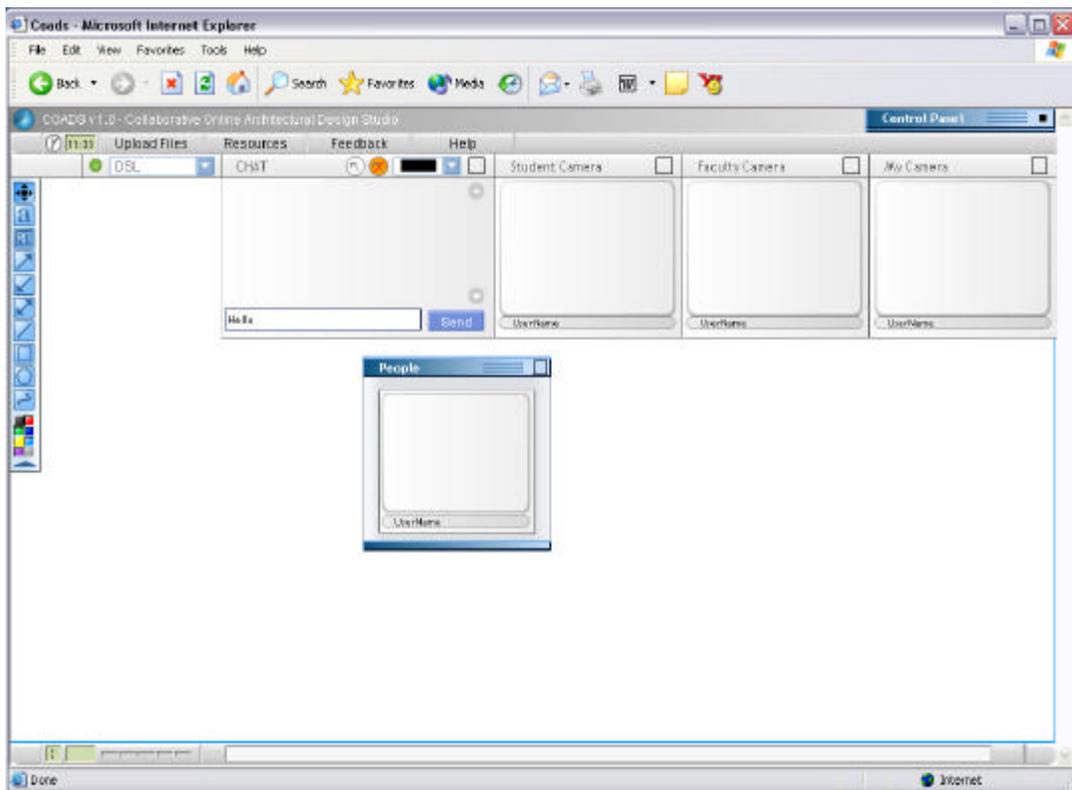


Figure 11: Screenshot - COADS Subscriber interface.

used frequently by the user. These are floating menus that can be minimized and positioned anywhere on screen based on the user's requirements. The main area of screen is designated for the Whiteboard that that allows the users to collaborate over the drawings. The Whiteboard component is an extension of the default Macromedia component with additional feature sets such as advanced drawing tools for freehand lines, rectangles and a circle. The users can use the drawing tools, a text tool, arrow tool

and can select different colors for their drawing entities. They can also select individual drawing entities and selectively delete them.

Figure 12 displays the Control Panel and People list tool. The Control Panel tool for the Subscriber allows the user to view the list of previously uploaded files and the common file queue. The user can select any of his files from the file list and add them to the queue for discussion. The file queue mechanism maintains the order of user files submitted and allows the Moderator to select one for discussion. The People List panel displays a list of people that are online and lets the user connect to their camera.

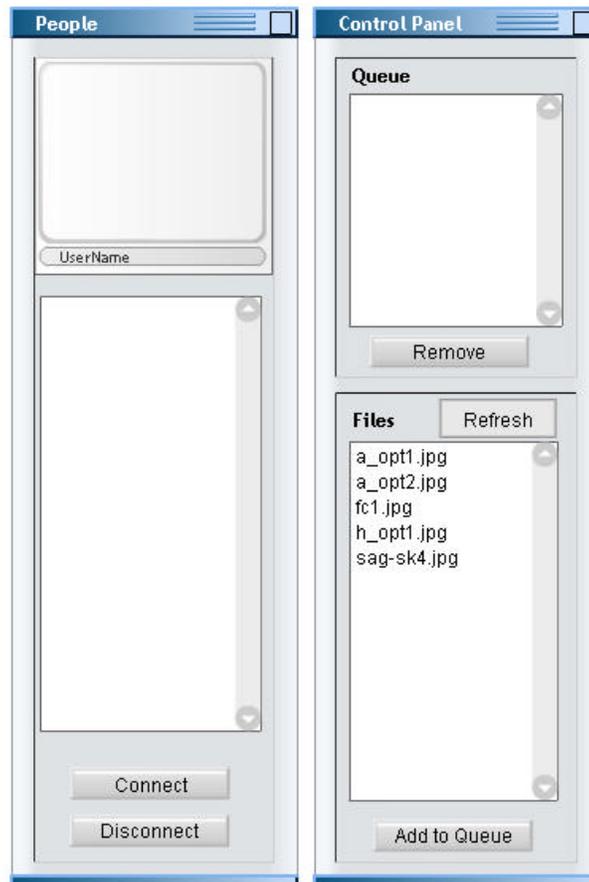


Figure 12: Screenshot – Detail [Control Panel and People List tools].

Connecting to a user streams the audio and video, if the selected user is currently publishing an audio – video feed. The People List can be minimized to display only the camera pane thus allowing more viewable area for the drawing board.

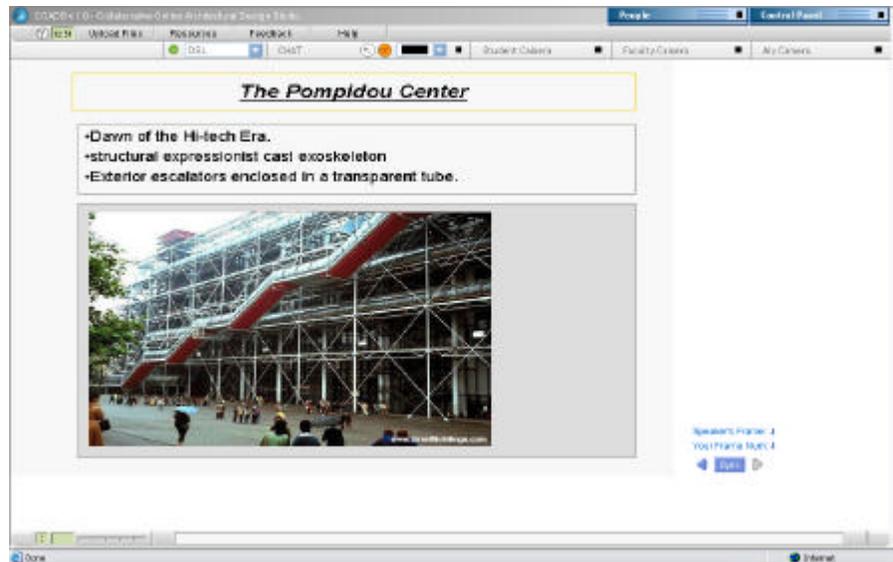


Figure 13: Screenshot – Detail [Presentation board].

Figure 13 demonstrates the Presentation board on the Subscriber interface. The Presentation board is invisible by default until invoked by the Moderator, and is used to deliver a presentation to multiple connected clients. The Subscriber has controls to move the presentation slides on his individual client and to synchronize his slide to the Moderators slide. When the Presentation board is activated, the Whiteboard is deactivated so as to focus the user's attention to the presentation.

Figure 14 demonstrates the Subscriber interface for uploading files. The

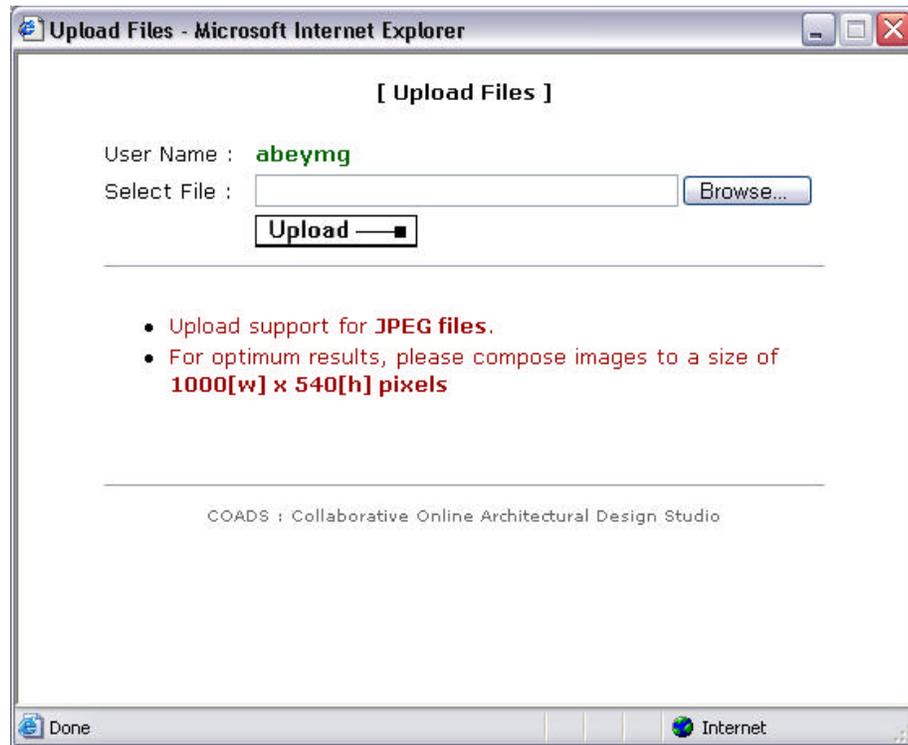


Figure 14: Screenshot - Upload files interface.

Subscriber is allowed to upload non-progressive JPEG files. The optimum size for the images is 1000 x 540 pixels. The user can browse his hard drive and select a file for upload and click “Upload” to submit his file to the COADS server. The status bar at the bottom of the popup window indicates the progress of the upload.

The Moderator interface is similar to the Subscriber interface except in the features available for the Control Panel. Figure 15 illustrates the Moderator Control Panel. The Moderator has control over managing the chat panel, launching discussions from the file queue, clearing discussions from the stage, removing requests from the file queue and launching/clearing shared presentations.



Figure 15: Screenshot - Moderator Control Panel.

Application Server

The Application server logic was developed using ASP-VBScript. Figure 16 illustrates the design of the application server. The Login interface sends the user login information to the authentication script on the server, after validating the user input on the client side. The authentication script verifies the user login information against the Access database and directs the user to the appropriate interface: Subscriber or Moderator. If the authentication fails, the script returns an appropriate error message to the Login interface that appears in the status bar informing the user about the cause of

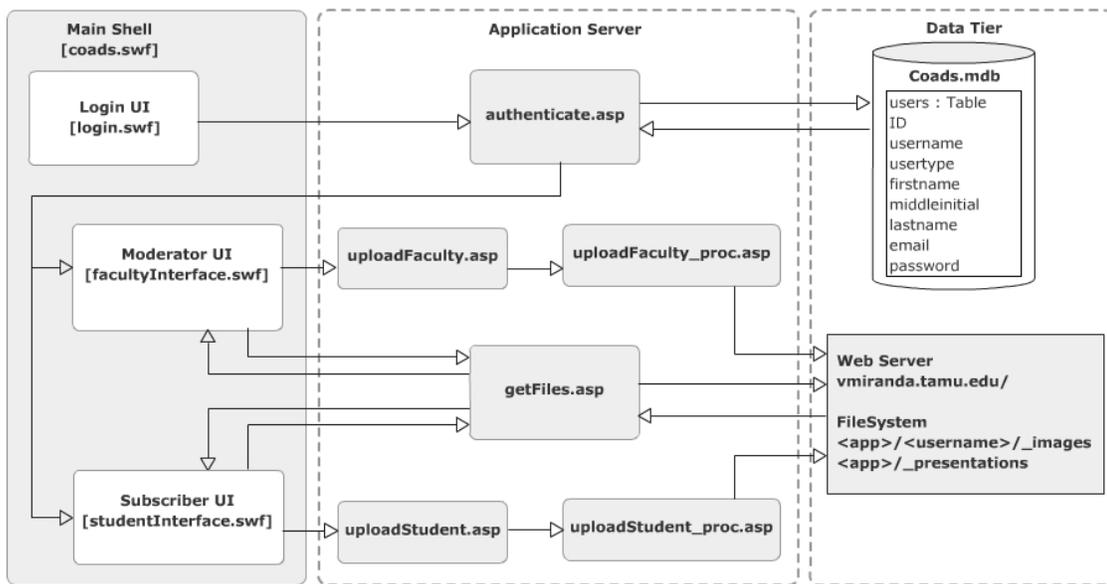


Figure 16: Application server design.

failure. The application server also hosts scripts to manage the storage and retrieval of user submitted files.

The files uploaded from the user interfaces are processed by the upload scripts that verify valid types and file sizes and store them on appropriate user directories on the web server. When the user logs into the system for the first time, the script creates a unique directory for the user based on the username on the web server's file system. All subsequent uploads are then stored in the user folder. In addition to user specific folders, the upload scripts also store presentation files in a global presentation folder, the contents of which are available to the Moderator interface. The "getFiles" script handles queries from the interface to list all files of the particular user and returns a list of files previously uploaded by the user. The script also handles request to list presentations files

and returns a list of previously uploaded presentations to the Moderator interface.

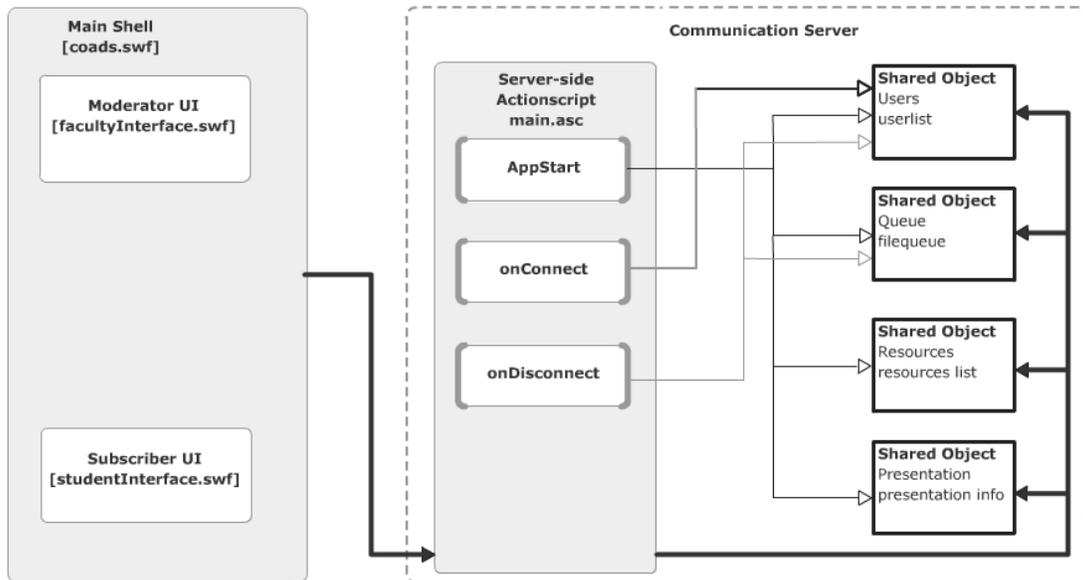


Figure 17: Communication Server application design.

Figure 17 describes the design of the Communication Server application. After authentication and redirecting the user to the appropriate interface, the interface connects to the Flash Communication Server. The server-side script, “main.asc” handles the different connection events and initializes/reads the Shared Objects that contain the relevant data.

The main script handles three specific events that correspond to, the start of the application, a user connect event and a user disconnect event. When the application first starts, the script initializes Shared Objects for the user list, the file queue and the presentation information. Its also reads the information from the resources Shared Object

which is persistent and carries information across application states. The script also handles default Shared Objects for all components utilized in the application such as the chat, whiteboard and presentation components. When a user connects to the application, user information is updated in the user Shared Object, and is broadcast to all connected clients that receive the event and handle it appropriately performing updates on the People List panel. If the user connecting is a Moderator, the event also publishes the Moderator video stream, which shows up on all subscribed clients. When the user disconnects the script performs clean up operations on the user, file list and presentation (if current user is set as a presenter) Shared Objects.

Apart from these connection specific events the communication server handles all other communication event requests from clients such as subscribing to video, publishing video, synchronizing whiteboard updates, synchronizing shared cursor movement, presentation board information, text chat communication, resource edits and retrieval, and shared image loads.

Database Server

The user information was stored in a MS Access database table. The table was designed to store unique ID's for the user in addition to the user specific information such as user name, user type , email and password. The user type specified whether the user was a Subscriber or Moderator. This design allowed for easy modification of user information and easy addition / deletion of users into the system.

The uploaded files consisting of images and presentations were stored on the web server in specific directories. All user image files were stored in unique folders based on

user ID, while presentation files were stored in a global presentation directory available to all Moderator users.

The data related to the communication server was stored in Shared Objects on the Flash Communication Server. Shared Objects are data objects stored on the Flash Communication server that allow for synchronous data sharing across multiple connected clients. Shared Objects are managed by the communication server and provide messaging, data synchronization and data storage services. Clients connect or subscribe to a Shared Object and receive updates whenever a change is made to that object. Shared Objects can be made to persist across application sessions. The following information was stored in Shared Objects :

- Text chat data – all text messages
- Whiteboard drawing data – point co-ordinates and line data for drawings
- Presentation board data – location of presentation file and frame information
- Shared cursor data – point coordinates of user cursor
- User list – list of connected users
- File queue – list of files submitted for discussions
- Resources data – persistent list of resources made available to the studio by the Moderators
- Presentation state data – information regarding the Presenter user and visibility states of the whiteboard and presentation board components

CHAPTER VI

EVALUATION

The goal of the evaluation was to examine whether the prototype application COADS, was useful in aiding design communication in the architectural studio. The evaluation process tested the usability of the prototype with respect to the stated goals of the thesis. As outlined in Chapter I, the main objectives of the prototype were the following:

- The development of a tightly integrated web application that allows for multiple collaborative activities from a single portal as opposed to disparate, point solutions that require the user to launch multiple applications.
- Providing for real-time collaboration using multi-way audio-video and text messaging within the application, tailored to the requirements of a distributed architectural design studio.
- The development of a collaborative drawing environment that provides for quick and intuitive conceptual sketching
- Provision of a user-based control structure that allows the “Controller-user” to structure and direct the online studio sessions, akin to a traditional studio setting.
- The capability to store and retrieve resources relevant to the studio either generated during online sessions or those that are external to the studio.

Studies of usability inspections methods by Mack & Nielsen [11], indicate that four to five users are sufficient in most cases to identify 80 percent of the total usability problems that are likely to be identified. Accordingly a combination of heuristic evaluation techniques and formal usability inspections [11] with end –users was used to evaluate the prototype by

- Testing the user interface on the sample user population for feedback
- Analysis of feedback and evaluation of tests

The evaluation was conducted as a usability analysis of the application by obtaining feedback from participants through an online questionnaire. The participants were graduate students in the Architecture 607 Design Studio in Fall 2003.

The students were asked to use COADS to discuss their design sketches during a studio session. These sketches were submitted to the COADS system prior to the session and were available online to the participants during the session. The students also selected a moderator among themselves to lead the online session. Figures 18, 19 and 20 show screenshots of the COADS session from the Moderator screen.

Evaluation Process

The students were initially assigned a login id and password, which was sent to them via email. The email contained instructions to access COADS, as well as directions to modify user information namely, the login password and email-id. The students were requested to upload their design sketches for a pre-selected design project to COADS in time for a designated COADS session. The session was conducted in the design studio

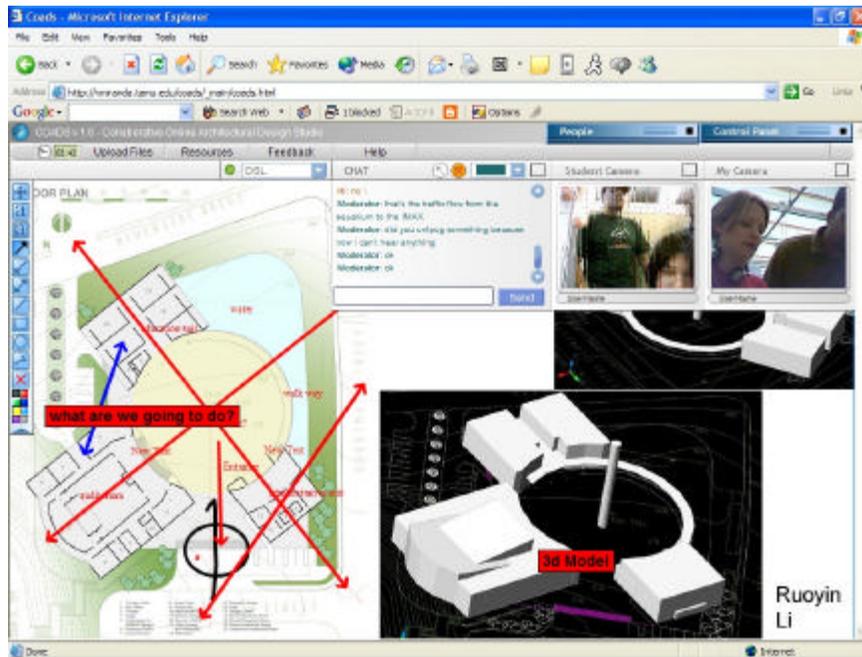


Figure 18: Screenshot one – COADS session .

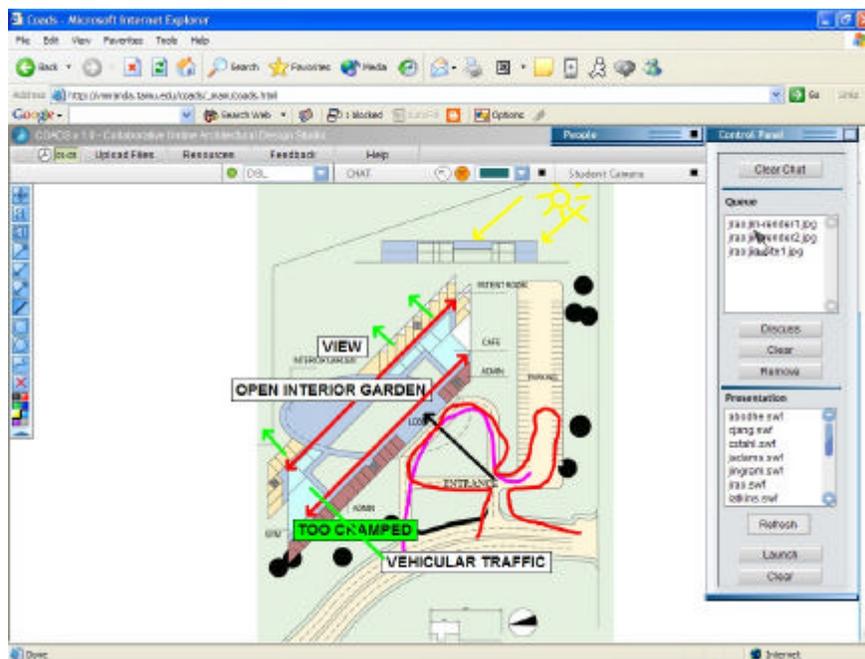


Figure 19: Screenshot two – COADS session.

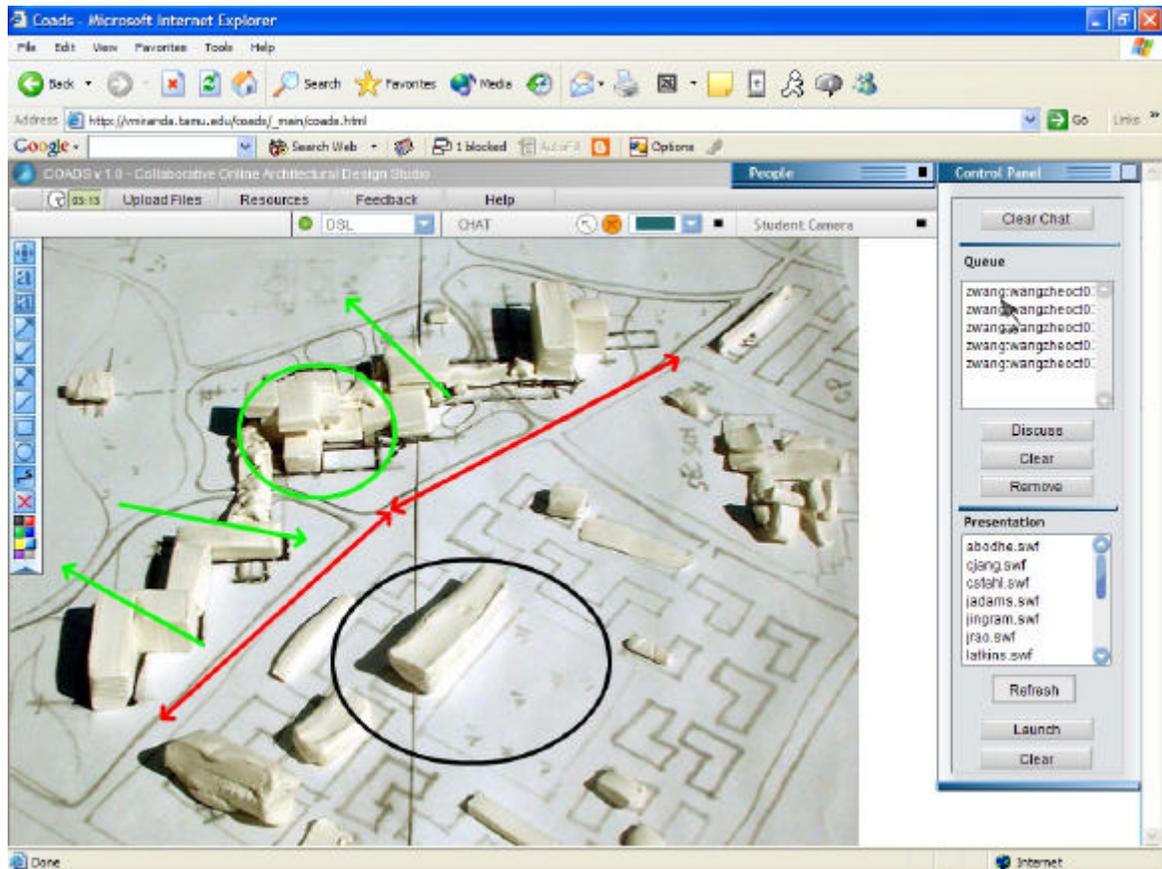


Figure 20: Screenshot three – COADS session.

using wireless laptops from different workstations. Due to the limitations of resources, some students shared a laptop and took turns submitting their work.

The designated Moderator logged into the system from a separate location in the design studio. The Subscriber users submitted their designs for discussions and the Moderator selected the Presenter to present their work. The users participated in the discussion using their web cameras, microphones and the collaborative drawing board.

The prototype was implemented using a developer version of server software with a restriction of 5 concurrent video streams, which effectively limited the system to supporting two simultaneous users. Consequently the collaborative experience was limited in scope. Additionally the quality of the communications hardware, affected the quality of collaboration. The evaluation process confirmed the importance of high quality web cameras and microphones to provide a better end-user experience.

Questionnaire

The participants of the COADS session were requested to answer an online questionnaire to obtain feedback on the quality of the user experience. The survey consisted of a set of questions and a Likert scale for rating user responses to the questions from 1 (strongly disagree) to 5 (strongly agree). The Summary section contained "Comment" boxes to obtain additional feedback relevant to the purpose of the survey. The questionnaire was designed based on usability heuristics and was divided into four parts, as follows:

- Domain Survey – Questions relating to effectiveness of the prototype in the context of the design studio
- Feature Survey – Questions relating to the specific features implemented in the prototype
- Usability Survey – Questions relating to specific usability issues of the prototype
- Summary – Questions to obtain user comments regarding the overall experience

Domain Survey

Question 1. Using the prototype could significantly improve communication with other students.

As indicated in Figure 21, approximately 57% of the respondents, either agreed or strongly agreed that the prototype helped to significantly improve communications with other students, with 43% remaining undecided. One factor affecting this result, might have been that the participants were located in close physical proximity, within the same design studio and in some cases were sharing a single laptop among themselves. Consequently, they had the option of communicating directly with each other, instead of communicating through COADS. A distance education environment with remotely located participants would make communication only through COADS, the most direct option available, to interact with other participants.

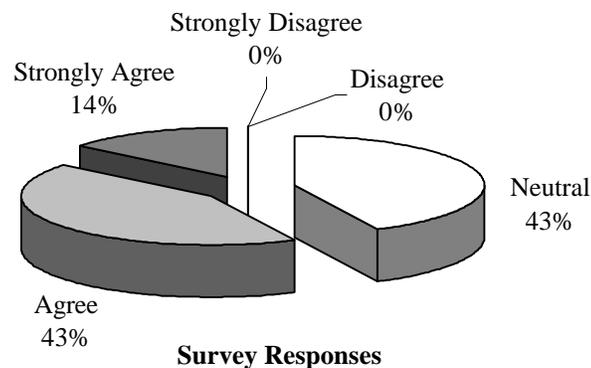


Figure 21: Using the prototype could significantly improve communication with other students.

Question 2. Using the prototype could significantly improve communication with faculty.

As indicated in Figure 22, approximately 57% of the respondents, either agreed or strongly agreed that the prototype helped to significantly improve communications with the Moderator, with 43% remaining undecided. This result was similar to Question 1 and would have been subject to the same effects as for Question 1.

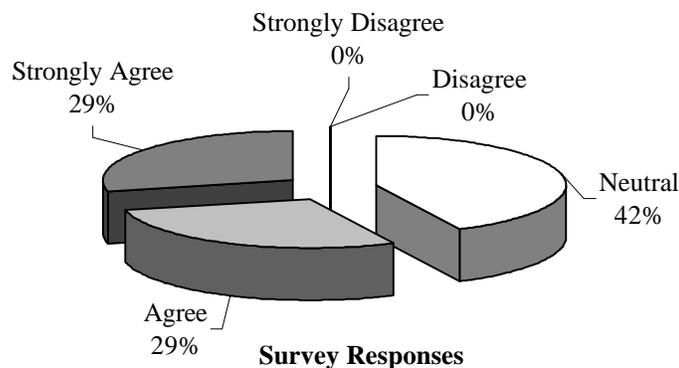


Figure 22: Using the prototype could significantly improve communication with faculty.

Question 3. Centrally located source of studio resources can allow for easy access of design information.

As indicated in Figure 23, approximately 86% of the respondents agreed and 14% strongly agreed that the centrally located studio resources within the prototype allowed for easy access of relevant design information. This suggests that COADS, provides an effective solution to access relevant online design resources, easily.

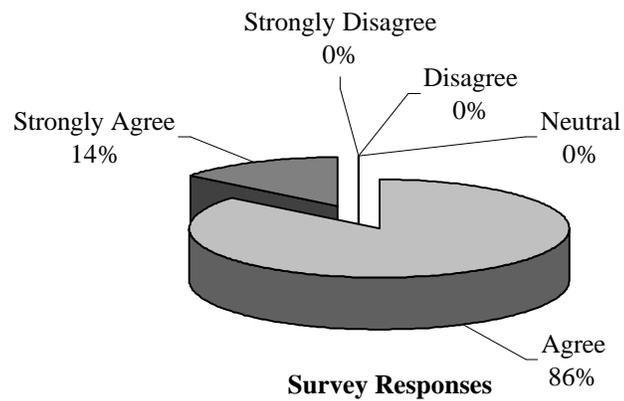


Figure 23: Centrally located source of studio resources can allow for easy access of design information.

Question 4. Using the prototype can result in significant feedback on my design.

As indicated in Figure 24, approximately 85% of the respondents agreed that the prototype resulted in significant feedback on their design studies. This suggests COADS, successfully facilitated communication among participants about their design studies.

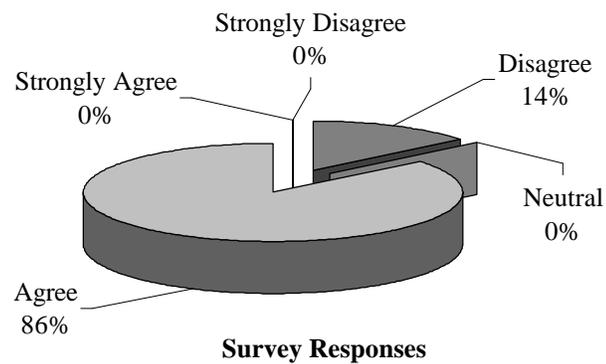


Figure 24: Using the prototype can result in significant feedback on my design.

Question 5. The prototype can integrate very well with my existing design studio workflow.

As indicated in Figure 25, approximately 28% of the respondents disagreed and 43% were about the quality of integration with the design studio workflow. This suggests that for COADS, to be effective, it is imperative that the use of COADS be very well integrated with design studio, for the duration of the semester. For future uses it is suggested that COADS be implemented in the studio at the start of the semester in an introductory phase and progressively be used at each level of the design process, from the problem definition stage, through the conceptual design stage and up until the final presentation stage.

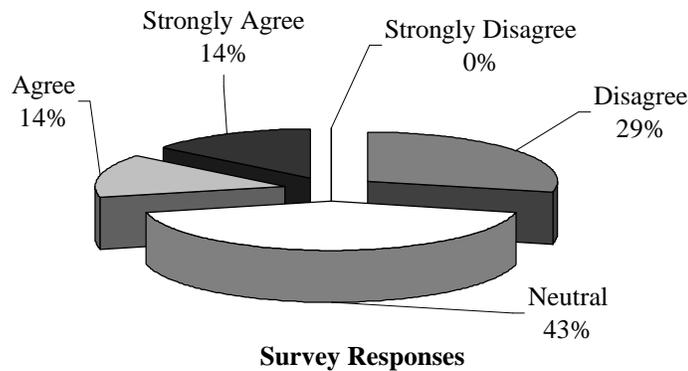


Figure 25: The prototype can integrate very well with my existing design studio workflow.

Features Survey

Question 6. The collaborative drawing tool can help significantly with design communication.

As indicated in Figure 26, approximately 29% of the respondents agreed and 57% were undecided that the collaborative drawing tool helped significantly with design communication. This result, when considered with the free-form text responses of the participants suggested that, though the whiteboard was considered to be a vital tool, its current implementation lacks two essential features, namely a pan & zoom tool and an erase/undo tool. The lack of these tools may have adversely affected the perception of the whiteboard's usefulness.

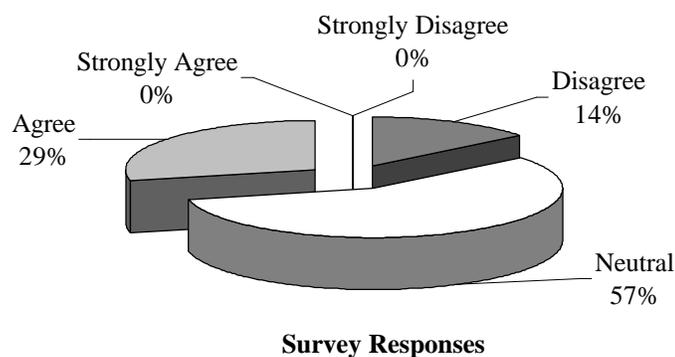


Figure 26: The collaborative drawing tool can help significantly with design communication.

Question 7. Face to face contact using web cameras can help facilitate better communication than text or audio only.

As indicated in Figure 27, approximately 86% of the respondents either agreed or strongly agreed that web camera enabled communication is better than using text and audio only communication.

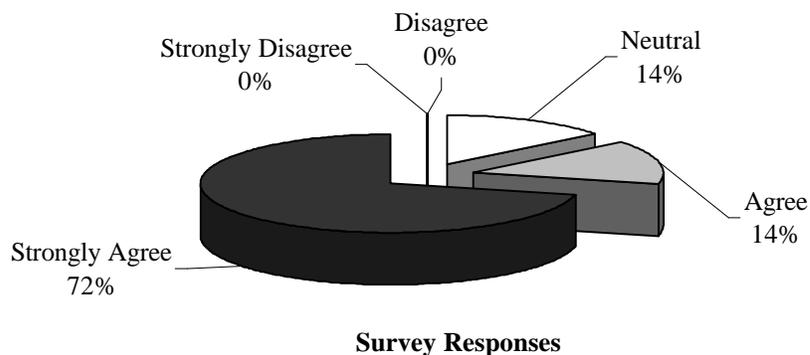


Figure 27: Face to face contact using web cameras can help facilitate better communication than text or audio only.

Question 8. Audio support for hands free conversation can help provide better focus for collaborative sketching.

As indicated in Figure 28, all the respondents either agreed or strongly agreed that audio enabled communication mode in COADS, allows for a better experience in using the collaborative whiteboard.

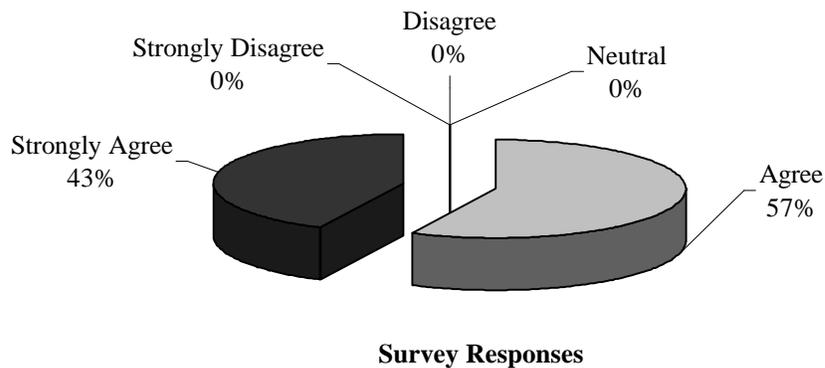


Figure 28: Audio support for hands free conversation can help provide better focus for collaborative sketching.

Question 9. The studio resources feature can allow for easy access of relevant studio material.

As indicated in Figure 29, approximately 57% of the respondents either agreed or strongly agreed and 29% were undecided about the ease of accessing the studio resources. This response when taken in conjunction with Question 3 suggests that while centrally located studio resources are perceived as a very useful feature, the implementation needs to be further clarified for easy access.

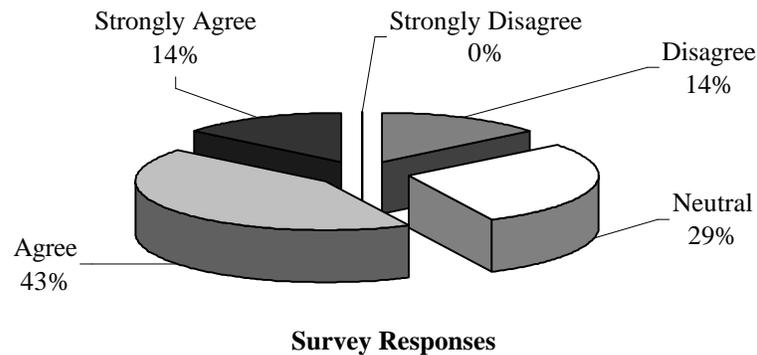


Figure 29: The studio resources feature can allow for easy access of relevant studio material.

Question 10. What features, if any, prove distracting to the collaborative design experience?

The comments from participants to this question identified various issues that prove distracting to the collaborative design experience while using COADS. The most prominent being, too many users at one time, limits of the screen size and limitations of the whiteboard.

The perception of too many users at one time, may have been affected by the fact that some participants had to share a laptop while using COADS. This led to frequent interruptions in the experience, as users had to relinquish control of the laptop to other users during the session. The observation could also be related to the optimum size of users in online collaborative sessions consistent with studies that show that the optimum size of users in online collaborative communities is around five.

The limitations of the whiteboard were essentially identified as the lack of tools related to zooming into and panning the drawing around the screen, and for an easy and intuitive erase and undo feature.

Question 11. What features, if any, would you propose that might improve the collaborative design experience?

Almost all the comments to this question, pertained to the limitations of the collaborative whiteboard. Suggestion ranged from software solutions for improving zoom and pan options, erases tools, to hardware solutions such as using a writing tablet instead of a mouse for a better drawing experience. This suggests that an effective implementation of the whiteboard is critical to the success of COADS.

Usability Survey

Question 12. The instructions to use the application were clearly specified online.

As indicated in Figure 30, approximately 45% of the respondents either agreed or strongly agreed and 45% were undecided about the effectiveness of the online documentation provided with COADS. This suggests that online documentation needs to be implemented more comprehensively and in conjunction with Question 29 implies that online documentation is critical to the effective use of COADS.

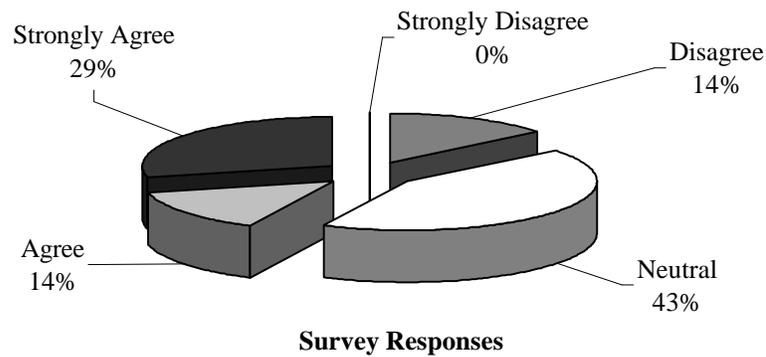


Figure 30: The instructions to use the application were clearly specified online.

Question 13. Navigation within the application was easy and intuitive.

As indicated in Figure 31, approximately 70% of the respondents either agreed or strongly agreed that navigation within COADS was easy and intuitive from a usability perspective.

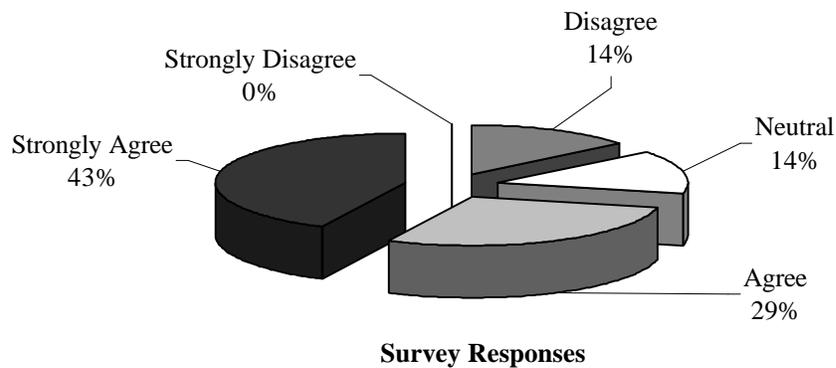


Figure 31: Navigation within the application was easy and intuitive.

Question 14. Finding information within the application was easy and intuitive.

As indicated in Figure 32, approximately 70% of the respondents either agreed or strongly agreed that finding information within COADS was easy and intuitive from a usability perspective.

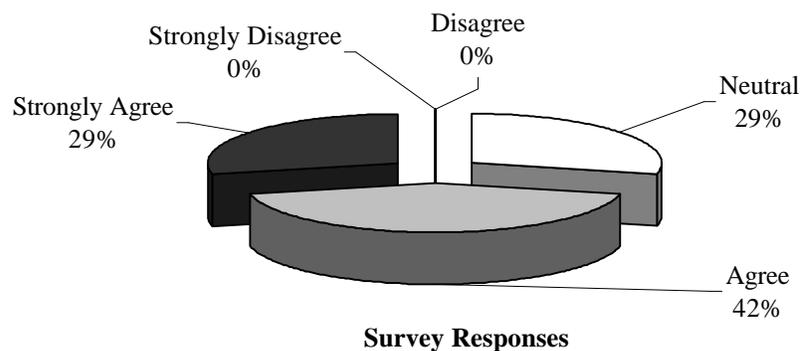


Figure 32: Finding information within the application was easy and intuitive.

Question 15. Uploading files/information to the application was easy and intuitive.

As indicated in Figure 33, all the respondents either agreed or strongly agreed that uploading files within COADS was easy and intuitive from a usability perspective.

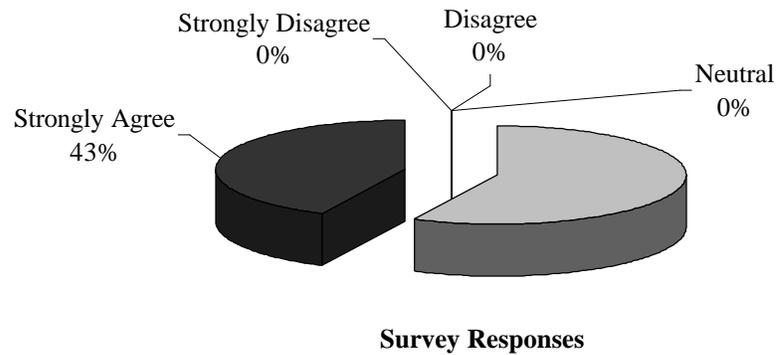


Figure 33: Uploading files/information to the application was easy and intuitive.

Question 16. Moving between my design programs and the prototype was easy and trouble free.

As indicated in Figure 34, approximately 57% of the respondents either agreed or strongly agreed and 28.6% were undecided regarding the ease of moving between frequently used design software such as AutoCAD, Photoshop etc. and COADS. This suggests the need for better integration of COADS with frequently used design software.

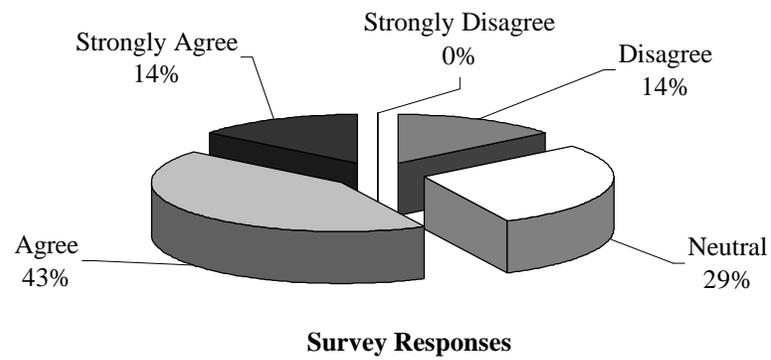


Figure 34: Moving between my design programs and the prototype was easy and trouble free.

Question 17. Material presented in the application was easy to read (colors, graphics, fonts etc.).

As indicated in Figure 35, all the respondents either agreed or strongly agreed that information provided within COADS was easy to read from a usability perspective.

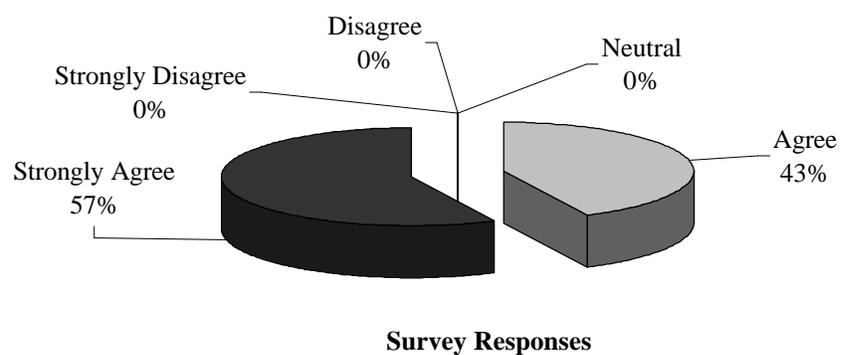


Figure 35: Material presented in the application was easy to read (colors, graphics, fonts etc.).

Question 18. Use of menus (draggable menus, drop-down menus etc.) was intuitive and user friendly.

As indicated in Figure 36, approximately 70% of respondents either agreed or strongly agreed that user interface widgets (menus, drop down lists etc.) were intuitive and user-friendly.

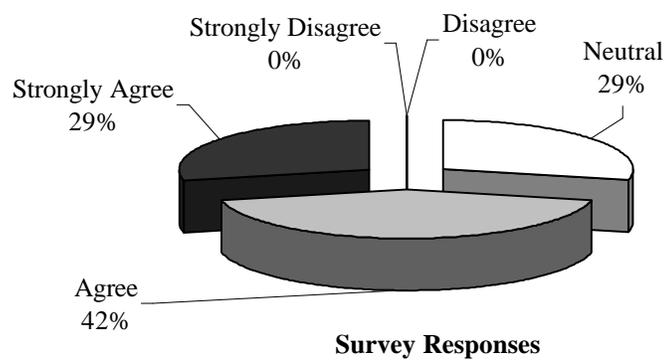


Figure 36: Use of menus (draggable menus, drop down menus etc.) was intuitive and user friendly.

Question 19. Use of color (e.g.: supports material, provides interest, appeal) was appropriate and consistent.

As indicated in Figure 37, all the respondents either agreed or strongly agreed that the user interface was consistent in the use of color throughout the applications.

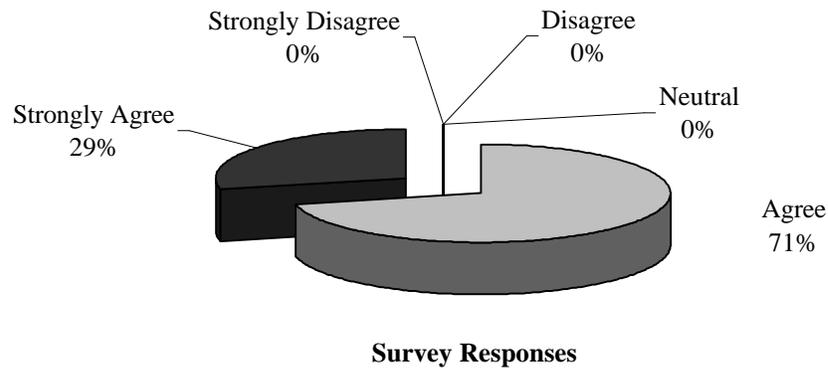


Figure 37: Use of color (e.g., supports material; provides interest, appeal) was appropriate and consistent.

Question 20. The application status was clearly visible at all times (the current state of the application, such as loading interface, loading images etc.).

As indicated in Figure 38, approximately 70% of respondents either agreed or strongly agreed that user interface widgets (menus, drop down lists etc.) were intuitive and user-friendly.

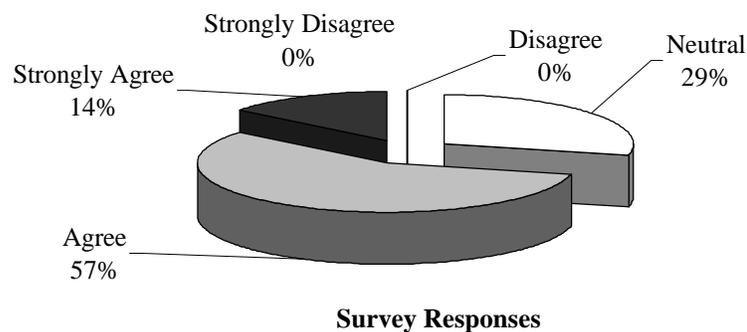


Figure 38: The application status was clearly visible at all times (the current state of the application, such as loading interface, loading images etc.).

Question 21. The symbols used in the application matched existing practices in the design studio (eg.: icons, nomenclature etc.).

As indicated in Figure 39, approximately 86% of respondents agreed that the user interface design for COADS used symbols and icons consistent with existing practices in the design studio.

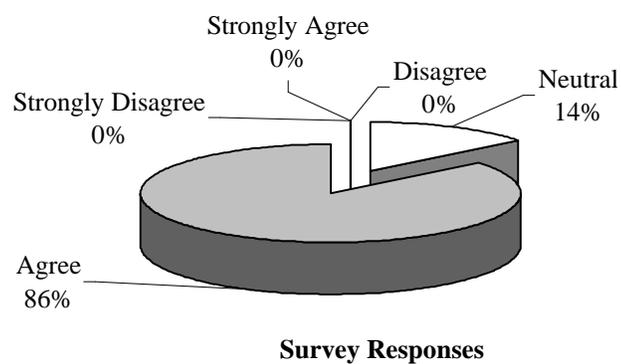


Figure 39: The symbols used in the application matched existing practices in the design studio (eg.: icons, nomenclature etc.).

Question 22. User control & rights were clearly specified and adequate for my role as a student.

As indicated in Figure 40, approximately 86% of respondents agreed that the user interface design for COADS used symbols and icons consistent with existing practices in the design studio.

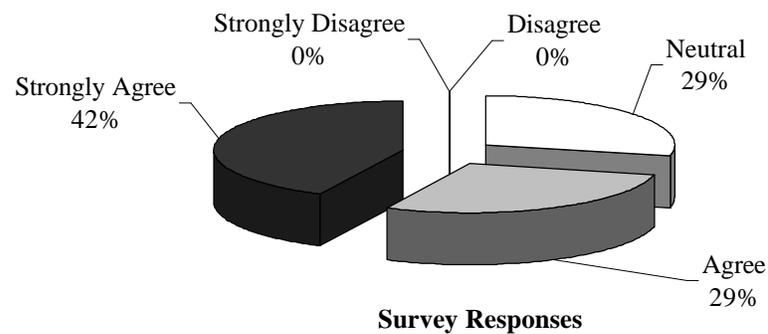


Figure 40: User control & rights were clearly specified and adequate for my role as a student.

Question 23. The user-interface is consistent with standard applications that I use (eg.: Internet Browsers, graphic applications etc.).

As indicated by Figure 41, approximately 86% of respondents agreed that the user interface design for COADS used symbols and icons, consistent with standard applications regularly used by the participants.

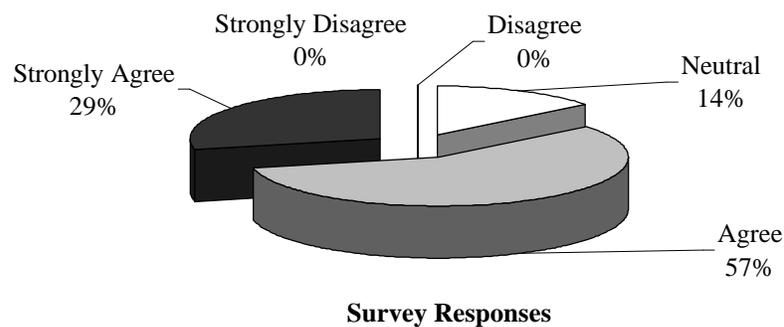


Figure 41: The user-interface is consistent with standard applications that I use (eg.: Internet Browsers, graphic applications etc.).

Question 24. The user-interface is appealing & aesthetically pleasing to use.

As indicated by Figure 42, approximately 86% of respondents agreed that the user interface design for COADS was appealing and aesthetically pleasing to use.

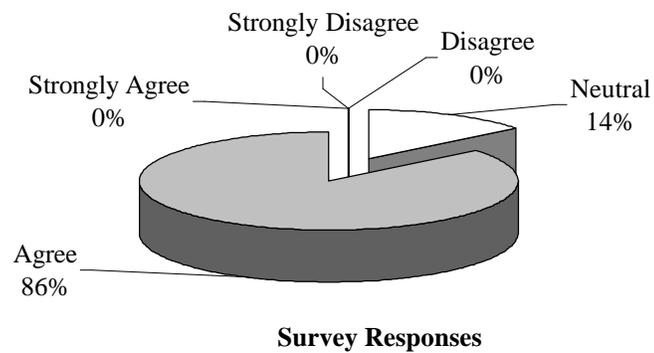


Figure 42: The user-interface is appealing & aesthetically pleasing to use.

Question 25. The user-interface design is minimalistic and does not interfere with the use of the application.

As indicated in Figure 43, approximately 71% of respondents agreed that the user interface design for COADS was not intrusive to the use of the application.

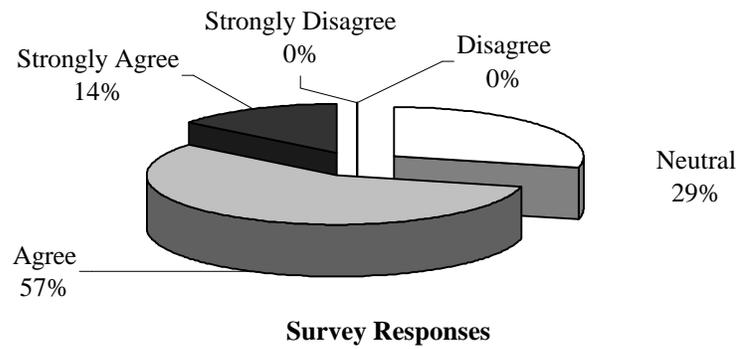


Figure 43: The user-interface design is minimalistic and does not interfere with the use of the application.

Question 26. The user-interface is easy to use.

As indicated in Figure 44, approximately 84% of respondents either agreed or strongly agreed that the user interface COADS was easy to use.

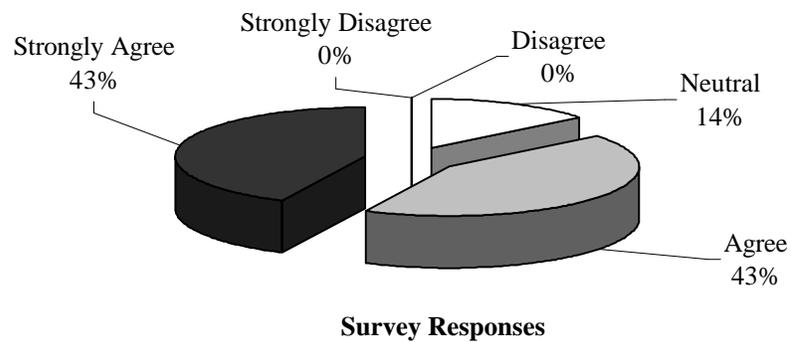


Figure 44: The user-interface is easy to use.

Question 27. The system and error messages are clear in their meaning.

As indicated in Figure 45, approximately 29% of respondents agreed and 57% were undecided that system and error messages were clear in their meaning. When considered in conjunction with Question 28, this suggests that though error messages and system status messages were adequately provided, their meaning was not clear enough to the participants.

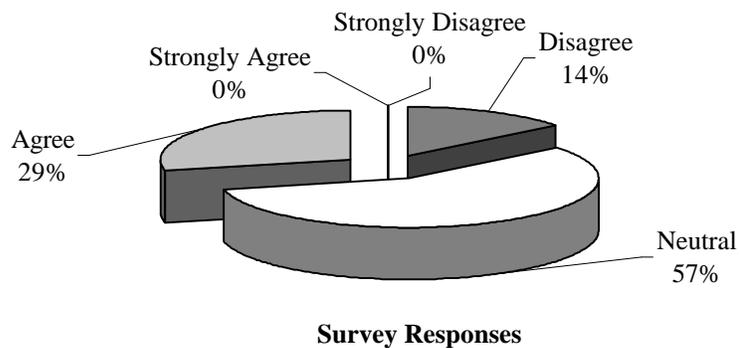


Figure 45: The system and error messages are clear in their meaning.

Question 28. The system and error messages are adequately provided.

As indicated in Figure 46, approximately 43% of respondents agreed and 43% were undecided that system and error messages were adequately provided.

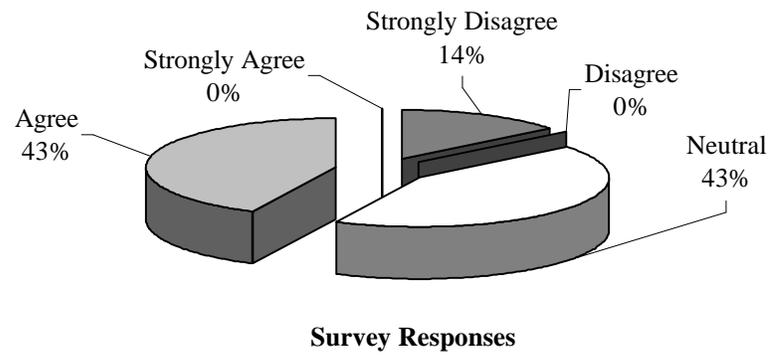


Figure 46: The system and error messages are adequately provided.

Question 29. Online help and documentation are easy to use.

As indicated in Figure 47, approximately 57% of respondents agreed and 42% disagreed that online help and documentation was easy to use.

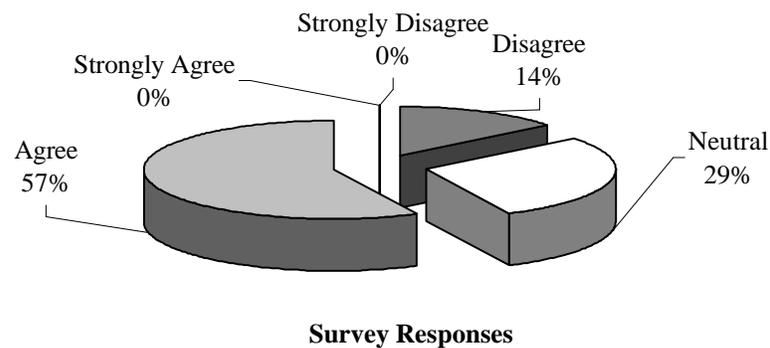


Figure 47: Online help and documentation are easy to use.

Question 30. Online help and documentation provide adequate information about all areas of the application.

As indicated in Figure 48, approximately 57% of respondents agreed and 42% disagreed that online help and documentation was adequately provided to cover all areas of the application. Responses to Questions 27 – 29 suggest that the help and documentation system for COADS needs to be implemented in more comprehensive and accessible manner.

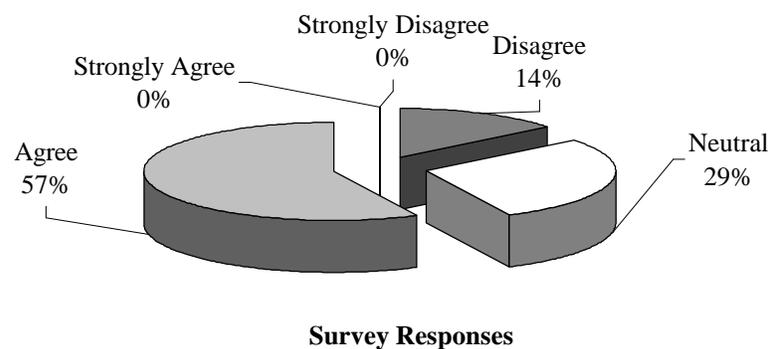


Figure 48: Online help and documentation provide adequate information about all areas of the application.

Question 31. Overall, how would you rate this application as an effective communication solution for the architectural design studio.

As indicated in Figure 49, all the respondents agreed that COADS was an effective solution for communication in the design studio.

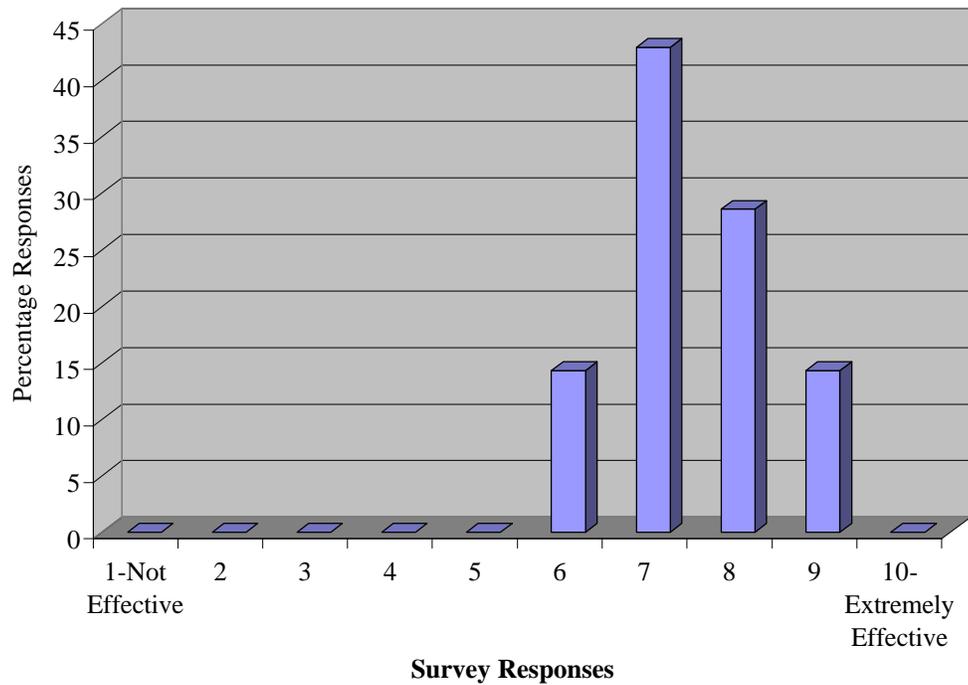


Figure 49: Overall, how would you rate this application as an effective communication solution for the architectural design studio.

Question 32. What features, if any, would you propose that might improve the collaborative design experience?

This question was presented to the respondents as a free-form comment . Most of the responses to this question dealt with improving the features of the whiteboard, notably the zoom & pan tool and a more user-friendly and intuitive erase/undo tool.

Question 33. In what ways, if any, do you believe that this application represents an important innovation or advance in architectural education?

All of the responses to this question indicated that the respondents felt COADS to be an important tool in virtual communication for the design studio. Most of them felt that COADS would be especially valuable as a distance learning tool, rather than as a tool to be used in the physical design studio which corresponds very well with the stated goals of COADS.

CHAPTER VII

CONCLUSION

Summarizing the evaluation of responses in Chapter VI, the usability testing of the Collaborative Online Design Studio (COADS) demonstrated that the prototype system can be a successful tool for remote collaboration in design studio. The system was successful in providing an integrated means for design collaboration with other participants, especially when the participants are remotely located.

The prototype successfully used current web-based technology to achieve an online collaborative environment that supports the following:

- Collaborative sketching overlaid on design studies – A collaborative whiteboard that supports real-time drawing and image information sharing.
- Real-time audio/video communications capabilities – Real-time audio and video streaming, that enhanced the collaborative experience.
- Controlled moderation during presentations – A role based control mechanism that allows for users to moderate online sessions in a manner similar to the functioning of the physical design studio.
- Presentation system – A collaborative presentation mechanism that allows users to deliver their individual slide-based presentations to all connected users in real-time.

- Support for different media types – A system that supports both static image media as well as slide based and keyframe based media.
- Easy access to online resources relevant to the design studio session – An archival system that allows for easy storage and access of secondary information relevant to a design studio.

The use of COADS presents various advantages and disadvantages. The main advantages are

- Users can receive immediate feedback about their designs from other participants.
- COADS provides users with an integrated single point access to all functionality necessary for remote collaboration in the design studio.
- COADS provides the users with a an environment that is familiar for the users of a physical design studio.
- COADS can provide improved design feedback and communication.
- COADS can be setup on end-user machines with relative ease.

The disadvantages of using COADS are:

- The limitations imposed on the system by the hardware available. The most prominent being the limitation imposed by the size of the monitor used to view the application. Though this can be resolved to some extent by implementing zoom and pan tools, the most effective solution remains investing in large screens that are currently expensive and not easily accessible to the typical user.

- The number of concurrent users of the system also affects the quality of the communication. Consistent with the problems of distance learning, a user group with more than five users results in a significant drop in the quality of collaboration. This can be resolved by implementing a system of virtual sessions within COADS that can be created by individual users to invite up to four users per room to participate in discussions.
- For production environments, software and hardware requirements on the hosting server tend to be high, requiring significant investment. Investment needs to be made both for software as well as hardware to be able to support the demands of multiple users, streaming audio video and text and data simultaneously.

Future Work

COADS affords various possibilities for future research work. These range from detailed testing on larger sample population, improvements in features and usability to exploring applicability in various related domains such as collaboration in engineering firms, healthcare and imaging and educational fields.

The areas for further research include

- Extensive testing on larger groups of sample population in remote locations using production servers.
- Improvements in the user interface design to provide better accessibility to resources.

- Improvements to the whiteboard toolset, especially the addition of erase tools and zoom and pan tools.
- Allow the creation of virtual rooms with support for upto five concurrent users per room.
- Explore the potential for additional collaborative design tools such as a collaborative 3D modeling environment.
- Archival features to document and store collaborative drawing sessions, audio, video and text data.

This research finds that it is possible to develop an integrated system for augmenting communication in the design studio through remote collaboration. However it is imperative that such a system be closely integrated with the coursework and be used in conjunction with the regular activities of the physical design studio.

REFERENCES

- [1] B.R. Lawson, *Design in Mind*, Butterworth Architecture, Oxford, 1994.
- [2] J.S. Busby, The neglect of feedback in engineering design organisations, *Design Studies* 19 (1998) 103-117.
- [3] G. Mark, S. Poltrock, Diffusion of a Collaborative Technology Across Distance, *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Groupwork* 6 (2001) 232-241.
- [4] R. Burton, *Report of the Steering Group on Architectural Education*, RIBA, London, 1992.
- [5] D. Cuff, *Architecture: The Story of Practice*, MIT Press, Cambridge, Massachusetts, 1991.
- [6] B.W. Boehm, A Spiral Model of Software Development and Enhancement, *IEEE Computer* 21 (2) (1988) 61-72.
- [7] R. Rubenstein, H.M. Hersh, H.F. Ledgard, *The Human Factor: Designing Computer Systems for People*, Digital Press, Burlington, Massachusetts, 1984.
- [8] R.L. Mack, N. Jakob, *Usability Inspection Methods: Executive Summary*, John Wiley & Sons, New York, 1994.
- [9] B.R. Lawson, *How Designers Think : The Design Process Demystified*, third ed., Architectural Press, London, 1997.
- [10] C. Green, *Gambit*, University of Sheffield, Sheffield, England, 1977.

- [11] B. Uluoglu, Design knowledge communicated in studio critiques, *Design Studies* 21 (2000) 33-58.
- [12] C. Peng, Exploring communication in collaborative design: co-operative architectural modelling, *Design Studies* 15 (1) (1994) 19-44.
- [13] N. Cross, Creativity in design: not leaping but bridging, in: *Creativity and Cognition Second International Symposium*, Loughborough, England, 1996.
- [14] T. Porter, *Designer Primer for Architects, Graphic Designers and Artists*, Butterworth Architecture, London, 1988.
- [15] J. Stempfle, P. Badke-Schaub, Thinking in design teams - an analysis of team communication, *Design Studies* 23 (2002) 473-496.
- [16] D.L. Craig, C. Zimring, Supporting collaborative design groups as design communities, *Design Studies* 21 (2000) 187-204.
- [17] J. Sigle, Synchronous distance learning via the Internet - experience and status report, *Journal for Computing in Small Colleges* 16 (4) (2001) 92-98.
- [18] T.A. Finholt, E. Rocco, D. Bree, N. Jain, NotMeeting: A field trial of NetMeeting in a geographically distributed organization, *SigGroup Bulletin* 21 (1) (1998) 66-69.
- [19] M.W. Goldberg, S. Salari, P. Swoboda, World Wide Web - course tool: an environment for building WWW-based courses, *Computer Networks* 28 (1996) 1219-1231.

- [20] B. Shneiderman, A taxonomy and rule base for the selection of interaction styles, Human Factors for Informatics Usability, Cambridge University Press, Cambridge, 1991.
- [21] A. Marcus, Principles of effective visual communication for graphical user interface design. UnixWorld, (August 1990) 121-124.

APPENDIX A

SCRIPTS

Main Interface scripts

```

/*=====//
  Desc: Main Shell Interface
        draws initial interfaces
        loads secondary interfaces
  Filename: coads.fla
  Layer: Actions
  Frame: Init
  Author: Abey George
  Contact: abeymg@tamu.edu
  //=====*/
  //=====//
  //Start Script////////////////////////////////////
  //=====//
  #include "D:/Data/Flash/AS/MovieClipLoader.as"

  // GLOBAL VARIABLES //
  var depth = 0; // level variable
  var TOP_NAMEBAR_HEIGHT = top_nameBar_mc._height; // height of top
label bar
  var resizeHandler = new Object(); // object to handle stage
resizes
  var MENU_LEVEL = 100;

  // GENERAL METHODS //
  getMovie = function(clip,lvl,message){
    _root.status_mc.display_txt.htmlText = message;
    mcl.loadClip(lvl,clip);
  }
  // unstick all buttons on welcome_mc
  enableButtons = function(){
    for(i=1;i<5;i++){
      var ptr = eval("welcome_mc.btn"+i+"_mc");
      var exclude_ptr = eval("welcome_mc."+arguments[0]);
      if(ptr == exclude_ptr){
        ptr.enabled = false;
      }
      else {
        ptr.enabled = true;
        ptr.gotoAndStop(1);
      }
    }
  }
  //INITIALIZE MENU ITEMS//

```

```

        init_menu = function(){
            // initialize top menu //
            _root.attachMovie("end_mc", "topL_end_mc", depth++); // top
menu left corner
            _root.attachMovie("end_mc", "topR_end_mc", depth++); // top
menu right corner
            topR_end_mc._xscale *=-1;
            _root.attachMovie("FPushButtonSymbol", "filler_pb", depth++);
// top menu filler
            filler_pb.setEnabled(false);
            // initialize bottom menu //
            _root.attachMovie("end_mc", "botL_end_mc", depth++); // top
menu left corner
            _root.attachMovie("end_mc", "botR_end_mc", depth++); // top
menu right corner
            botR_end_mc._xscale *=-1;
            _root.attachMovie("FPushButtonSymbol", "bot_filler_pb", depth
++); // bot menu filler
            bot_filler_pb.setEnabled(false);
            // initialize status bar
            _root.attachMovie("status", "status_mc", depth++); // top
menu left corner

        }
        // INITIALIZE WELCOME PANEL //
        init_welcome = function(){
            welcome_mc.btn1_mc.label_txt.text = "INTRO";
            welcome_mc.btn1_mc.onPress = function(){
                enableButtons(this._name);
                this.enabled = false;
            }
            welcome_mc.btn2_mc.label_txt.text = "QUICKSTART";
            welcome_mc.btn2_mc.onPress = function(){
                enableButtons(this._name);
                this.enabled = false;
            }
            welcome_mc.btn3_mc.label_txt.text = "LOGIN";
            welcome_mc.btn3_mc.onPress = function(){
                getMovie("Login.swf", MENU_LEVEL, "Loading Login Interface
...");

                enableButtons(this._name);
                this.enabled = false;
            }
            welcome_mc.btn4_mc.label_txt.text = "HELP";
            welcome_mc.btn4_mc.onPress = function(){
                enableButtons(this._name);
                this.enabled = false;
            }
        }
        // MAIN INIT FUNC //
        init = function(){
            // stage setup
            Stage.scaleMode = "noScale";
            Stage.align = "TL";

```

```

Stage.showMenu = false;
Stage.addListener(resizeHandler);

    // initialize MovieLoaderClass //
    listener = {};
listener.onLoadComplete = function(result, target, url){
    trace("onLoadComplete " + arguments);
    preloader_mc.slider_mc.gotoAndStop(1);
    preloader_mc.display_txt.text = "";
    _root.status_mc.display_txt.htmlText = "";
    for(var i=0;i<Stage._listeners.length;i++){
        Stage._listeners[i].onResize();
    }
}
listener.onLoadProgress = function(bytesLoaded, bytesTotal,
kbLoaded, kbTotal, percentLoaded, bytesPerSecond){
    trace("onLoadProgress " + percentLoaded);
    preloader_mc.slider_mc.gotoAndStop(percentLoaded);
    preloader_mc.display_txt.text = percentLoaded+"%";
    preloader_mc.dial_mc._rotation = 18*percentLoaded;
}
mcl = new MovieClipLoader();
mcl.addListener(listener);

    //
    init_menu();
    init_welcome();
    resizeHandler.onResize();
}
////////////////////////////////////
// HANDLE STAGE RESIZE //
resizeHandler.onResize = function(){
    var WIDTH = Stage.width;
    var HEIGHT = Stage.height;
    if(Stage.width <= 800){
        WIDTH = 800;
    }
    if(Stage.height <= 600){
        HEIGHT = 600;
    }
}

//locate top name bar//
top_nameBar_mc._x = 0;
top_nameBar_mc._y = 0;
top_nameBar_mc._width = WIDTH;
top_nameBar_mc._height = TOP_NAMEBAR_HEIGHT;
// locate logo //
coadsLogo_mc._x = 0;
coadsLogo_mc._y = (top_nameBar_mc._height -
coadsLogo_mc._height)/2;
//locate top menu//
topL_end_mc._x = 0;
topL_end_mc._y = TOP_NAMEBAR_HEIGHT;
topR_end_mc._x = WIDTH;

```

```

topR_end_mc._y = TOP_NAMEBAR_HEIGHT;
clock_mc._x = topL_end_mc._width;
clock_mc._y = TOP_NAMEBAR_HEIGHT;
filler_pb._x = clock_mc._x + clock_mc._width;
filler_pb._y = TOP_NAMEBAR_HEIGHT;
filler_pb.setSize(WIDTH - (2*topL_end_mc._width +
clock_mc._width),20);
//locate bottom menu//
var BOT_HEIGHT = HEIGHT - topL_end_mc._height;
botL_end_mc._x = 0;
botL_end_mc._y = BOT_HEIGHT;
botR_end_mc._x = WIDTH;
botR_end_mc._y = BOT_HEIGHT;

// locate preloader
preloader_mc._x = botL_end_mc._width;
preloader_mc._y = BOT_HEIGHT;

// locate bottom filler
bot_filler_pb._x = preloader_mc._x + preloader_mc._width;
bot_filler_pb._y = BOT_HEIGHT;
bot_filler_pb.setSize(WIDTH-
(preloader_mc._width+(2*botL_end_mc._width)),topL_end_mc._height);

// locate status bar
status_mc._x = preloader_mc._x + preloader_mc._width;
status_mc._y = BOT_HEIGHT;
status_mc._width = WIDTH-
(preloader_mc._width+(3.5*botL_end_mc._width));

// locate welcome pane
welcome_mc._x = (WIDTH - welcome_mc._width)/2;
welcome_mc._y = (HEIGHT- welcome_mc._height)/2;

// locate any sub movies on level 100
_level100._root.subWindow_mc._x = welcome_mc._x;
_level100._root.subWindow_mc._y = welcome_mc._y;
}
////////////////////////////////////
init();
stop();
//=====//
////////////////////////////////////End Script////////////////////////////////////
//=====//

/*=====//
Desc: Main Shell Interface
      draws initial student interface
      loads student interface
Filename: coads.fla

```

```

Layer: Actions
Frame: Student
Author: Abey George
Contact: abeymg@tamu.edu
//=====*/
//=====//
////////////////////Start Script////////////////////
//=====//

var resizeHandlerStudent = new Object(); // object to handle
stage resizes
init_student_menu = function(){
    // setup menu //
    menuStudent_mc.upload_pb.setClickHandler("onUpload",_root);
    menuStudent_mc.resources_pb.setClickHandler("onResources",_
root);

}
init_student = function(){
    init_student_menu();
    //setup Stage //
    Stage.scaleMode = "noScale";
    Stage.align = "TL";
    Stage.showMenu = false;
    Stage.addListener(resizeHandlerStudent);

    // load student movie //
    _root.getMovie("StudentInterface.swf",100,"Loading Student
Interface ...");
    resizeHandlerStudent.onResize();
}

//////////////////// EVENT HANDLERS //////////////////////
resizeHandlerStudent.onResize = function(){
    WIDTH = Stage.width;
    HEIGHT = Stage.height;
    if(Stage.width <= 835){
        WIDTH = 835;
    }
    if(Stage.height <= 600){
        HEIGHT = 600;
    }

    // locate menu //
    _root.menuStudent_mc._x = _root.clock_mc._x +
_root.clock_mc._width;
    _root.menuStudent_mc._y = _root.topL_end_mc._y;
    _root.filler_pb._x = _root.menuStudent_mc._x +
_root.menuStudent_mc._width;
    _root.filler_pb._width = WIDTH -
(2*_root.topL_end_mc._width + _root.clock_mc._width) -
_root.menuStudent_mc._width;
}

```

```

        onUpload = function(){
            if(Session.connected){

getURL(" javascript:openWindow('uploadStudent.asp?path="+Session.uName+"
')");
            }
        }
        onResources = function(){
            if(Session.connected){
                getMovie("resources_win.swf",110,"Loading Resources
Interface ...");
            }
        }

////////////////////////////////////
init_student();
stop();
//=====//
////////////////////////////////////End Script////////////////////////////////////
//=====//

/*=====//
Desc: Main Shell Interface
       draws initial faculty interface
       loads faculty interfaces
Filename: coads.fla
Layer: Actions
Frame: Faculty
Author: Abey George
Contact: abeymg@tamu.edu
//=====*/
//=====//
////////////////////////////////////Start Script////////////////////////////////////
//=====//

var resizeHandlerFaculty = new Object(); // object to handle
stage resizes
init_faculty_menu = function(){
    // load menu //
    //setup faculty menu
    menuFaculty_mc.upload_pb.setClickHandler("onUpload",_root);
    menuFaculty_mc.resources_pb.setClickHandler("onResources",_
root);
}
init_faculty = function(){
    init_faculty_menu();
    //setup Stage //
    Stage.scaleMode = "noScale";
    Stage.align = "TL";

```

```

Stage.showMenu = false;
Stage.addListener(resizeHandlerFaculty);

// load student movie //
_root.getMovie("FacultyInterface.swf",100,"Loading Faculty
Interface ...");
resizeHandlerFaculty.onResize();
}
////////// EVENT HANDLERS //////////
resizeHandlerFaculty.onResize = function(){
WIDTH = Stage.width;
HEIGHT = Stage.height;
if(Stage.width <= 835){
WIDTH = 835;
}
if(Stage.height <= 600){
HEIGHT = 600;
}

// locate menu //
_root.menuFaculty_mc._x = _root.clock_mc._x +
_root.clock_mc._width;
_root.menuFaculty_mc._y = _root.topL_end_mc._y;
_root.filler_pb._x = _root.menuFaculty_mc._x +
_root.menuFaculty_mc._width;
_root.filler_pb._width = WIDTH -
(2*_root.topL_end_mc._width + _root.clock_mc._width) -
_root.menuFaculty_mc._width;
}

onUpload = function(){
if(Session.connected){

getURL("javascript:openWindow('uploadFaculty.asp?path="+Session.uName+"
')");
}
}
onResources = function(){
if(Session.connected){
getMovie("resources_win.swf",110,"Loading Resources
Interface ...");
}
}

//////////
init_faculty();
stop();
//=====//
//////////End Script//////////
//=====//

```

Common Interface scripts

```

/*=====//
  Desc: Common Interface Communication Scripts
        handles communication method common to both
        Subscriber and Moderator interfaces
  Filename: fcs_common.as
  Layer: N/A
  Frame: N/A
  Author: Abey George
  Contact: abeymg@tamu.edu
  //=====*/
  //=====//
  ///////////////////////////////////////////////////////////////////Start Script/////////////////////////////////////////////////////////////////
  //=====//

init_connection = function(){
  nc = new NetConnection();
  nc.owner = this;
  nc.onStatus = function(info){
    trace("LEVEL: "+info.level+" CODE: "+info.code);
    if(info.code == "NetConnection.Connect.Success"){
      Session.connected = true;
      Session.connection = this;
      start_connection();
    }
    else if(info.code == "NetConnection.Connect.Rejected"){
      Session.connected = false;
      _level0.status_mc.display_txt.htmlText = "<font
color=\">#FF0000\ ">Connection rejected by server. Please try again
later.</font>";
    }
  }
  nc.connect(FCS_URL,Session);
}

start_connection = function(){
  connection_mc.conn_lt.connect(nc);
  connection_mc.conn_bw.connect(nc);
  conn_uc.connect(nc);
  wb_mc.connect(nc);
  cursor_mc.connect(nc);
  presContainer_mc.connect(nc);
  chatPanel_mc.chat.connect(nc);
  chatPanel_mc.ucolor.connect(nc);
  initStreamPublish();
  if(Session.uType == "Student"){
    connectToFaculty();
  }
}

```

```

    getUserList();
    getQueue();
    getPresentation();
    refreshFiles();
    getPresenter();
}

initStreamPublish = function(){
    if(myCam == undefined){
        myCam = Camera.get();
        myMic = Microphone.get();
    }
    else{
        if(myCam.muted){
            System.showSettings(0);
        }
        System.onStatus = function(info){
        }
        startStreamPublish();
    }
    myCam.onStatus = function(info){
        if(info.code == "Camera.Muted"){
            return false;
        }
        else{
            startStreamPublish();
        }
    }
    myPanel_mc.the_video.attachVideo(myCam);
}

startStreamPublish = function(){
    if(publish_ns == undefined){
        publish_ns = new NetStream(nc);
    }
    publish_ns.attachAudio(myMic);
    publish_ns.attachVideo(myCam);
    if(Session.uType == "Faculty"){
        publish_ns.publish(Session.uType);
        //var temp_so =
SharedObject.getRemote("StudentSO",nc.uri,true);
        //temp_so.connect(nc);
        //temp_so.send("connectToFaculty");
    }
    else{
        publish_ns.publish(Session.uName);
    }
}

stopStreamPublish = function(){
    if(publish_ns == undefined){
        return;
    }
    myPanel_mc.the_video.attachVideo(null);
}

```

```

        myPanel_mc.the_video.clear();
        publish_ns.close();
        delete publish_ns;
    }

    resumeStreamPublish = function(){
        if(!Session.connected){
            return false;
        }
        else{
            initStreamPublish();
        }
    }

    getUserList = function(){
        users_so = SharedObject.getRemote("users",nc.uri);
        users_so.onSync = updateUserList;
        users_so.connect(nc);

        users_dp = new DataProviderClass();
        peopleList_mc.queue_lb.setDataProvider(users_dp);
    }

    updateUserList = function(list){
        users_dp.removeAll();
        var len = users_so.data.userList.length;
        for(var i=0;i<len;i++){
            var ref = users_so.data.userList[i];
            if(ref.uName == Session.uName){
                continue;
            }
            users_dp.addItem({label:ref.fName + " " +
ref.lName,data:ref.uName});
        }
    }

    showPeopleCam = function(){
        var selected = this.peopleList_mc.queue_lb.getSelectedItemAt();
        if(selected == undefined){
            return;
        }
        var name = selected.label;
        var uname = selected.data;

        if(sub_ns == undefined){
            sub_ns = new NetStream(nc);
        }
        peopleList_mc.name_txt.text = name;
        peopleList_mc.the_video.attachVideo(sub_ns);
        sub_ns.play(uname,-2,-1);
    }

    clearPeopleCam = function(){
        peopleList_mc.the_video.attachVideo(null);
    }

```

```

        peopleList_mc.the_video.clear();
        peopleList_mc.name_txt.text = "";
        sub_ns.close();
        delete sub_ns;
    }

    // remove user's file the common queue
    removeQueue = function(){
        // return if no file selected in the file list
        var controller = (Session.uType == "Faculty") ?
facultyController_mc : studentController_mc;

        var selected = controller.queue_lb.getSelectedItemAt();
        if(selected == undefined){
            return false;
        }
        // see if file belongs to current user
        var uname = selected.label.split(":")[0];
        if(Session.uType == "Student"){
            if(uname != Session.uName){
                return false;
            }
        }
        var index = controller.queue_lb.getSelectedIndex();
        controller.queue_lb.removeItemAt(index);
        var temp_so = SharedObject.getRemote("queue",nc.uri,true);
        var temp_array = temp_so.data.fileList;
        var len = temp_array.length;
        for(var i=0;i<len;i++){

if((temp_array[i].uname+"."+temp_array[i].fname)==selected.label){
            temp_array.splice(i,1);
            break;
        }
    }

        }

        getPresentation = function(){
            presentation_so =
SharedObject.getRemote("presentation",nc.uri,true);
            presentation_so.onSync = updatePresentation;
            presentation_so.connect(nc);
        }

        updatePresentation = function(){
            var presState = presentation_so.data.presState;
            var wbState = presentation_so.data.wbState;
            debug_txt.text = presState;
            presContainer_mc._visible = presState;
            wb_mc._visible = wbState;
        }

        // on starting connection get list of files in queue
        getQueue = function(){

```

```

        var controller = (Session.uType == "Faculty") ?
facultyController_mc : studentController_mc;
        queue_so = SharedObject.getRemote("queue",nc.uri,true);
        queue_so.onSync = updateQueue;
        queue_so.connect(nc);

        queue_dp = new DataProviderClass();
        controller.queue_lb.setDataProvider(queue_dp);
    }

    // sync queue shared object
    updateQueue = function(list){
        // update file queue
        queue_dp.removeAll();
        var len = queue_so.data.fileList.length;
        for(var i=0;i<len;i++){
            var ref = queue_so.data.fileList[i];
            queue_dp.addItem({label:ref.uname + ":" +
ref.fname,data:ref.url});
        }
        // update presenter info
        if(queue_so.data.presenter.name == "empty"){
            clearDiscussion();
            disconnectStudent();
        }
        else{
            launchDiscussion();
            connectToStudent();
        }
    }

    // connect to presenter cam
    connectToStudent = function(){
        if(student_ns == undefined){
            student_ns = new NetStream(nc);
        }
        studentPanel_mc.the_video.attachVideo(student_ns);
        studentPanel_mc.uname_txt.text =
queue_so.data.presenter.name;
        student_ns.play(queue_so.data.presenter.name,-2,-1);
    }

    disconnectStudent= function(){
        if(student_ns == undefined){
            return false;
        }
        studentPanel_mc.uname_txt.text = "UserName";
        studentPanel_mc.the_video.clear();
        studentPanel_mc.the_video.attachVideo(null);
        student_ns.close();
        delete student_ns;
    }

```

```

// launch discussion
launchDiscussion = function(){
    _level100.wb_mc._visible = true;
    _level100.wbControl_mc._visible = true;
    _level100.imageContainer_mc._visible = true;

_level0.getMovie(queue_so.data.presenter.url,_level100.imageContainer_m
c,"Loading Image ...");

    _level100.debug_txt.text += "presenter name:" +
queue_so.data.presenter.name + newline;
    _level100.debug_txt.text += "presenter url:" +
queue_so.data.presenter.url + newline;
}

// clear Discussion
clearDiscussion = function(){
    //if(_level100.imageContainer_mc._visible){
        _level100.imageContainer_mc.unloadMovie();
    //}
    clearStudentCam();
}

//=====//
//////////End Script//////////
//=====//

```

Subscriber Interface scripts

```

/*=====//
Desc: Subscriber Interface
        draws subscriber interfaces
        handles subscriber communications methods
Filename: studentInterface.fla
Layer: Actions
Frame: 1
Author: Abey George
Contact: abeymg@tamu.edu
//=====*/
//=====//
//////////Start Script//////////
//=====//

#include "NetDebug.as"
#include "_includes/fcs_common.as"

// Global Variables //
// rtmp://coads.no-ip.org/COADS/myInstance
status = _level0.status_mc.display_txt;

```

```

FCS_URL = "rtmp:/COADS/myInstance";
DEBUG = false;
_global.speakerMode = false;

var main_ptr = eval(_level0);
var resizeHandlerStudent = new Object();

//////////===== FCS =====//////////
// refresh uploaded file list for student
refreshFiles = function(){
    if(files_dp == undefined){
        files_dp = new DataProviderClass();

studentController_mc.files_lb.setDataProvider(files_dp);
    }
    var send_lv = new LoadVars();
    send_lv.uname = Session.uName;
    send_lv.type = "_images";
    var get_xml = new XML();
    get_xml.ignoreWhite = true;
    get_xml.onLoad = function(success){
        if(success){
            files_dp.removeAll();
            var file_array = this.firstChild.childNodes;
            for(var i=0;i<file_array.length;i++){

files_dp.addItem({label:file_array[i].firstChild.nodeValue,data:file_ar
ray[i].attributes.Url});
                                //status.htmlText =
file_array[i].firstChild.nodeValue;
                                }
                                }
                                else{
status.htmlText = "<font color=\"#FF0000\">No Files
Retrieved!</font>";
                                }
                                }
                                send_lv.sendAndLoad("getFiles.asp",get_xml);
        }
// add files from user's file list to the common queue
addQueue = function(){
    // return if no file selected in the file list
    var selected =
studentController_mc.files_lb.getSelectedItemAt();
    if(selected == undefined){
        return;
    }

    // see if file already in queue //
    var exists = false;
    var index =
studentController_mc.files_lb.getSelectedIndex();
    var fileName = selected.label;
    var len = studentController_mc.queue_lb.getLength();

```

```

        for(var i=0;i<len;i++){
            var temp = studentController_mc.queue_lb.getItemAt(i);
            var tempName = temp.label;
            if((Session.userName+":"+fileName) == tempName){
                exists = true;
                break;
            }
        }
        // if file exists return //
        if(exists){
            return;
        }
        // else update shared object //
        else{
            var temp_so =
SharedObject.getRemote("queue",nc.uri,true);
            var temp_array = temp_so.data.fileList;
            var temp_obj = new Object();
            temp_obj.uname = Session.userName
            temp_obj.fname = fileName;
            temp_obj.url = selected.data;
            temp_array.push(temp_obj);
        }
    }
    connectToFaculty = function(){
        if(teacher_ns == undefined){
            teacher_ns = new NetStream(nc);
        }
        teacherPanel_mc.the_video.attachVideo(teacher_ns);
        teacher_ns.play("Faculty",-2,-1);
    }

    disconnectFaculty = function(){
        if(teacher_ns == undefined){
            return false;
        }
        teacherPanel_mc.the_video.clear();
        teacherPanel_mc.the_video.attachVideo(null);
        teacher_ns.close();
        delete teacher_ns;
    }

    // removeQueue()
    // getQueue()
    // updateQueue()
    // clearPeopleCam()
    // showPeopleCam()
    // getUserList()
    // updateUserList()
    // resumeStreamPublish()
    // stopStreamPublish()
    // startStreamPublish()
    // initStreamPublish()
    // start_connection()

```

```

// init_connection()
//////////==== End =====//

wbControl_mc.undo_btn.onRelease = function(){
    _level100.wb_mc.undo();
}
wbControl_mc.clear_btn.onRelease = function(){
    _level100.wb_mc.clearPad();
}
wbControl_mc.change_cb.setChangeHandler("change",_root);

change = function(obj){
    var t = obj.getSelectedIndex();
    wb_mc.setThickness(t);
}

////////// INITIALIZE INTERFACE MENU //////////
init_interface = function(){
    // setup button labels //
    studentPanel_mc.display_txt.text = "Student Camera";
    teacherPanel_mc.display_txt.text = "Faculty Camera";
    myPanel_mc.display_txt.text = "My Camera";

    // locate whiteboard //
    wb_mc._x = 0;
    wb_mc._y = 66.3;

    // locate presentation Container //
    presContainer_mc._x = 0;
    presContainer_mc._y = 66.3;

    // locate image holder //
    imageContainer_mc._x = wb_mc._x;
    imageContainer_mc._y = wb_mc._y;

    myPanel_mc.close_btn.onPress = function(){
        this._parent._parent.stopStreamPublish();
        this._parent.nextFrame();
    }
    myPanel_mc.max_btn.onPress = function(){
        this._parent._parent.resumeStreamPublish();
        this._parent.prevFrame();
    }
    teacherPanel_mc.max_btn.onPress = function(){
        this._parent._parent.connectToFaculty();
        this._parent.prevFrame();
    }
    teacherPanel_mc.close_btn.onPress = function(){
        this._parent._parent.disconnectFaculty();
        this._parent.nextFrame();
    }
    studentPanel_mc.max_btn.onPress = function(){
        this._parent._parent.connectToStudent();
    }
}

```

```

        this._parent.prevFrame();
    }
    studentPanel_mc.close_btn.onPress = function(){
        this._parent._parent.disconnectStudent();
        this._parent.nextFrame();
    }
    chatPanel_mc.shared_btn.onPress = function(){
        cursor_mc.close();
        this.enabled = false;
        chatPanel_mc.shared_btn2.enabled = true;
        chatPanel_mc.shared_btn2.gotoAndStop(1);
    }
    chatPanel_mc.shared_btn2.onPress = function(){
        cursor_mc.connect(nc);
        this.enabled = false;
        chatPanel_mc.shared_btn.enabled = true;
        chatPanel_mc.shared_btn.gotoAndStop(1);
    }
    debug_txt._visible = DEBUG;
    debug_sb._visible = DEBUG;
    debug_lbl._visible = DEBUG;
}
//////////===== End =====//////////

////////// MAIN INIT FUNCTION //////////
init = function(){
    //_global.userType = "Student";
    init_interface();
    Stage.scaleMode = "noScale";
    Stage.align = "TL";
    Stage.showMenu = false;
    Stage.addListener(resizeHandlerStudent);
    init_connection();
    resizeHandlerStudent.onResize();
}

////////// EVENT HANDLERS //////////
// HANDLE STAGE RESIZES //
resizeHandlerStudent.onResize = function(){
    var WIDTH = Stage.width;
    var HEIGHT = Stage.height;
    if(Stage.width <= 835){
        WIDTH = 835;
    }
    if(Stage.height <= 600){
        HEIGHT = 600;
    }
    // locate local video panel //
    myPanel_mc._x = WIDTH - myPanel_mc._width;
    myPanel_mc._y = main_ptr.top_nameBar_mc._height +
main_ptr.topR_end_mc._height;

    // locate teacher video panel //

```

```

        teacherPanel_mc._x = myPanel_mc._x -
teacherPanel_mc._width;
        teacherPanel_mc._y = myPanel_mc._y;

        // locate student video panel //
        studentPanel_mc._x = teacherPanel_mc._x -
studentPanel_mc._width;
        studentPanel_mc._y = teacherPanel_mc._y;

        // locate chat panel
        chatPanel_mc._x = studentPanel_mc._x - chatPanel_mc._width;
        chatPanel_mc._y = studentPanel_mc._y;

        // locate connections panel

        connection_mc._x = chatPanel_mc._x - connection_mc._width;
        connection_mc._y = chatPanel_mc._y;

        // locate filler mc //
        filler_mc._x = 0;
        filler_mc._y = connection_mc._y;
        filler_mc._height = 22.9;
        filler_mc.bg_mc._width = connection_mc._x;
        filler_mc.right_mc._x = connection_mc._x;
        filler_mc.bot_mc._height = 1;
        filler_mc.bot_mc._width = filler_mc.top_mc._width =
filler_mc.bg_mc._width;

        // locate teacher control panel //
        if(!studentController_mc.moved){
            studentController_mc._x = WIDTH -
studentController_mc._width;
            studentController_mc._y = 0;
        }

        // locate peopleList control panel //
        if(!peopleList_mc.moved){
            peopleList_mc._x = WIDTH - 2*peopleList_mc._width;
            peopleList_mc._y = 0;
        }
        wbControl_mc._x = 0;
        wbControl_mc._y = filler_mc._y + filler_mc._height;

        imageContainer_mc._width = wb_mc._width;
        imageContainer_mc._height = wb_mc._height;
    }

    //////////////////////////////////////
    init();
    stop();
    //=====//
    //////////////////////////////////////End Script////////////////////////////////////
    //=====//

```

Moderator Interface scripts

```

/*=====//
  Desc: Moderator Interface
        draws moderator interfaces
        handles moderator communications methods
  Filename: moderatorInterface.fla
  Layer: Actions
  Frame: 1
  Author: Abey George
  Contact: abeymg@tamu.edu
  //=====*/
  //=====//
  ///////////////////////////////////////////////////////////////////Start Script/////////////////////////////////////////////////////////////////
  //=====//

#include "_includes/fcs_common.as"

// Global Variables //
status = _level0.status_mc.display_txt;
//rtmp://coads.no-ip.org/COADS/myInstance
FCS_URL = "rtmp://COADS/myInstance";
DEBUG = false;
_global.speakerMode = true;

var main_ptr = eval(_level0);
var resizeHandlerFaculty = new Object();

//GENERAL METHODS for Faculty //

discuss = function(){
  // clear presentation
  clearPres();

  var selected =
this.facultyController_mc.queue_lb.getSelectedItemAt();
  if(selected == undefined){
    return false;
  }
  var index = selected.label.indexOf(":");
  var uname = selected.label.slice(0,index);
  var fileName = selected.data;
  //update presenter info in shared object
  queue_so.data.presenter.name = uname;
  queue_so.data.presenter.url = fileName;
}

clearDisc = function(){
  queue_so.data.presenter.name = "empty";
  queue_so.data.presenter.url = "empty";
}

```

```

// launch presentation
launchPres = function(){
    // clear Discussion
    clearDisc();

    presentation_so.data.presState = true;
    presentation_so.data.wbState = false;
    var selected =
this.facultyController_mc.pres_lb.getSelectedItemAt();
    var fileName = "upLoaded/_presentations/"+selected.label;
    debug_txt.text = "blah" + this.presContainer_mc + fileName;
    if(selected == undefined){
        return false;
    }
    this.presContainer_mc.loadSWF(fileName);
}

clearPres = function(){
    presentation_so.data.presState = false;
    presentation_so.data.wbState = true;
}

wbControl_mc.undo_btn.onRelease = function(){
    _level100.wb_mc.undo();
    trace(this);
}

wbControl_mc.clear_btn.onRelease = function(){
    _level100.wb_mc.clearPad();
    trace(this);
}

wbControl_mc.change_cb.setChangeHandler("change",_root);
//change line wieth for whiteboard
change = function(obj){
    var t = obj.getSelectedIndex();
    trace(t);
    wb_mc.setThickness(t);
}

// clearChat History
clearChat = function(){
    chatPanel_mc.chat.clearHistory();
}

// refresh uploaded file list for student
refreshFiles = function(){
    if(files_dp == undefined){
        files_dp = new DataProviderClass();
    }

facultyController_mc.pres_lb.setDataProvider(files_dp);
}

var send_lv = new LoadVars();
send_lv.uname = Session.uName;
send_lv.type = "_presentations";
var get_xml = new XML();
get_xml.ignoreWhite = true;

```

```

get_xml.onLoad = function(success){
    if(success){
        files_dp.removeAll();
        var file_array = this.firstChild.childNodes;
        for(var i=0;i<file_array.length;i++){

files_dp.addItem({label:file_array[i].firstChild.nodeValue,data:file_ar
ray[i].attributes.Url});
//status.htmlText =
file_array[i].firstChild.nodeValue;
        }
    }
    else{
        status.htmlText = "<font color=\"#FF0000\">No Files
Retrieved!</font>";
    }
}
send_lv.sendAndLoad("getFiles.asp",get_xml);
}
////////// INITIALIZE INTERFACE MENU //////////
init_interface = function(){
    // setup button labels //
    studentPanel_mc.display_txt.text = "Student Camera";
    myPanel_mc.display_txt.text = "My Camera";

    // locate whiteboard //
    wb_mc._x = 0;
    wb_mc._y = 66.3;

    // locate presentation Container //
    presContainer_mc._x = 0;
    presContainer_mc._y = 66.3;

    // locate image holder //
    imageContainer_mc._x = wb_mc._x;
    imageContainer_mc._y = wb_mc._y;

myPanel_mc.close_btn.onPress = function(){
    this._parent._parent.stopStreamPublish();
    this._parent.nextFrame();
}
myPanel_mc.max_btn.onPress = function(){
    this._parent._parent.resumeStreamPublish();
    this._parent.prevFrame();
}
chatPanel_mc.shared_btn.onPress = function(){
    cursor_mc.close();
    this.enabled = false;
    chatPanel_mc.shared_btn2.enabled = true;
    chatPanel_mc.shared_btn2.gotoAndStop(1);
}
chatPanel_mc.shared_btn2.onPress = function(){
    cursor_mc.connect(nc);
    this.enabled = false;
}

```

```

        chatPanel_mc.shared_btn.enabled = true;
        chatPanel_mc.shared_btn.gotoAndStop(1);
    }
    debug_txt._visible = DEBUG;
    debug_sb._visible = DEBUG;
    debug_lbl._visible = DEBUG;
}

////////// MAIN INIT FUNCTION ////////////
init = function(){
    _global.userType = "Faculty";
    init_interface();
    Stage.scaleMode = "noScale";
    Stage.align = "TL";
    Stage.showMenu = false;
    Stage.addListener(resizeHandlerFaculty);
    init_connection();
    resizeHandlerFaculty.onResize();
}

////////// EVENT HANDLERS ////////////
// HANDLE STAGE RESIZES //
resizeHandlerFaculty.onResize = function(){
    var WIDTH = Stage.width;
    var HEIGHT = Stage.height;
    if(Stage.width <= 835){
        WIDTH = 835;
    }
    if(Stage.height <= 600){
        HEIGHT = 600;
    }

    // locate teacher video panel //
    myPanel_mc._x = WIDTH - myPanel_mc._width;
    myPanel_mc._y = main_ptr.top_nameBar_mc._height +
main_ptr.topR_end_mc._height;

    // locate student video panel //
    studentPanel_mc._x = myPanel_mc._x -
studentPanel_mc._width;
    studentPanel_mc._y = myPanel_mc._y;

    // locate chat panel
    chatPanel_mc._x = studentPanel_mc._x - chatPanel_mc._width;
    chatPanel_mc._y = studentPanel_mc._y;

    connection_mc._x = chatPanel_mc._x - connection_mc._width;
    connection_mc._y = chatPanel_mc._y;

    // locate filler mc //
    filler_mc._x = 0;
    filler_mc._y = connection_mc._y;
    filler_mc._height = 22.9;

```

```

        filler_mc.bg_mc._width = connection_mc._x;
        filler_mc.right_mc._x = connection_mc._x;
        filler_mc.bot_mc._height = 1;
        filler_mc.bot_mc._width = filler_mc.top_mc._width =
filler_mc.bg_mc._width;

        // locate teacher control panel //
        if(!facultyController_mc.moved){
            facultyController_mc._x = WIDTH -
facultyController_mc._width;
            facultyController_mc._y = 0;
        }

        // locate peopleList control panel //
        if(!peopleList_mc.moved){
            peopleList_mc._x = WIDTH - 2*facultyController_mc._width;
            peopleList_mc._y = 0;
        }

        imageContainer_mc._width = wb_mc._width;
        imageContainer_mc._height = wb_mc._height;
    }

    //////////////////////////////////////
    init();
    stop();
    //=====//
    //////////////////////////////////////End Script////////////////////////////////////
    //=====//

```

Login Interface scripts

```

/*=====//
    Desc: Login Interface
           draws login interfaces
           handles authentication and validation
    Filename: login.fla
    Layer: Actions
    Frame: 1
    Author: Abey George
    Contact: abeymg@tamu.edu
    //=====*/
    //=====//
    //////////////////////////////////////Start Script////////////////////////////////////
    //=====//

    // GLOABL VARIABLES //
    status = _level0.status_mc.display_txt;

```

```

URL = "authenticate.asp";
err = 0; // level for error messages
menu_array = new Array("accmenu_1_mc", "accmenu_2_mc");

// GENERAL METHODS //
show_hint = function(message){
    subWindow_mc.hint_mc._visible = true;
    subWindow_mc.hint_mc.display_txt.htmlText = message;
}
hide_hint = function(message, pos){
    subWindow_mc.hint_mc.display_txt.htmlText = "";
    subWindow_mc.hint_mc._visible = false;
}
show_errors = function(n){
    subWindow_mc.hint_mc._visible = false; // remove startup hint
message
    err++;
    var err_lvl = 10 + err; // start error movie clips at
level 11 //
    subWindow_mc.hint_mc.duplicateMovieClip("error_mc"+err_lvl,
err_lvl);
    var curr = eval("subWindow_mc.error_mc"+err_lvl)
    curr._visible = true;
    curr._y = eval("subWindow_mc."+n)._y;
    curr.display_txt.htmlText = errorList[n];
}
clear_errors = function(){
    var err_lvl;
    for(err;err>=0;err--){
        err_lvl = 10 + err;

eval("subWindow_mc.error_mc"+err_lvl).removeMovieClip();
    }
}
// INITIALIZE ACCORDION MENU //
arrangeMenu = function(){
    var len = menu_array.length;
    for(var i=1;i<len;i++){
        var cur = subWindow_mc[menu_array[i]];
        var prev = subWindow_mc[menu_array[i-1]];
        cur._y = prev._y + prev._height;
    }
}
openMenu = function(obj){
    //close the rest //
    var len = menu_array.length;
    for(var i=0;i<len;i++){
        if(obj._name!=menu_array[i]){
            var cur = subWindow_mc[menu_array[i]];
            cur.gotoAndStop(1);
            cur.open = false;
        }
    }
}

```

```

        // open this menu //
        obj.gotoAndStop(2);
        arrangeMenu();
    }
    closeMenu = function(obj){
        obj.gotoAndStop(1);
        arrangeMenu();
    }
    handleMenu = function(){
        if(this.open){
            this.open = false;
            closeMenu(this);
        }
        else{
            this.open = true;
            openMenu(this);
        }
    }
    initAccMenu = function(){
        var len = menu_array.length;
        for(var i=0;i<len;i++){
            var ref = subWindow_mc[menu_array[i]];
            ref.onPress = handleMenu;
        }
    }
    //INITIALIZE FORM ITEMS//
    init_form = function(){
        // array of hint messages
        hintList = [];
        hintList["email_txt"] = "Please enter your Email ID.";
        hintList["password_txt"] = "Please enter your
Password.<br>The Password is case sensitive.";

        // array of error messages
        errorList = [];
        errorList["email_txt"] = "<font
color=#FF0000><b>Invalid Email!</b></font><br>";
        errorList["password_txt"] = "<font
color=#FF0000><b>Invalid Password!</b></font><br>";

        // assign tabindex
        subWindow_mc.email_txt.tabIndex = 1;
        subWindow_mc.password_txt.tabIndex = 2;

        // assign hint messages to text fields
        for(var n in hintList){
            var field_ptr = eval("_root.subWindow_mc."+n);
            field_ptr.hint = hintList[n];
            field_ptr.onSetFocus = function(){
                clear_errors(); // clear any error messages //
                this.backgroundColor = 0xECFBBD;
                show_hint(this.hint);
                subWindow_mc.hint_mc._y = this._y;
            }
        }
    }

```

```

        field_ptr.onKillFocus = function(){
            this.backgroundColor = 0xeeeeee;
            hide_hint();
        }
    }
    // initialise submit button
    subWindow_mc.submit_mc.display_txt.text = "Submit";
    subWindow_mc.submit_mc.onRelease = onSubmit;

    // initialize accordion menu //
    initAccMenu();
}

// VALIDATE FORM //
validEmail = function(){
    var field_ptr = subWindow_mc.email_txt;
    var bad_entries;
    if(field_ptr.text == ""){
        bad_entries++;
    }
    if(field_ptr.text.indexOf("@") < 1){
        bad_entries++;
    }
    if(field_ptr.text.lastIndexOf(".") <=
(field_ptr.text.indexOf("@")+2)){
        bad_entries++;
    }
    if(field_ptr.text.length < 8){
        bad_entries++;
    }
    if(bad_entries >0){
        show_errors("email_txt");
        return false;
    }
    else return true;
}
validPassword = function(){
    if(subWindow_mc.password_txt.text == ""){
        show_errors("password_txt");
        return false;
    }
    else return true;
}
validateForm = function(){
    var bad_entries;
    if(!validEmail()){
        bad_entries++;
    }
    if(!validPassword()){
        bad_entries++;
    }
    if(bad_entries >0){
        return false;
    }
}

```

```

    }
    else return true;
}

// AUTHENTICATE USER //
authenticateUser = function(){
    status.htmlText = "Authenticating user ...";
    var sendData_lv = new LoadVars();
    sendData_lv.email = subWindow_mc.email_txt.text;
    sendData_lv.password = subWindow_mc.password_txt.text;
    var retData_lv = new LoadVars();
    retData_lv.owner = this;
    retData_lv.onLoad = function(success){
        if(success){
            if(this.authenticate == "true"){
                _global.Session = new Object();
                _global.Session.uName = this.uName;
                _global.Session.uType = this.uType;
                _global.Session.fName = this.fName;
                _global.Session.lName = this.lName;
                this.owner.status.htmlText = Session.uName +
Session.uType + Session.fName + Session.lName;
                _level100.unloadMovie();
                _level0._root.gotoAndStop(Session.uType);
                status.htmlText = "";
            }
            else{
                this.owner.status.htmlText = "<font
color=#FF0000>Invalid email/password combination.</font>";
                subWindow_mc.submit_mc.enabled = true;
            }
        }
        else{
            this.owner.status.htmlText = "<font
color=#FF0000>Login Failed! Please try again later.</font>";
            subWindow_mc.submit_mc.enabled = true;
        }
    }
    sendData_lv.sendAndLoad(URL,retData_lv);
}

// MAIN INIT FUNC //
init = function(){
    // initialize movie location //
    with(subWindow_mc){
        _x = _level0.welcome_mc._x;
        _y = _level0.welcome_mc._y;
        close_btn.onPress = function(){
            _level100.unloadMovie();
            _level0.enableButtons();
        }
    }
    // initialize form
    init_form();
}

```

```

// EVENT HANDLERS //
onSubmit = function(){
    if(validateForm()){
        subWindow_mc.submit_mc.enabled = false;
        authenticateUser();
    }
}
////////////////////////////////////
init();
stop();

//=====//
////////////////////////////////////End Script////////////////////////////////////
//=====//

```

Resources Interface scripts

```

/*=====//
Desc: Resources Interface
     draws resources interfaces
     handles resource management
Filename: resources_win.fla
Layer: Actions
Frame: 1
Author: Abey George
Contact: abeymg@tamu.edu
//=====*/
//=====//
////////////////////////////////////Start Script////////////////////////////////////
//=====//

status = _level0.status_mc.display_txt;
// GENERAL METHODS //
drawResources = function(){
    // delete previous clips
    this.scroll_sp.setScrollPosition(0,0);
    for(var i in this.resourceHolder_mc){
        this.resourceHolder_mc[i].removeMovieClip();
    }
    // createnew clips
    var ref = this.resourceHolder_mc;
    for(var i=0;i<resource_data.length;i++){
        var cur =
ref.attachMovie("resources_mc","resource"+i,depth++);
        cur._y = ref._y + i*(cur._height+5);
        cur.id = i;
        cur.num_txt.text = cur.id + 1;
        var format = new TextFormat();
        format.url = resource_data[i].link;
        format.target = "_blank";
        format.underline = true;
        cur.link_txt.text = resource_data[i].link;
        cur.link_txt.setTextFormat(format);

```

```

        cur.desc_txt.text = resource_data[i].desc;
        if(Session.uType == "Faculty"){
            cur.delete_btn._visible = true;
            cur.delete_btn.onPress = deleteResource;
        }
        else{
            cur.delete_btn._visible = false;
        }
    }
    this.scroll_sp.setSize(ref._width+20,260);
    this.scroll_sp.refreshPane();
}
addResource = function(){
    var temp_so =
SharedObject.getRemote("resources",Session.connection.uri,true);
    var temp_array = temp_so.data.resourceList;
    temp_obj = new Object();
    temp_obj.link = _root.resourceEntry.link_txt.text;
    temp_obj.desc = _root.resourceEntry.desc_txt.text;
    temp_array.push(temp_obj);
}
deleteResource = function(){
    var temp_so =
SharedObject.getRemote("resources",Session.connection.uri,true);
    var temp_array = temp_so.data.resourceList;
    var id = this._parent.id;
    if(id==0){
        return;
    }
    temp_array.splice(id,1);
}

initResources = function(){
    _level100.debug_txt.text = Session.connection.uri;
    resources_so =
SharedObject.getRemote("resources",Session.connection.uri,true);
    resources_so.onSync = updateResources;
    resources_so.connect(Session.connection);
}
updateResources = function(list){
    if(resource_data != undefined){
        delete resource_data;
    }
    resource_data = new Array();
    resource_data = resources_so.data.resourceList;
    drawResources();
}
//////////
init = function(){
    fader_mc._x = 0;
    fader_mc._y = 0;
    fader_mc._width = Stage.width;
    fader_mc._height = Stage.height;
    fader_mc.useHandCursor = false;
}

```



```
//=====//
//////////End Script//////////
//=====//
```

COADS Communication Server Application scripts

```
/*=====//
  Desc: COADS FCS Server-side script
        handles server side communication components
        handles shared objects
  Filename: main.asc
  Layer: N/A
  Frame: N/A
  Author: Abey George
  Contact: abeymg@tamu.edu
  //=====*/
//=====//
//////////Start Script//////////
//=====//

load("components.asc");
load("NetServices.asc");

// this func called when app is first started
application.onAppStart = function(){
  trace("starting application");
  this.clientList = new Object();
  this.userList = new Array();

  this.users_so = SharedObject.get("users");
  this.users_so.setProperty("userList",this.userList);

  this.queue_so = SharedObject.get("queue",true);
  this.queue_so.setProperty("fileList", new Array());
  var temp = new Object();
  temp.name = "empty";
  temp.url = "empty";
  this.queue_so.setProperty("presenter", temp);

  this.resources_so = SharedObject.get("resources",true);

  this.presentation_so = SharedObject.get("presentation",true);
  this.presentation_so.setProperty("presState",false);
  this.presentation_so.setProperty("wbState",true);

  /*
  var temp_array = new Array();
  temp_array.push({link:"http://www.coads.stuff", desc: "COADS
  Help Files"});
```

```

        this.resources_so.setProperty("resourceList", temp_array);
    */
}
// this func called when user connects
application.onConnect = function(newClient,session_obj){
    if(this.alreadyLoggedIn(session_obj.uName)){
        trace("User " +session_obj.uName+" Already Logged On");
        application.rejectConnection(newClient);
        return false;
    }
    else{
        gFrameworkFC.getClientGlobals(newClient).username =
session_obj.uName;
        this.acceptConnection(newClient);
        this.userList.push(session_obj);
        this.users_so.setProperty("userList", this.userList);
        trace("user: " + session_obj.uName + " connected");
    }
}

// this func called when user disconnects
application.onDisconnect = function(oldClient){
    var username =
gFrameworkFC.getClientGlobals(oldClient).username;
    trace("user: " + username + " disconnected");
    this.removeUser(oldClient);
    this.removeQueue(oldClient);
}

// helper functions
application.alreadyLoggedIn = function(uName){
    var match = false;
    for(var i=0;i<this.userList.length;i++){
        if(this.userList[i].uName == uName){
            match = true;
            break;
        }
    }
    trace("match->" + match);
    return match;
}
application.removeUser = function(oldClient){
    var username =
gFrameworkFC.getClientGlobals(oldClient).username;
    for(var i=0;i<this.userList.length;i++){
        if(this.userList[i].uName == username){
            // if user is faculty ...clear presentation & presenter
            if(this.userList[i].uType == "Faculty"){
                this.removePresenter();

                this.presentation_so.setProperty("presState",false);
                this.presentation_so.setProperty("wbState",true);
            }
            this.userList.splice(i,1);
        }
    }
}

```

```

        this.users_so.setProperty("userList",this.userList);
        return true;
    }
}
}
application.removeQueue = function(oldClient){
    var uname = gFrameworkFC.getClientGlobals(oldClient).username;
    var temp_array = this.queue_so.getProperty("fileList")
    for(var i=temp_array.length;i>0;i--){
        if(temp_array[i-1].uname == uname){
            temp_array.splice(i-1,1);
        }
    }
    this.queue_so.setProperty("fileList",temp_array);
    // if this user is set as presenter clear the so
    var temp_presenter = this.queue_so.getProperty("presenter")
    if(temp_presenter.name == uname){
        this.removePresenter();
    }
}

// removes the Presenter from SO
application.removePresenter = function(){
    var temp_obj = new Object();
    temp_obj.name = "empty";
    temp_obj.url = "empty";
    this.queue_so.setProperty("presenter",temp_obj);
}

//=====//
//////////////////////End Script//////////////////////
//=====//

```

User authentication script

```

<%
'=====//
'Desc: User Authentication script
'     handles user validation against access database
'Filename: authenticate.asp
'Author: Abey George
'Contact: abeymg@tamu.edu
'=====*/
'=====//
'////////////////////Start Script//////////////////////
'/=====//

Option Explicit
' -----
' Connect to the database

```

```

' This could be placed in an include as its used in
' multiple files, but we keep it here for simplicity
' -----
Dim objConn, strConn
' Create the database connection
Set objConn = Server.CreateObject("ADODB.Connection")
' Create the connection string
strConn = "DSN=coads"
objConn.ConnectionString = strConn

' Open the connection
objConn.Open()

Dim objRS, strSQL
' Create the getUserID record set to use
Set objRS = Server.CreateObject("ADODB.Recordset")
strSQL = "SELECT * FROM users WHERE Email =
'"&Request("email")&"'"

objRS.Open strSQL, objConn, 3, 3
Dim strResult

If Not objRS.EOF Then
  If objRS("Pwd") = Request("password") Then
    strResult = "&authenticate=true"
    strResult = strResult & "&uName="&objRS("UName")
    strResult = strResult & "&uType="&objRS("UType")
    strResult = strResult & "&fName="&objRS("FName")
    strResult = strResult & "&lName="&objRS("LName") & "&"
    'store login info in access log
    Dim objRS2, strSQL2
    Set objRS2 = Server.CreateObject("ADODB.Recordset")
    strSQL2 = "SELECT * FROM access_log"
    objRS2.Open strSQL2, objConn, 3, 3
    objRS2.addNew
    objRS2("userID") = objRS("UName")
    objRS2("userType") =objRS("UType")
    objRS2("IP") = Request.ServerVariables("REMOTE_ADDR")
    objRS2("Date") = now
    objRS2.update
    objRS2.Close
    Set objRS2 = Nothing

  Else
    strResult = "&authenticate=false&"
  End If
Else
  strResult = "&authenticate=false&"
End If

Response.Write(strResult)

objRS.Close

```

```

Set objRS = Nothing
objConn.Close
Set objConn = Nothing
'=====//
'//////////End Script//////////
' /=====//
%>

```

Subscriber file upload script

```

<!--#include file="Loader.asp"-->
<%
'=====//
' Desc: Subscriber upload files script
'       handles subscriber user file uploads
' Filename: uploadStudent.asp
' Author: Abey George
' Contact: abeymg@tamu.edu
'=====*/
'=====//
'//////////Start Script//////////
' /=====//

        Server.scripttimeout = 5400
        ' if the upload page is accessed directly without logging in
send back to login page with an error meage
        Response.Buffer = True
        ' load object
        Dim load
            Set load = new Loader
            ' calling initialize method
            load.initialize
        ' File binary data
        Dim fileData
            fileData = load.getFileData("file")
        ' File name
        Dim fileName
            fileName = LCase(load.GetFileName("file"))
        ' File path complete
        Dim filePathComplete
            filePathComplete = load.getFilePathComplete("file")
        ' File size
        Dim fileSize
            fileSize = load.getFileSize("file")
        ' File size translated
        Dim fileSizeTranslated
            fileSizeTranslated =
load.getFileSizeTranslated("file")
        ' Content Type
        Dim contentType
            contentType = load.getContentType("file")

```

```

        ' Check content type and assign respective path where file
will be uploaded
        Dim pathToFile
        Dim validFile
        'If contentType = "application/x-shockwave-flash" Then
        ' pathToFile = Server.MapPath("upLoaded/"
&Session("path")& "/_images") & "/" & fileName
        'validFile = True
        'Else
        If contentType = "image/pjpeg" Then
        pathToFile = Server.MapPath("upLoaded/" &Session("path")&
"/_images") & "/" & fileName
        validFile = True
        Else
        validFile = False
        End If

        If validFile = True Then
        ' check to see if username folder exists otherwise create
it
        Dim myFSO
        Set myFSO =
Server.CreateObject("Scripting.FileSystemObject")
        If Not myFSO.FolderExists(Server.MapPath("upLoaded/"
&Session("path"))) Then
        myFSO.CreateFolder(Server.MapPath("upLoaded/"
&Session("path")))
        End If

        ' check to see if _images folder exists otherwise create
it
        If Not myFSO.FolderExists(Server.MapPath("upLoaded/"
&Session("path")& "/_images")) Then
        myFSO.CreateFolder(Server.MapPath("upLoaded/"
&Session("path")& "/_images"))
        End If

        Set myFSO = Nothing

        'upload files
        Dim fileUploaded
        fileUploaded = load.saveToFile ("file", pathToFile)
        fileUploaded = true
        End If

        ' destroying load object
        Set load = Nothing
    %>

<html>
<head>
    <title>Upload Files</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1"></head>

```



```

    </tr>
    <tr bordercolor="#999999">
      <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2">Source
        File Path :</font></td>
      <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2"><%= filePathComplete %></font></td>
    </tr>
    <tr bordercolor="#999999">
      <td bordercolor="#FFFFFF" bgcolor="#FFFFFF"><font
face="Verdana, Arial, Helvetica, sans-serif" size="2">Source
        File Size :</font></td>
      <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2"><%= fileSize %></font></td>
    </tr>
    <tr bordercolor="#999999">
      <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2">Source
        File Size(Kb) :</font></td>
      <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2"><%= fileSizeTranslated %></font></td>
    </tr>
    <tr bordercolor="#999999">
      <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2">Content
        Type :</font></td>
      <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2"><%= contentType %></font></td>
    </tr>
    <tr bordercolor="#999999">
      <td colspan="2" bordercolor="#FFFFFF">&nbsp;</td>
    </tr>
  </table>
  <table width="400" align="center">
    <tr>&nbsp;</tr>
    <tr>
      <td><hr size="1" noshade><div align="center">
        <font color="#666666" size="1" face="Verdana, Arial,
Helvetica, sans-serif">COADS : Collaborative Online Architectural
Design Studio </font> </td>
    </tr>
  </table>
</body>
</html>

<%
'=====//
'//////////End Script//////////
' /=====//
%>

```

Moderator file upload script

```

<!--#include file="Loader.asp"-->
<%
'=====//
'Desc: Moderator upload files script
'   handles moderator user file uploads
'Filename: uploadFaculty_proc.asp
'Author: Abey George
'Contact: abeymg@tamu.edu
'=====*/
'=====//
'//////////Start Script//////////
'/=====//
    Server.scripttimeout = 5400
    ' if the upload page is accessed directly without logging in
send back to login page with an error meage
    Response.Buffer = True
    ' load object
    Dim load
        Set load = new Loader
        ' calling initialize method
        load.initialize
    ' File binary data
    Dim fileData
        fileData = load.getFileData("file")
    ' File name
    Dim fileName
        fileName = LCase(load.GetFileName("file"))
    ' File path complete
    Dim filePathComplete
        filePathComplete = load.getFilePathComplete("file")
    ' File size
    Dim fileSize
        fileSize = load.getFileSize("file")
    ' File size translated
    Dim fileSizeTranslated
        fileSizeTranslated =
load.getFileSizeTranslated("file")
    ' Content Type
    Dim contentType
        contentType = load.getContentType("file")

    ' Check content type and assign respective path where file
will be uploaded
    Dim pathToFile
    Dim validFile
    If contentType = "application/x-shockwave-flash" Then
        pathToFile = Server.MapPath("upLoaded/_presentations") &
"/" & fileName
        validFile = True
    'ElseIf contentType = "image/pjpeg" Then

```

```

        ' pathToFile = Server.mapPath("upLoaded/"
&Session("path")& "/_images") & "/" & fileName
        ' validFile = True
    Else
        validFile = False
    End If

    If validFile = True Then
        ' check to see if username folder exists otherwise create
it
        Dim myFSO
        Set myFSO =
Server.CreateObject("Scripting.FileSystemObject")
        'If Not myFSO.FolderExists(Server.mapPath("upLoaded/"
&Session("path"))) Then
        ' myFSO.CreateFolder(Server.mapPath("upLoaded/"
&Session("path")))
        'End If

        ' check to see if _images folder exists otherwise create
it
        'If Not myFSO.FolderExists(Server.mapPath("upLoaded/"
&Session("path")& "/_images")) Then
        ' myFSO.CreateFolder(Server.mapPath("upLoaded/"
&Session("path")& "/_images"))
        'End If

        ' check to see if _presentations folder exists otherwise
create it
        If Not
myFSO.FolderExists(Server.mapPath("upLoaded/_presentations")) Then
myFSO.CreateFolder(Server.mapPath("upLoaded/_presentations"))
        End If

        Set myFSO = Nothing

        'upload files
        Dim fileUploaded
        fileUploaded = load.saveToFile ("file", pathToFile)
        fileUploaded = true
    End If

    ' destroying load object
    Set load = Nothing
%>

<html>
<head>
    <title>Upload Files</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1"></head>
    <body bgcolor="#FFFFFF">

```



```

        <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2">Source
        File Path :</font></td>
        <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2"><%= filePathComplete %></font></td>
    </tr>
    <tr bordercolor="#999999">
        <td bordercolor="#FFFFFF" bgcolor="#FFFFFF"><font
face="Verdana, Arial, Helvetica, sans-serif" size="2">Source
        File Size :</font></td>
        <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2"><%= fileSize %></font></td>
    </tr>
    <tr bordercolor="#999999">
        <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2">Source
        File Size(Kb) :</font></td>
        <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2"><%= fileSizeTranslated %></font></td>
    </tr>
    <tr bordercolor="#999999">
        <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2">Content
        Type :</font></td>
        <td bordercolor="#FFFFFF"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2"><%= contentType %></font></td>
    </tr>
    <tr bordercolor="#999999">
        <td colspan="2" bordercolor="#FFFFFF">&nbsp;</td>
    </tr>
</table>
<table width="400" align="center">
    <tr>&nbsp;</tr>
    <tr>
        <td><hr size="1" noshade><div align="center">
            <font color="#666666" size="1" face="Verdana, Arial,
Helvetica, sans-serif">COADS : Collaborative Online Architectural
Design Studio </font> </td>
        </tr>
</table>
</body>
</html>
<%
'=====//
'//////////End Script//////////
'/////
%>

```

Retrieve file list script

```

<%
'=====//
'Desc: retrieve files script
'retrieves list of uploaded files
'Filename: getFiles.asp
'Author: Abey George
'Contact: abeymg@tamu.edu
'=====*/
'=====//
'//////////Start Script//////////
'/=====//
    ' File System Object
    Dim fso
    Set fso = Server.CreateObject("Scripting.FileSystemObject")
    ' "Uploaded/userid" Folder
    Dim rootFolder
    If Request("type") = "_presentations" Then
        Set rootFolder =
fso.GetFolder(Server.MapPath("upLoaded/"&Request("type")))
    Else
        Set rootFolder =
fso.GetFolder(Server.MapPath("upLoaded/"&Request("uname")&"/"&Request("
type")))
    End IF
    Dim rootFiles
    Set rootFiles = rootFolder.Files
    Response.ContentType="text/xml"
    'Response.Write("<?xml version='1.0' encoding='iso-8859-
1'?">")
    If rootFolder.Size > 0 Then
        Response.Write("<Files>")
        For Each file in rootFiles
            Response.Write("<File Url='"&
"upLoaded/"&Request("uname")&"/"&Request("type")&"/"& file.Name
&"'>"&file.Name&"</File>")
        Next
        Response.Write("</Files>")
    Else
        Response.Write "<ul><li type=""circle"">No Files
Uploaded.</ul>"
    End If
    Set files = nothing
    Set rootFolder = nothing
    Set rootFiles = nothing
    Set fso = nothing

'=====//
'//////////End Script//////////
'/=====//

%>

```

VITA

GEORGE, ABEY M.

1974 Born in Kerala, India

ACADEMIC BACKGROUND

1991-1997 B. Arch.,
University of Kerala, Trivandrum, India.

2001-2004 M. S. in Architecture
Texas A&M University, College Station, TX, USA.

PROFESSIONAL EXPERIENCE

1997-1999 Architect
Stein Mani Chowfla Architects, New Delhi, India.

1999-2001 Architect Designer
Dar Al Riyadh Consultants, Dhahran, K.S.A.

Mailing Address:
4302 College Main, #201
Bryan, TX – 77801

Permanent Address:
67-C, Pocket 3,
Mayur Vihar 1,
New Delhi – 110091,
India.