

Web Based Course Management and
Delivery System (COMMAND)

by

Abel Sanchez

B.S., Civil Engineering (1995)

University of Arizona

Submitted to the Department of Civil and Environmental Engineering
in Partial Fulfillment of the Requirements for the Degree of

Master of Science

at the

Massachusetts Institute of Technology

June 1998

© 1998 Massachusetts Institute of Technology
All rights reserved

Signature of Author
Department of Civil and Environmental Engineering
May 8, 1998

Certified by
✓
John R. Williams
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by
Joseph M. Sussman
Chairman, Department Committee on Graduate Students

JUN 02 1998

LIBRARY

ENG

Web Based Course Management and
Delivery System (COMMAND)

by

Abel Sanchez

Submitted to the Department of Civil and Environmental Engineering
on May 8, 1998 in Partial Fulfillment of the
Requirements for the Degree of Master of Science

ABSTRACT

Among the most important issues facing educators today are the instructional integration of technology and user support. These issues, coupled with the trend toward using web sites as a point of comparison for courses as well as schools, provide strong motivations for investigating how the web can be used to support and augment classroom instruction.

This thesis, building on work started three years ago, will discuss the requirements, design, development and implementation, of web sites, as a course delivery system scales up and plans for institutionalization

Three years ago, a framework to address the instructional integration of technology was developed and tested using three prototype web applications deployed in the schools of Engineering, Business, and Architecture. These courses involved faculty, teaching staff, and students, as well as professionals in industry. The prototypes sought to identify the following: (a) educational opportunities afforded by the web based on various teaching styles; (b) tasks that lent themselves well to its asynchronous, albeit ubiquitous, nature; (c) issues involved in web site preparation and management, including web authorship and student use.

Emphasis was placed then, as in this work, on developing interactive features and designing general templates or containers that could accommodate various subjects and empower both professors and students, while shielding them from complexity, to create and maintain web content without having to rely on an intermediary.

Thesis Supervisor: John R. Williams

Title: Professor of Civil and Environmental Engineering

Acknowledgments

Professor **John R. Williams** for his irrepressible enthusiasm, knowledge, and guidance. For always taking time out of his busy schedule to discuss my projects. For always making sure I had the hardware and software I needed. It has been a pleasure having him as my advisor.

My **wife**, my better half (media naranja), for her undevoted support through out the years. For being my best friend and always being in my cheering section. Thank you for knowledge, understanding, and love.

My **father**, who always believed that education is the most important gift a person can receive and who instilled in me the value of continuous learning. To my **parents**, I owe what I am today and what I can achieve in the future to them.

My friends, for making MIT fun. The IESL bunch for being more than lab mates, especially **Gilbert, Emma, and Kathy**. To the structures group, for being good friends through out the last two years, especially **David, Larry, and Ramon** (even though you are ME). To **Hai** and **JP**, for their support in IESL projects.

Clarissa Hidalgo for her efforts above and beyond the call of duty working on COMMAND.

Kimberly Ryan for her contribution to COMMAND and software from Lotus.

Sameer Reheja, for sharing MIT culture and passing on all those cool system tricks.

To **all MIT students and faculty using the COMMAND system**. Your feedback has been very valuable.

Joan MacCusker, for her help and advice on administrative issues.

Lotus Development Corporation for donating Lotus Notes/Domino 4.6, Bean Machine for Java, and VideoNotes 1.2.

Thanks to the **Systems Design and Management** program and **Ford** for providing funding for my research.

ACKNOWLEDGMENTS.....	3
CHAPTER 1	12
INTRODUCTION.....	12
1.1 GOALS	13
1.2 RESEARCH METHODOLOGY	14
1.3 PROJECT TIMELINE	15
1.4 CHAPTER GUIDE	15
CHAPTER 2	17
AUDIENCE, WEB MODEL, AND DEVELOPMENT TEAM.....	17
2.1 AUDIENCE.....	17
2.2 WEB MODEL.....	19
2.3 TEAM BUILDING	20
2.4 DEVELOPMENT GROUP	21
CHAPTER 3	23
INSTITUTION NEEDS	23
3.1 INITIAL MEETINGS	23
3.2 CONCERNS	25
3.3 CONCERNS ADDRESSED.....	25
CHAPTER 4	26
FUNDAMENTAL COMPONENTS OF A GOOD SYSTEM.....	26
4.1 SCALABILITY	26
4.1.1 <i>Software</i>	26
4.1.2 <i>Hardware</i>	27
4.1.3 <i>Personnel</i>	27
4.1.4 <i>Bandwidth</i>	27
4.2 ROBUSTNESS.....	28

4.2.1	<i>Redundancy System</i>	28
4.3	NEW FEATURES	31
4.3.1	<i>Improve Recitation Sections Automatic Assignments and Conflict resolution</i>	31
4.3.2	<i>Full Support For Grade Tabulation</i>	31
4.3.3	<i>Lectures Online</i>	31
4.3.4	<i>Live Video Encoding - Video Streaming</i>	32
4.3.5	<i>Hand Raising Software for Remote Questions</i>	32
4.3.6	<i>Online Teaching Assistant</i>	33
4.4	SECURITY	33
4.4.1	<i>Internet Security Threats</i>	34
4.4.2	<i>Secure Socket Layer (SSL)</i>	35
4.4.3	<i>SSL Handshake</i>	36
4.4.4	<i>Kerberos</i>	38
4.4.5	<i>Final Words</i>	41
CHAPTER 5	42
DESIGNING FOR COMMUNICATION: LAYOUT, STRUCTURE, AND NAVIGATION.	42
5.1	STRUCTURAL GUIDELINES	42
5.2	EXPLICATE STRUCTURE	43
5.3	EMPHASIZE CENTRAL STRUCTURE	44
5.4	GRAPHICS DESIGN	44
5.5	TEXT DESIGN	46
5.6	THE GRAPHICAL USER INTERFACE	48
5.6.1	<i>The Principle of User Correctness</i>	48
5.7	THE PRINCIPLE OF LEAST ASTONISHMENT	49
5.7.1	<i>Navigational Bars</i>	49
5.7.2	<i>Clarity</i>	50
5.7.3	<i>Completeness</i>	50
5.7.4	<i>Consistency</i>	51

5.7.5	<i>Robustness</i>	51
5.7.6	<i>User-Centeredness</i>	52
CHAPTER 6		53
CENTRAL ADMINISTRATION		53
6.1	AUTOMATED SITE GENERATION	53
6.1.1	<i>Site Generation</i>	53
6.2	NO TECHNICAL KNOWLEDGE REQUIRED	55
6.3	CENTRAL MANAGEMENT MODEL	55
6.3.1	<i>COMMAND's Home Page</i>	55
6.3.2	<i>Registration</i>	56
6.3.3	<i>Student Folder</i>	57
6.3.4	<i>Help</i>	58
6.3.5	<i>Central Structure</i>	59
6.3.6	<i>Access Model</i>	60
6.4	PLATFORM INDEPENDENCE	61
6.4.1	<i>Testing Platform</i>	61
6.5	SECURITY	62
6.6	VIDEO STREAMING	62
CHAPTER 7		64
THE COURSE DATABASE		64
7.1	ORGANIZATION	64
7.1.1	<i>Structure</i>	64
7.1.2	<i>Course Home Page</i>	65
7.2	COURSE DOCUMENTS	66
7.2.1	<i>About Course</i>	67
7.3	HOMEWORK ADMINISTRATION	69
7.3.1	<i>Interface</i>	69

7.3.2	<i>Late submissions and submitting a new version of the homework</i>	70
7.4	COLLABORATION	71
7.4.1	<i>Discussion</i>	71
7.4.2	<i>Chat</i>	72
7.4.3	<i>Contact Information</i>	73
7.5	ADMINISTRATION TOOLS	73
7.5.1	<i>Groups</i>	73
7.5.2	<i>Presentation Scheduler</i>	75
7.5.3	<i>Calendar</i>	76
7.5.4	<i>Sections</i>	76
7.5.5	<i>Announcements</i>	77
7.6	WORKFLOW	78
7.7	VIDEO STREAMING	79
CHAPTER 8	81
CLOSING REMARKS AND FUTURE WORK	81
8.1	CLOSING REMARKS	81
8.2	APPLICATION OF INFORMATION MODEL	81
8.3	ADDITIONAL FEATURES	82
8.3.1	<i>Compiling online</i>	82
8.4	IMPROVING THE REMOTE STUDENT EXPERIENCE	83
APPENDIX A	84
AVAILABLE INFORMATION TECHNOLOGY	84
A.1	WEB SERVERS	84
A.2	DATABASES	85
A.3	DATABASE – WEB SERVER	86
A.3.1	<i>Web Server, Database Server, with Plugins and Native ODBC</i>	86
A.3.2	<i>ActiveX and Java</i>	87

<i>HTML only</i>	87
<i>A.3.4 Overview</i>	88
A.4 WEB BROWSERS.....	88
<i>A.4.1 HTML 3.2</i>	88
A.5 REAL TIME GROUPWARE TOOLS	89
A.6 VIDEO STREAMING.....	90
A.7 SECURITY	90
A.8 LOTUS NOTES.....	91
<i>A.8.1 Typical Database File</i>	93
<i>A.8.2 Domino Architecture</i>	95
<i>A.8.3 Notes Security</i>	95
APPENDIX B	100
B.1 SPECIFICATIONS FOR COMMAND PHASE I.....	100
APPENDIX C	105
C.1 SYSTEMS ARCHITECTURE – 16.982/6.971	105
C.2 SYSTEM AND PROJECT MANAGEMENT – 15.965.....	106
C.3 COMPUTER AIDED ENGINEERING – 1.124	107
APPENDIX D	108
D.1 LIST OF COURSES ON COMMAND	108
BIBLIOGRAPHY	110

<i>Figure 1.2.1-1 Research Methodology</i>	14
<i>Figure 2.1.1-1 Models of Teaching and Web Use</i>	18
<i>Figure 4.2.1-1 Redundancy Model</i>	30
<i>Figure 4.4.2-1 Public Key Encryption</i>	35
<i>Figure 4.4.3-1 Secure Socket Layer</i>	37
<i>Figure 4.4.3-2 SSL Handshake</i>	38
<i>Figure 4.4.4-1 Request to Ticket Granting Service</i>	39
<i>Figure 4.4.4-2 Ticket Creation</i>	39
<i>Figure 4.4.4-3 Reply to Client</i>	40
<i>Figure 4.4.4-4 Kerberos Reply</i>	40
<i>Figure 5.7.1-1 Navigational Structures</i>	49
<i>Figure 5.7.1-2 Navigational Aides</i>	50
<i>Figure 6.1.1-1 Course Request Interface</i>	54
<i>Figure 6.3.1-1 COMMAND's Home Page</i>	56
<i>Figure 6.3.2-1 Registration</i>	56
<i>Figure 6.3.3-1 Student Folder</i>	57
<i>Figure 6.3.3-2 Account Management</i>	58
<i>Figure 6.3.4-1 Video Tutorials</i>	59
<i>Figure 6.3.5-1 COMMAND Central Structure</i>	60
<i>Figure 6.3.6-1 Access Model</i>	60
<i>Figure 6.4.1-1 Platform Independence</i>	61
<i>Figure 7.1.1-1 Structure</i>	64
<i>Figure 7.1.2-1 Course Home Page</i>	65
<i>Figure 7.1.2-2 Course Home Page - Administrative</i>	65
<i>Figure 7.1.2-3 Urgent Announcements</i>	66
<i>Figure 7.1.2-4 Course Staff</i>	66
<i>Figure 7.2.1-1 About Course Form</i>	67
<i>Figure 7.2.1-2 About Course Documents</i>	68

<i>Figure 7.2.1-3 Readings</i>	68
<i>Figure 7.2.1-4 Lecture Notes, Assignments, and Solutions</i>	69
<i>Figure 7.3.1-1 Homework Form</i>	69
<i>Figure 7.3.2-1 Homework Evaluation</i>	70
<i>Figure 7.3.2-2 Homework Documents</i>	71
<i>Figure 7.4.1-1 Discussion Entry</i>	71
<i>Figure 7.4.1-2 Class Discussion Documents</i>	72
<i>Figure 7.4.2-1 Chat</i>	73
<i>Figure 7.4.3-1 Class Directory</i>	73
<i>Figure 7.5.1-1 Groups Form</i>	74
<i>Figure 7.5.1-2 Groups</i>	74
<i>Figure 7.5.2-1 Presentation Form</i>	75
<i>Figure 7.5.2-2 Presentations</i>	75
<i>Figure 7.5.4-1 Section Form</i>	77
<i>Figure 7.5.4-2 Sections</i>	77
<i>Figure 7.5.5-1 Announcements</i>	78
<i>Figure 7.6.1-1 Workflow Showing Databases, Templates, and Documents that Support</i>	78
<i>Figure 7.6.1-2 Workflow Showing Databases, Templates, and Documents for Managing Assignments</i>	79

CHAPTER 1 Introduction[†]

Among the most important issues facing educators today are the instructional integration of technology and user support. These issues, coupled with the trend toward using web sites as a point of comparison for courses as well as schools, provide strong motivations for investigating how the web can be used to support and augment classroom instruction. With this backdrop in mind, the goals of the thesis will be described and the research methodology articulated.

Three years ago, a framework to address the instructional integration of technology was developed and tested using three prototype web applications deployed in the schools of Engineering, Business, and Architecture. These courses involved faculty, teaching staff, and students, as well as professionals in industry. The prototypes sought to identify the following: (a) educational opportunities afforded by the web based on various teaching styles; (b) tasks that lent themselves well to its asynchronous, albeit ubiquitous, nature; (c) issues involved in web site preparation and management, including web authorship and student use.

Emphasis was placed then, as in this work, on developing interactive features and designing general templates or containers that could accommodate various subjects and empower both professors and students, while shielding them from complexity, to create and maintain web content without having to rely on an intermediary.

This thesis, building on the work started three years ago, will discuss the requirements, design, development and implementation, of web sites, as a course delivery system scales up and plans for institutionalization.

[†] The first two sections of this chapter and the abstract have been adapted from Hidalgo [6].

1.1 Goals

The goals of this thesis are:

1. **Develop 'generic' web-based models/templates that can be easily customized and extended to meet the needs of various classes and students at a distance.**

Design general templates and continuously refine these to create a base set of containers that are scaleable and customizable.

2. **Facilitate dissemination and institutionalization of web-based teaching tools and ensure quality control in course module development.**

Use the standard set of templates to initiate a “mass customization” effort targeted at enabling teachers to set up home pages quickly and effectively, while shielding them from complexity, and institutionalizing this practice.

3. **Empower users to create their own content.**

Users in this respect refers to professors as well as their teaching assistants, and students. One goal is to develop a model that doesn't rely on the webmaster to post materials because: (a) many professors can't rely on technical support at the institutional or even departmental level; and/or (b) they simply want control over what's published and how it appears over the web because they feel that they're the domain experts and know best how to present the material. In addition, empowering the students to contribute directly to the class web site lessens the overhead involved in administering the class and gives them more room for creativity.

1.2 Research Methodology

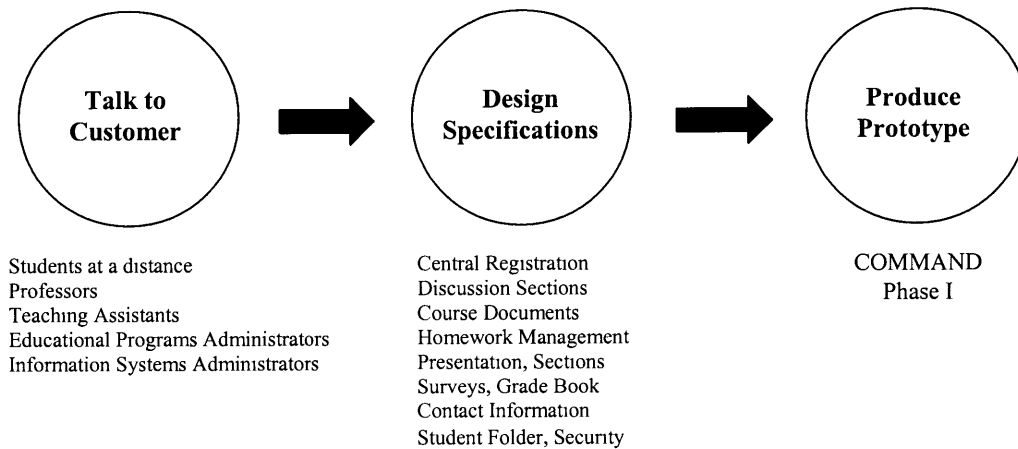


Figure 1.2.1-1 Research Methodology

The approach taken in conducting this research was threefold: talk to customer, design specifications, produce the prototype (see figure above).

Initially this work sought academic programs with remote students located at a distance. The Systems Design and Management¹ (SDM) program was targeted due for its high percentage (75% [7]) of remote students.

During the fall of 1997, scaled down prototypes, of the proposed system, were created to serve as proof-of-concept. Three courses ran, two from SDM, and one in the School of Engineering (see appendix c). During the life of the prototypes, the full scale course management and delivery system (COMMAND) was developed. Feedback from participating faculty, students, and the institutions information systems managers were used as design guidelines. The feedback from information systems administrators was considered because it was important to create a system that could grow into the institutions information and security model.

¹ A two year distance education program designed for company-sponsored practicing engineers

The first phase of COMMAND's development cycle was released in January 1, 1998. During the course of the 1998 spring term, feedback from students and teaching staff was used to further refine the system, set priorities, and optimize the graphical user interface. Feedback came in the form of e-mail messages, face to face meetings, and informal conversations.

1.3 Project Timeline

	Design Period	Release Date
Prototypes	Summer 1997	August 1997
COMMAND 1 st Phase	Fall 1997	January 1998
COMMAND 2 nd Phase	Spring – Summer 1998	August 1998
COMMAND 3 rd Phase	Fall 1998	January 1999

1.4 Chapter Guide

The evolution, difficulties, and discoveries that evolved from the design and development of COMMAND are laid out over 8 chapters:

- Chapter 2 defines the audience, learning goals, web models, and development team members.
- Chapter 3 addresses the concerns and needs of an institution when considering a course delivery system.
- Chapter 4 focuses on the key components of an effective system. The areas considered are scalability, robustness, system features, and security.
- *Considering structure and organization as content*, chapter 5, presents guidelines for the design of site structure. It also goes on to discuss graphic and text design in the context of graphical user interfaces.

- Chapter 6 presents the implementation of the proposed web model. It discusses the centrally managed model needed to scale up and automate the course web sites. It also discusses registration, site generation, access model, platform independence, and high level features.
- Chapter 7 goes into the specifics of the course web sites. It discusses the organization, course documents, homework administration, collaboration, and administrative tools.
- Chapter 8. Future work on COMMAND can be in additional features to the system, applications of the current information model, or improving the remote student experience. It also makes some closing remarks.
- Appendix A, This appendix introduces available information technology and possible configurations. It goes on to discuss several models to access database engines through the web. It also presents current tools for course delivery.
- Appendix B, specifications for COMMAND phase I.
- Appendix C, screen shots of prototypes.
- Appendix D, list of course on COMMAND for the 1998 spring term.

CHAPTER 2

Audience, Web Model, and Development Team

This chapter defines the audience, learning goals, web models, and development team members.

2.1 Audience

The target audience is students and faculty, both on-campus and remote. Consequently, users will connect from commercial internet service providers (ISP) and local area networks (LAN). The bandwidth for this connections will be 14.4 kbps – 56 kbps for ISPs, 128 kbps for ISDN², and 1 mbps for LANs.

As the course delivery system scales up the audience diversity is expected to increase. The different users will come from the schools of Engineering, Science, Humanities, Architecture, and Management [2]. User needs, both students and faculty, will vary across disciplines due to academic culture, teaching models, computer facilities, and the degree of computer literacy.

Hidalgo [6] studied classes across several schools, “These classes, conducted in different schools within MIT, subscribed to different teaching models, thereby providing a representative cross-section of the academic population from which to base the functional requirements and system design”. Her findings are shown in the following diagram.

² integrated services digital network

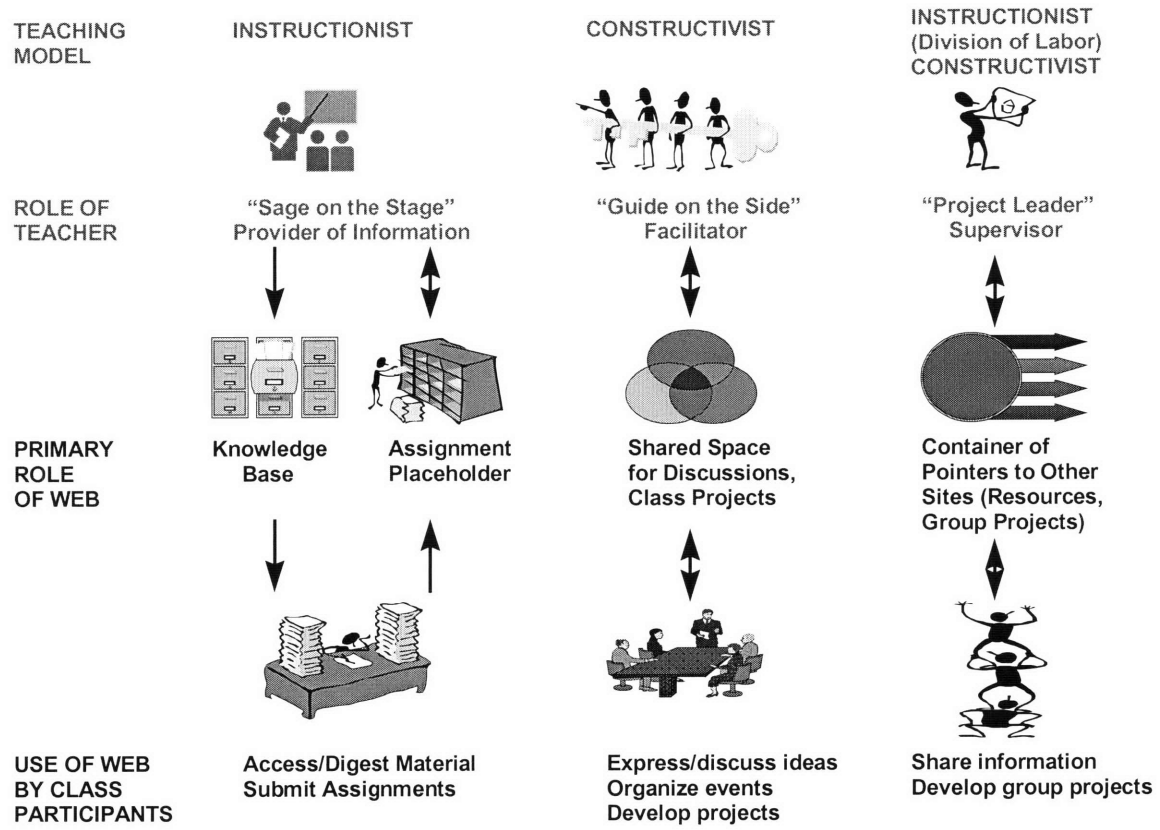


Figure 2.1.1-1 Models of Teaching and Web Use

2.2 Web Model

Currently most courses follow a centralized or decentralized web site management model [6].

1. Centralized Model

- Hardware, software and human (application development and technical support) resources maintained at a central site. Content provider needs to go through this intermediary for application design changes and possibly publication of content.
- Advantages: Economies of scale resulting from sharing of resources, control of standards, availability of critical mass of skills.
- Disadvantages: Author does not have full control over application design and publication of content, larger start-up costs (for central site), slower response/turnaround time for user requests.

2. Decentralized Model

- Content provider has full control over the design and content of the web site. In a fully decentralized model, the 'course developers' (i.e. professor, teaching staff) maintain the hardware and server software needed to keep the web site up and running. A deviation of this model has the hardware and server software maintained by a central resource, such as Athena, but leaves development of the web application and html pages to the course developers. The latter model may constrain the application design capabilities. For instance, relying on the Athena web server prevents course developers from using cgi-scripts in their web sites.
- Advantages: Author has full control over publication and can see results immediately, no need to rely on an intermediary
- Disadvantages: Lack of hardware/software support and technical expertise, larger over-all costs to the university, reinvention of wheels, variable standards, no synergy and integration.

3 Proposed Model

This thesis will base its course template on Hidalgo's model. The templates have the following characteristics:

- Support a common set of course delivery and management tasks;
- Be easily customized and extended to meet to the unique needs of the various classes;
- Empower both the teaching staff and students to publish and edit content without the intervention of a webmaster;
- Run on a development and production environment that supports multiple platforms, is scaleable, provides industrial-strength security and database management support.

This paradigm lends itself well to prototyping, rapid application development and mass customization, and can be adapted to both the centralized and decentralized models. The use of generic templates that can be easily replicated and customized enables, but does not require, development of standards, promotion of best practices, flexibility in both application design and publication of content. Given the short lead times and limited technical resources available to most professors in developing web content, the generic toolkit described above enables them to get a jumpstart on their web site and focus on development of course material.

2.3 Team Building

Before determining what types of team members are needed one needs to know the project requirements.

Technical Needs:

- Lotus Notes Domino Database Development and administration
- Coding in HTML, JavaScript, CGI, and Java
- Client Server Architecture
- Network Services for TCP/IP
- Windows NT Server, Multiprocessor, RAID
- Video Capture and Streaming
- Network Topology for Large Scale Systems

- Security for NT servers, Kerberos, Firewalls, SSL
- Windows 95&NT, UNIX, Mac
- Web Graphics Artist

2.4 Development Group

The course delivery system, technical needs, were met by the following development team:

Lotus Internet Engineer

- Free Software
- Technical Advice on Lotus Products
- Training on Lotus Products
- Experienced Developer
- Professional Business Skills

Web Developer, Intelligent Engineering Systems Lab. (IESL) Research Assistant

- Lotus Notes Domino Database Development and administration
- Coding in HTML, JavaScript, CGI, and Java
- Windows 95&NT, UNIX, Mac, Linux
- Client Server Architecture
- Network Services for TCP/IP
- System Administration experience with Windows NT Server, Multiprocessor, RAID
- Video Capture and Streaming
- Network Topology for Large Scale Systems
- Security for NT servers, Kerberos, Firewalls, SSL

Consultant, MIT alumni

- Experienced Information Technology Consultant

- Experienced Lotus Notes Developer
- Coding in HTML, JavaScript, CGI, and Java
- Experience Introducing Web Applications in an Educational Setting

Graphic Designer, IESL Research Assistant

- Adobe PhotoShop, Premiere
- CAD, 3D Studio Max
- Web Graphics
- Video Capture, Editing
- Multimedia on the web

Security Specialist, IESL Research Assistant

- Firewalls
- Secure web connections, SSL
- UNIX, MS-Windows, Linux, Mac
- Network Security
- Cryptography

CHAPTER 3

Institution Needs

This chapter addresses the concerns and needs of an institution when considering deployment of a course delivery system.

3.1 Initial Meetings

The web course prototypes offered a limited version of COMMAND as shown below:

Prototypes	COMMAND 1 st Phase
Basic Presentation Scheduler	Full Presentation Scheduler
Registration needed for every course	Central Registration
Discussion Sections (only public)	Discussion Sections (public and private)
Course Documents	Course Documents
Homework Upload	Homework Upload & Grading
	Administrative Area
	Course Listing
	Contact Information and Course Selection
	Hot Topics
	Student Folder
	System Log

After students, faculty, and staff had used the prototypes and were informed of the planned features for COMMAND, they had the following suggestions (the check marks indicate the projected implementation date).

Prototypes	Students, Faculty, and Staff Requests	COMMAND 1 st Phase Jan 98	COMMAND 2 nd Phase August 98
√	Password Protect site (copyright material)	√	√
√	universal guest account	√	√
	group homework submission	√	√
	personal/team calendars		√
	end of term grading support		√
	grade statistics		√
	online testing & evaluation		√
	facility to download all homework submitted	√	√
√	chat facility	√	√
	personal preferences		√
	notify students before a database is purged/archived	√	√
	support for recitation sections	√	√
	support for course-specific surveys	√	√
	support Kerberos certificates		√
	student pictures in the contact information		√
	archiving databases	√	√
	logging statistics	√	√
√	hit counters	√	√

For a detailed list of the specifications of the first phase of COMMAND's development cycle, see appendix B.

3.2 Concerns

Beyond system features, faculty and staff had the following concerns:

- Robustness – data backups, redundant servers
- Scalability – capacity: users, disk space, number of courses
- How long would COMMAND exist
- Security

3.3 Concerns Addressed

One the goals of this project is the dissemination and institutionalization of the course delivery system. Thus, the plan is to make this system a permanent addition to the institution's academic support resources.

Robustness, scalability, and security are all discussed in the next chapter.

CHAPTER 4

Fundamental Components of a Good System

This chapter focuses on the key components of an effective system. The areas considered are scalability, robustness, system features, and security.

4.1 Scalability

4.1.1 Software

What does scalability mean for a course delivery system? How many users should it support?

How many courses? How long should the course creation process take?

In the system's first phase, scalability meant being able to handle as many courses as requested.

Since the system was new, the high-end estimate was 150 courses for the first release, but the expected number was 25-35.

The overlap of users was expected to be minimal since users were going to be from several different departments. A conservative estimate of 50 users per class was used, taking into account that some classes were going to be over 250 and others just over 10.

From past experience classes that were very small, less than ten students, did not use web support. The amount of effort needed to keep track of information and paper work was minimal and most professors considered the system as something new to learn and would not venture unless the benefits outweighed their efforts in learning a new system.

Assuming 1750 students (35 courses times 50 students/course), software and hardware were more than adequate. In fact, the system could grow to be three times as large and there would be no problem. Also if the system ever grew to be larger than MIT the preparations for server clustering and load balancing were in place (see appendix A).

Lotus Notes was used as a database engine and Domino as the web server. Built into this package is the availability to cluster up to 6 servers with load balancing. For specifics of this configuration see appendix A.

4.1.2 Hardware

Once the projected loads on the machines were established, the chosen machines were wintel (MS Windows – Intel) quadruple processor machines, and wintel dual processor machines as backup servers. The main servers were equipped with redundant power supplies, redundant array of inexpensive disks (RAID), capacity for 12 hard drives, Windows NT operating systems, and tape drives.

4.1.3 Personnel

One of the objectives of the system was to automate the life span of a course. For the first phase of the project all but the initial creation of the course in Lotus Notes was automated. This task typically takes three to five minutes. Thus, even if something unexpected happened and over 150 course requests were made, this was still feasible.

4.1.4 Bandwidth

The theoretical bandwidth of the subnet the server would reside in is 10 mbps internal and 1 mbps to the outside. The reality was that the bandwidth was shared by all the computers (over 200) in

the subnet. This meant that during peak hours the server response was slower than would be expected.

4.2 Robustness

4.2.1 Redundancy System

The ideal system should be available to take web requests no matter what happens. Achieving this goal requires additional servers, software, and adds a level of complexity to the existing configuration. In spite of this obstacles, the system that was designed will work in all situations even during power outages, as long as they do not exceed 1/2 hour. This was achieved through a combination of backup servers, uninterruptible power supplies (UPS), server replication, RAID, and data backups.

The server that runs the course delivery system has a backup server that will be brought on line if the main server ever fails. The backup server maintains a mirror image of the material on the main server. This is achieved by running a replication service between the machines.

The replication that is done between machines is done at the field level, which makes the process very efficient. In this process, only the initial replication of a course database copies everything to the backup server. The following replications ignore all the data that existed prior to the last replication date and transfer only the new information. This results in a transfer of data that is minimal and very fast.

Another feature of the configuration that adds to speedy replication is that the main server is only pushing information out to the backup server and no changes are being pulled back.

Having replication in place is very important as it allows for multiple servers around the world. If the project grows to include overseas students, a server could be located in each participating country. This way overseas students would not have to suffer slow response times by accessing remote databases over low bandwidths and congested world networks. By having a local server overseas students would enjoy fast response times and the information could still be updated with MIT's server as often as necessary (daily, every 12 hours, etc).

UPS systems with a life of thirty minutes have been coupled with every server to account for the possibility of a power outage. In the event of a power outage the server will continue to have power and after alerting users will begin a shutdown process, saving all data and closing all applications in a conservative manner.

As an extra precaution, backups online and on tape are done on a nightly basis. Data is backed up on magnetic tape drives on the server. The online backup, is a system called ADSTAR Distributed Storage Manager (ADSM). ADSM is an IBM software product that lets the user back up, archive, and restore data files on a computer using a VM system as a server.

Both, the main server and the backup server are using a redundant array of inexpensive disks (RAID) system. RAID spreads stored data over several disks instead of only one; this process creates faster performance when retrieving data and greater data security.

The system servers are using RAID 5. This number gives the most robust improvements in performance and security for servers by distributing both data and parity information across three hard drives.

A model of the configuration follows.

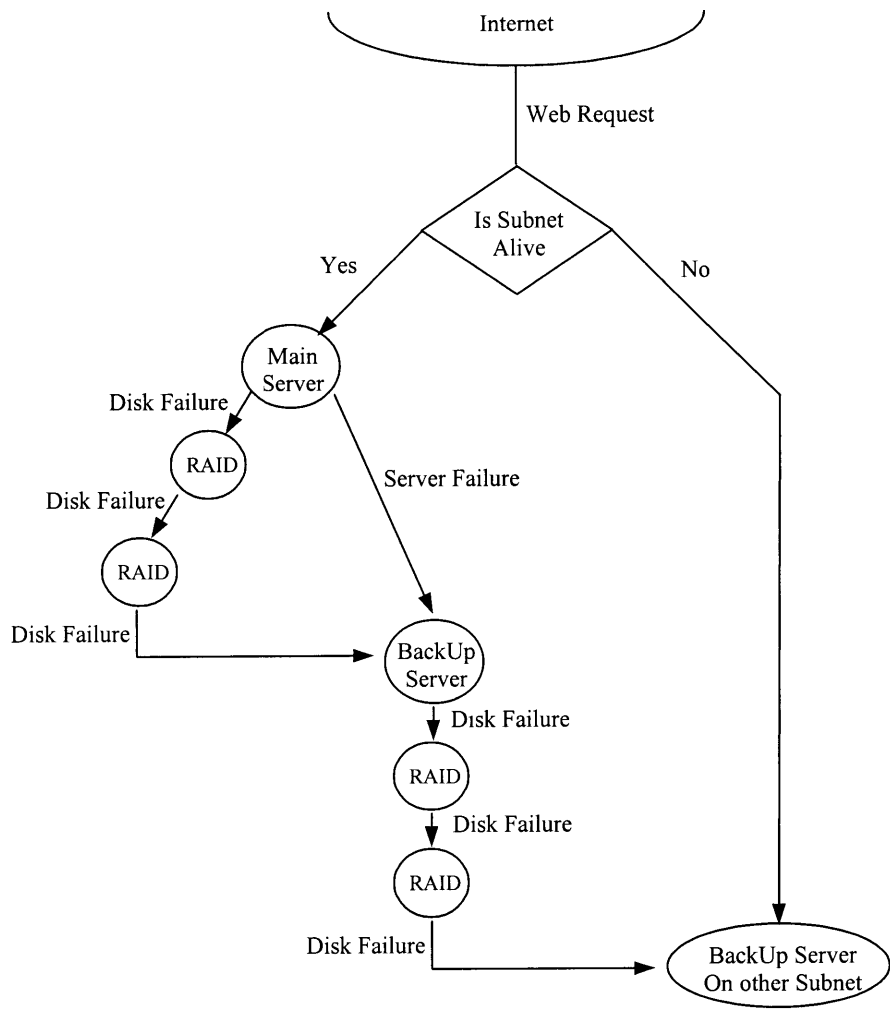


Figure 4.2.1-1 Redundancy Model

4.3 New Features

4.3.1 Improve Recitation Sections Automatic Assignments and Conflict resolution

Currently (phase 1) the assignment of recitations is automated but the logic of assignment is limited. First come first served is the criteria used for creating recitations. As recitations fill up the conflict resolution is left to the teaching assistants (TA). In the stage of development are algorithms, which will prioritize a student's chances at a requested recitation section. The algorithms will also attempt to resolve scheduling problems, students trying to change recitations once assigned, and balancing the number of students per recitation.

4.3.2 Full Support For Grade Tabulation

On line grading of homework and record of grades is a current feature of COMMAND. What is not, is a running calculation of students comprehensive averages through out the semester. In the next release (phase 2) the system will be extended to function as a grade book. All marks a student receives throughout a semester will be kept on line. The cumulative averages will also be computed and be made available to the professor. Students will also be able to view their grades as the semester progresses.

4.3.3 Lectures Online

Many lectures are currently being taped and shipped to remote students in VHS tapes. As an alternative to this method, lectures will be encoded and made available on the web. Being able to listen to a lecture for a second time will be as simple as opening a web page. The ease of access will benefit both remote and on-campus students.

Having a system which has all lecture notes, assignments, discussions, and lectures on video will create a complete package. All these resources can be used to complement a class or deliver a complete course to remote students.

4.3.4 Live Video Encoding - Video Streaming

The next step up from taped lectures is to encode live lectures. Currently students of large corporations receive live broadcasts through video conferencing systems typically on integrated services digital network (ISDN) lines. The cumulative costs of video conferencing equipment plus ISDN lines make it difficult or impossible for most students. In contrast, live video streaming over the web requires simply a computer and a connection to an internet service provider (ISP).

This feature could potentially open the way for a new market of students. Students who are qualified to attend academic programs but can not leave their current employment and their companies can not afford to, or will not, pay for video conferencing systems and cost of ISDN lines.

4.3.5 Hand Raising Software for Remote Questions

An important feature for students watching live lectures over the web is the ability to ask questions online. Several new applications have been introduced into the market in the last six months that can be used for this purpose. One of the friendliest is "Instant Message" which is bundled with Netscape's Communicator 4.04. This application creates a pop up window every time a message is received in the host computer, containing the sender's text message.

For this configuration to work, a professor would need a computer next to him while he was lecturing and stop every 10 minutes or so, depending on his teaching style, to take questions.

4.3.6 Online Teaching Assistant

Through the advent of free tools like MS-NetMeeting it is now possible to have a face to face virtual meeting over the internet. These new technologies can be exploited to ask a question to a TA online from an off-campus location. The advantage of these tools is that in addition to audio and video there is also application sharing. Thus, a student can ask a question and at the same time show the work he is doing.

The same technology can be used to collaborate with other class members from remote locations.

4.4 Security

At the beginning of the 1998 spring term, over 50 computers were broken into at MIT in less than 3 days. This attack was not unique to MIT, over the months that followed the Pentagon, Berkeley, Lawrence Livermore Labs, and lately the Department of Commerce have been broken into. Even properly configured firewalls have not stopped attacks. The Rochester Institute of Technology configured its firewall to filter out broadcast attacks, but experts believe that hackers used an automated attack that targeted Windows users on a one by one basis.

John Vranesevich founder of AntiOnline, a web site dedicated to computer security, feels hacker groups are targeting high profile sites. Typically, sites that people associate with being very high tech like MIT, Berkeley, national labs, and well-known military sites.

Persistent attacks have even caught the attention of the government. In a speech at Lawrence Livermore National Labs in California, U.S. Attorney General Janet Reno said: "Our systems are more vulnerable than ever ... because of our unprecedented reliance on technology". She also advocated the creation of a new government organization to serve as the government's central command center for responding to attacks on government networks and systems, on private banks, utilities, and telecommunications providers [8].

4.4.1 Internet Security Threats

The internet is a global web connecting more than a million computers. Information sent through the internet passes through many computers on the way to its final destination

This infrastructure is ideal for collaboration but it is also a risk for sensitive information. The security threats are:

- Eavesdropping:** Information is retrieved and privacy is breached.
- Modification:** Information is captured and altered. The recipient is unaware of the alteration.
- Impersonation:** The information sent goes to a user who is impersonating the true recipient.

Given this threats the security solution for sending data over the web must protect *confidentiality*, *data integrity*, and sender *authentication*.

Security for the course delivery system was planned in three phases. The first would be basic web authentication. The second, secure socket layer (SSL), and the third, the Kerberos system in conjunction with SSL.

4.4.2 Secure Socket Layer (SSL)

SSL was originally developed by Netscape Communications Corp. and uses RSA Public Key Cryptography for Internet security.

SSL encrypts data that will go over the internet. Encryption is the process of transforming information into a format that can not be read (decrypted) by anyone but the intended user. This unreadable format is called ciphertext.

In public key encryption, which Netscape uses, two keys are used. The first, called the public key, encrypts data while the second, called the private key, decrypts it.

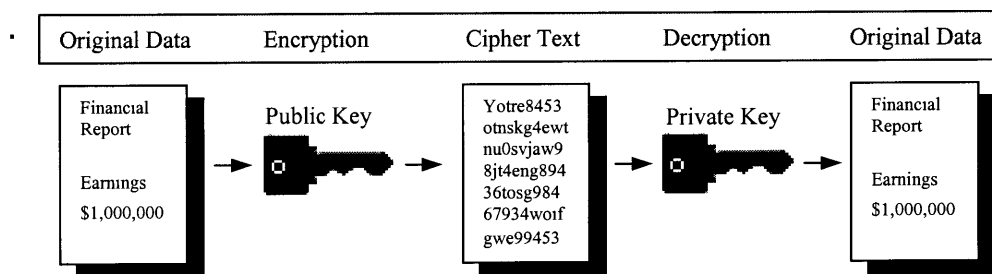


Figure 4.4.2-1 Public Key Encryption

The security of this system lies in never sharing the private key with anyone. The encryption keys are complex mathematical functions that are straightforward in one direction. However, the opposite is not true, it is practically impossible to reverse.

SSL encrypts data by performing three services. The first is message privacy. Once the initial handshake has taken place between the server and client all ensuing transmissions are encrypted. The encryption is based on public and symmetric keys.

Secondly, the SSL protocol ensures that messages between client and server are not tampered with. Through this service SSL ensures session integrity and a secure channel during the session.

The final step is mutual authentication. In this step client and server identify each other through the exchange of X.509 certificates. X.509 is a CCITT³ standard for security services.

4.4.3 SSL Handshake

Typically, the user connects to the SSL-capable server through a common HTML page hyperlink. Most SSL-capable servers will take the SSL connection on a different port (default 443) than the one used for standard web requests (port 80).

Once the port connection is established the handshake begins. From this point on, all communication is encrypted.

At a lower level this process goes through three steps:

1. Client and server exchange X.509 certificates. The following checks are made:
 - Expiration date
 - Non-tampered signature
2. The server sends the clients its public key. The client then responds with cipher key encrypted with the server's public key. If when the server receives the key and

³ CCITT: Comite Consultatif International de Telegraphique et Telephonique

decrypts it with its private key matches its original message then the negotiation continues.

3. In the final step, the client and server determine the hash function for integrity and the message encryption algorithm. First, the client sends all the possible ciphers and then the server encrypts the session with the strongest cipher in common.

The next diagrams present the information in the previous two sections.

Secure Socket Layer (SSL)

- Cryptograph Extension of Socket
- Uses Port 443(https), 465(ssmtp),

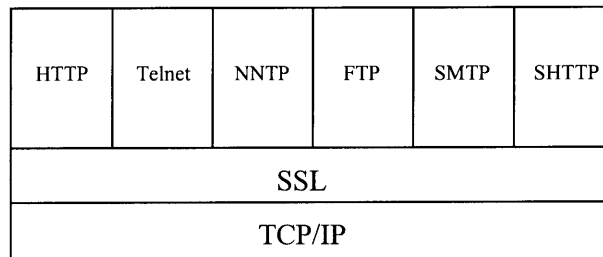


Figure 4.4.3-1 Secure Socket Layer

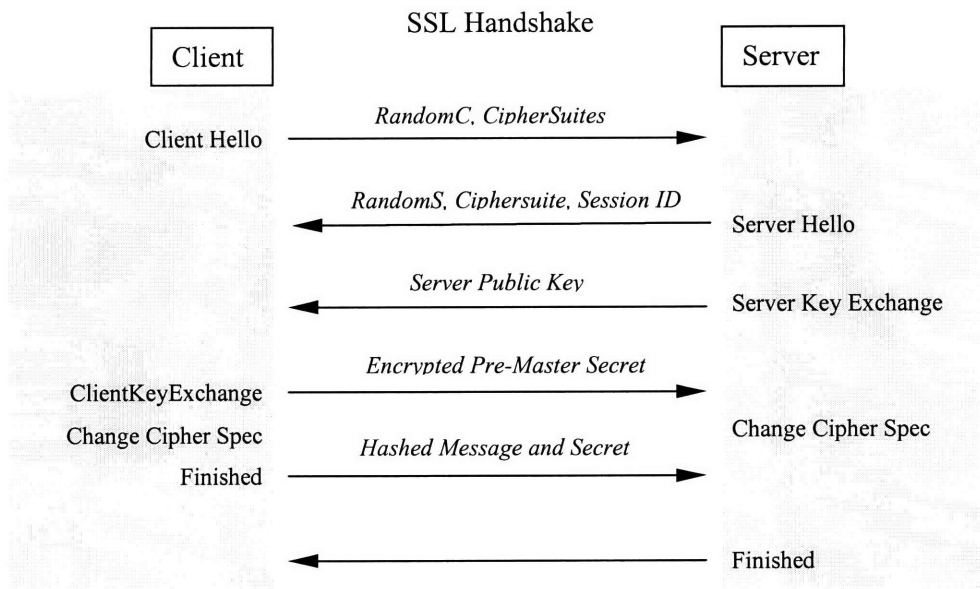


Figure 4.4.3-2 SSL Handshake

4.4.4 Kerberos

Kerberos is a network authentication system for physically insecure networks. Some of its strongest features are:

- prevention of eavesdropping
- prevention of replay attacks
- data stream integrity - detection of modification
- secrecy - preventing unauthorized reading
- data encryption standard (DES) encryption model compliant

Proof of identity over networks is established by providing a password. Exchange with the authentication server must then minimize the chance that this password will be compromised.

This is the objective of Kerberos.

Initially the password does not go over the network. What does, is a request which carries the user's name and a request to the ticket granting service (tgs), see figure below.

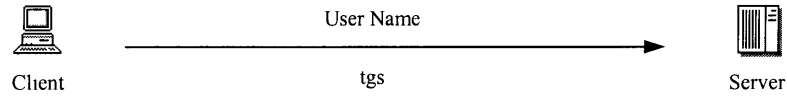


Figure 4.4.4-1 Request to Ticket Granting Service

The authentication server verifies that it knows the client. If successful, it generates a random session key to be used between the client and the tgs. The next step is the creation of the ticket

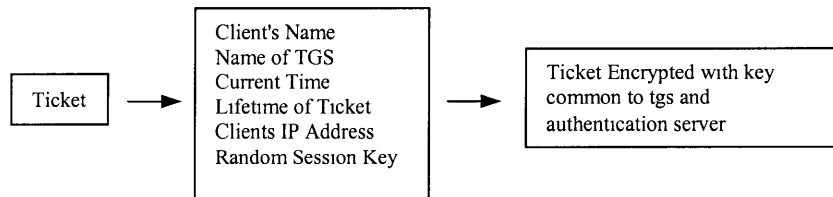


Figure 4.4.4-2 Ticket Creation

Kerberos then sends back a ticket and a copy of the random session key to the client. This message is encrypted by Kerberos with the client's private key; this key is derived from the client's password. However, it is important to note that the password itself does not travel the network. The private key is known by both the client and the server before hand.

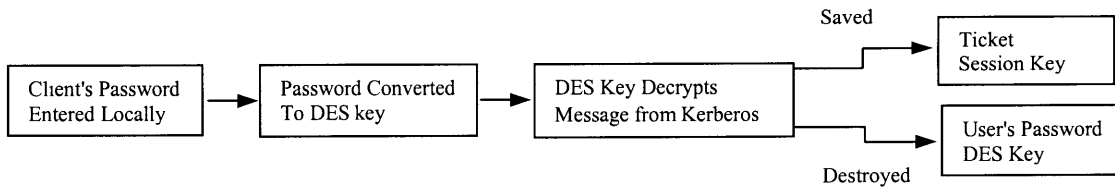


Figure 4.4.4-3 Reply to Client

Using the nomenclature below, Schiller [1] represents the transaction in the following way:

c = client

s = server

tgs = ticket granting server

Kerberos = authentication server

K_x = x's private key

$K_{x,y}$ = session key for x and y

$\{abc\}K_x$ = abc encrypted with x's key

$T_{x,y}$ = x's ticket to use with y

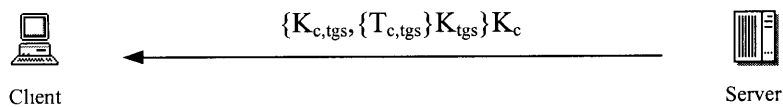


Figure 4.4.4-4 Kerberos Reply

Kerberos will be incorporated in the next version (phase 2) of COMMAND for the following reasons. It has proven itself to be a good security system (in the private and public sectors), it is

used by MIT (tying into MIT's security model will be easier), Windows NT 5.0 will have native Kerberos support.

4.4.5 Final Words

System security is a never-ending task. System administrators must be ever vigilant. Build encryption into all their data systems. Stay current; cover holes as soon as they are discovered. Apply operating system patches and fixes as soon as they become available. Monitor loads and access on application critical servers, go as far as setting alarm systems to alert system administrators when the server loads are atypical.

CHAPTER 5

Designing for Communication: Layout, Structure, and Navigation.[†]

This chapter presents guidelines for the design of the web site structure. It discusses graphic and text design in the context of graphical user interfaces.

5.1 Structural Guidelines

Once the features in the application have been determined the next step is to construct the containing structure. The designer should think on structure from the conceptual stage and revisit often his vision of structure.

As the structure begins to take form it is useful to simulate the users experience to the design group and get feedback on shortcomings. Team meetings are a good opportunity to present potential site structures on whiteboards or overheads. In this setting, everyone can contribute and quick changes can be incorporated.

Presenting information in familiar forms helps the learner feel comfortable. Fortunately, there are many classic informational structures that work well. Among these are:

- linear or temporal (e.g., process)
- topical or conceptual (e.g., perspectives)

[†] The content of this chapter has been adapted from T525 [5].

- hierarchic or categorical (e.g., analysis)
- logical (e.g., persuasion, discussion)
- pedagogical (e.g., tutorial)
- spatial (e.g., organization chart)

The forms of the structure should be varied and take advantage of web possibilities. For example, JavaScript can be used to recreate the Windows 95 desktop and plugins can create interactive structures. Also, the virtual reality modeling language (VRML) can be used to create a varied landscape and environments that can be manipulated.

5.2 Explicate Structure

A web site structure is it self a form of content. Therefore, the same care that is taken to explain content should be taken to explain structure. Explanations should not just be in the form of graphics but also in the form of strategically placed help through out the system.

The structure should be such that the learner knows where he/she is at all times. Transparent navigation will make the learner feel more secure in the web environment. It will also minimize or obliterate the time spent learning site structure.

Using common standard structural practices help the user by tapping into knowledge he already possesses. Examples are chapters, sections, introductions, headings, and other familiar forms.

Links to the home page are sometimes overlooked by designers. The learner must always be able to go home from any part of the site. This simple feature encourages the user to explore the site with out fear of getting lost.

5.3 Emphasize Central Structure

Most sites have complex file structures on the developer's side. It is the job of the designer to mask the file structure and present a central, one dimensional, continuous structure

Large jumps in the site structure should be hidden from the average user. Overlooking this point, can create discontinuities in the structure, through which, users unfamiliar with the site structure can lose their way. Thus, large jumps should be confined to a site map or made inaccessible to new users.

Layering can be used to provide information to the user, on demand. JavaScript permits an additional layer of information to be written into a web page. Designers use this layer to write help on a "onMouseOver" function. The help text is displayed when the user places the mouse over the hidden layer

Finally one must remember that with bookmarks users can access a site at any place. This means that regardless of where a user enters the site he/she must get some sense as to the structure and where he/she is.

5.4 Graphics Design

Graphic Design considerations:

- organizing the visual field
- guiding the hand and eye within that field
- encouraging exploration by creating an appealing visual environment

- ◇ unified (and thus comprehensible)
- ◇ varied (and thus interesting)
- ◇ balanced (and thus satisfying)

Variables Associated with Graphics:

- Location (relative to screen, other elements)
- Size (typesize, graphic size)
- Style (typeface or artistic style)
- Emphasis or visual weight (density, line weight)
- Texture
- Color
- Whitespace (amount, shape)

There are four principles of graphics design that will be discussed: contrast, repetition, alignment, and proximity.

The *contrast* of graphics can be a powerful tool to convey differences in a graphical user interface. Strong differences in size, style, weight, texture, and color send a clear message to the user saying “this are different”. However, in order to avoid clutter and overwhelming the user, the designer should take care to use strong elements sparingly, try to use them only to point out important features, and create a focal point. This way the user will quickly see the important points with out being distracted by the cosmetic aspects.

Repeating elements can imply unity. Emphasizing consistent elements such as navigational bars, headers, typefaces, colors, icons, locations creates a “feel” for a particular web site. It is the

designer's job to create an environment that will be consistent through out the entire site. A web site has achieved this goals if the user can immediately know if he/she is still in the site by looking (not reading) at a page.

Alignment can create visual connections between elements. It aids the organization of a page by directing the eye and it's important both in text justification and the alignment of images. Mixed text alignment should be avoided as it misdirects the eye. Also, alignment at a page edge is undesirable.

Organization can be improved by *grouping* related elements. The natural tendency of a viewer is to look for relationships and the proximity of elements implies relationships. A good example, is a paragraph header, the closer it is to the body of text, the clearer the message that they belong together. In general, try not to use more than 3-5 eye-stoppers groups.

5.5 Text Design

Text design (choosing font, style, color, weight) in web environments is limited unless the text is an image. (Users control default font, size choices; HTML specifies only relative sizing, and even specified fonts may not be available). Given this limitations consideration will only be given to:

- Concordant type
(one type family, *little variation* in style, size, weight...)
- Conflicting type
(two different type families, *similar* style, size, weight...)

- Contrasting type
(two different type families, *distinct* style, size, weight...)

In most situations the only true font type variations are serif and sans serif.

Concordant layouts are very soothing and should be used to achieve a calm and formal effect.

The opposite practice creates a disconcerting feeling on the user. However, contrasting type elements such as monotype vs. proportional can be used as a simple way to achieve contrast (outline differences) in design.

Web pages differ from software screens and print media in the following ways:

- Variable page size. The designer can not be sure if the user will view material at 600x400, 800x600, 1024x768, 1152x864, or 1280x1024.
- Primacy of the top four inches: people will not scroll. Thus, the top four inches must engage the user.
- Bandwidth limitations of users. In some cases limited bandwidth will eliminate many TCP/IP applications. For example, real time video is virtually impossible to do over 14.4 kbps connection.
- Client Browser setup. Users on occasion may disable Java, JavaScript, and Cookies. Other may choose unusually large or small fonts destroying an otherwise effective layout.
- Version of HTML the client browser will support. Currently only the last releases of the major browsers support dynamic HTML (DHTML) a feature of HTML 4.0. In contrast it is still possible to download browsers that are several generations old and that only support HTML 3.0 or HTML 2.0.

Thus, due to the many configurations available to the user a developer must do extensive testing on different:

- operating systems
- display sizes
- browsers
- bandwidths

5.6 The Graphical User Interface

The inessential parts of a graphical user interface (GUI) are the page layout, site structure, and site navigation. Using lessons learned from software design the discussion to follow will focus on: 1) the principle of user correctness, 2) the principle of least astonishment, and 3) the principle of user centeredness.

5.6.1 The Principle of User Correctness

When it comes to user interfaces, the common expression of retail sales applies: the user is always right. A design can meet every theoretical principle but if users are confused, misled, or disconcerted by the GUI, it is useless. A successful design should require no explanation.

An effective GUI is a difficult task. The designer should not test his own GUI. Designers are not typical users. This is especially true when one person does all the development. The problem lies in familiarity; the designer becomes so familiar with the project that a logical navigation and transparent content are no longer necessary.

5.7 The Principle of Least Astonishment

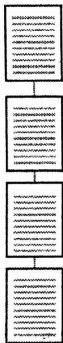
Some of the key elements in a predictable user interface are: simplicity, clarity, completeness, consistency, and robustness.

5.7.1 Navigational Bars

When designing navigational aids such as bars, preference should be given to simple structures.

Some of the least complicated and familiar to users are the horizontal and vertical structures (see below).

Vertical



Horizontal

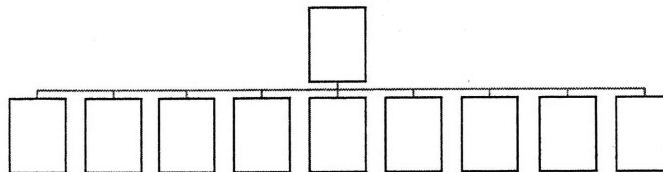


Figure 5.7.1-1 Navigational Structures

The key is making the structure apparent, easily comprehensible, and keeping the navigation decisions at any point, simple. For example, horizontal and vertical navigation styles could be combined as long as the end result was apparent to a casual user. In the example below, numbers are used to give the user a navigational reference.

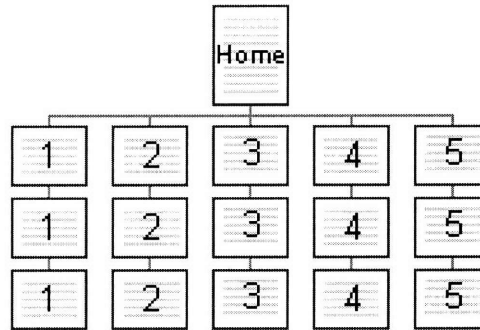


Figure 5.7.1-2 Navigational Aides

Other ways of promoting simplicity are: reducing the number of controls, no more than five is a good principle. Group related controls, separate control groupings that have different functional types, and use contrast to highlight the important controls

5.7.2 Clarity

The purpose and use of controls should be immediately clear. Important buttons should dominate the page. The designer should take advantage of web standards such as arrows, text navigation bars, house icon for home button, and rollover scripts. Another important aspect is to make use of the users natural tendencies. For example, people in western cultures search a page left to right and top to bottom. Thus, a good place for an essential piece of information is in the top left hand corner.

5.7.3 Completeness

Control buttons for essential features, should always be present. The designer must identify the key functions and incorporate them to every view the user will see. It is also a good idea to provide a link home from every page. In many occasions a user will enter the site at a random

point (as a result of a search or a bookmark) and without a link to the home page the user can become lost or confused as to its location.

Consideration should also be given to how a page will print. If the content of the pages will be printed often, the designer should take into consideration a color scheme that will be readable in gray scale. A better solution is to have an additional version (different format) of the information for printing purposes only. Some of the common current formats are portable document format and postscript.

In order to create a complete interface the designer must take into account the client's browser. Many browsers do not support the latest versions of HTML and thus may display nothing or incomplete information. One strategy used to avoid this problem is to discover the client's browser when the HTTP server is propositioned. Once the browser is determined the designer can conduct the client to the appropriate web page. Another option is to present a text only welcome page with several browser options.

5.7.4 Consistency

The user should be able to know if he is in the web site by looking at the page (not reading it).

Similarities can be supported using appearance, placement, and operation. It follows that all aspects of the site should appear and function consistently. Successful implementation will align the designer's conceptual organization with the user's perceived site structure.

5.7.5 Robustness

Users will commit errors and perform unexpected tasks. Therefore, the designer must anticipate user actions, prevent accidents, and give helpful feedback when errors occur.

5.7.6 User-Centeredness

Place the user at the center of your creation.

- Control: give the user control, or the appearance of it
- Responsivity: respond immediately to user actions, and keep him/her informed
 - ◇ Keep download times short
 - ◇ specify height, width of graphics
 - ◇ use fast servers
- Forgiveness: (related to Robustness) assume users will make mistakes, and provide a means to correct them.

CHAPTER 6

Central Administration

This chapter presents the implementation of the proposed web model (see chapter 2). It discusses the centrally managed model needed to scale up and automate the course web sites. It also discusses registration, site generation, access model, platform independence, and high level features.

6.1 Automated Site Generation

In the past, the generation of a web site for a course was something that had to be planned carefully. Typically, several meetings took place between the professor and the web designer to decide:

- structure
- look
- feel
- features included
- update procedure
- host for site

This process can take weeks to months depending on the experience of the designers and how ambitious the project goals are.

6.1.1 Site Generation

In the COMMAND system this process is eliminated. In its place, the course professor simply fills in his/her contact information (see below).

The screenshot shows a web form titled "New Course Website Request" with a "common" logo in the top left. Below the title, it indicates "✓ = Required Fields". The form contains the following fields:

- ✓ Instructor's Command Name: A text input field.
- ✓ Instructor's E-Mail Address: A text input field.
- ✓ Departmental Affiliation of Course: A dropdown menu currently showing "-Select One-".
- ✓ Course Name: A text input field.
- ✓ Course Number: A text input field.
- ✓ TA(s) Command Name: A text input field.
- ✓ TA E-Mail Address(es): A text input field.

A "Submit Request" button is positioned at the bottom center of the form.

Figure 6.1.1-1 Course Request Interface

Filling in this form and submitting it over the web is all that is required for the professor to have a course web site on COMMAND. Once the course is processed (24 hours) the teaching staff is notified through e-mail to begin populating content.

On the administrative side the creation of the database entails: verification of identity, creation of databases, and universal resources locator (URL) redirection.

A check is run on the identity on the person making the request. In order to avoid malicious users from making phantom course requests a quick check is run on the MIT information system to verify the identity of the request.

The next step is the creation of the databases. The system was created to automate the creation of course creation. Thus, the creation is a one step process.

In order to make navigation easier for users a simplified URL shortcut is made to each course that is created. The standard is the same for all classes: domain name plus class number. For example, for course 1.00 the shortcut is <http://command.mit.edu/1.00>.

6.2 No Technical Knowledge Required

An advantage of the system is that no technical knowledge is required by the professor or the teaching assistant, wizards on the web's interface will guide the user through the setup.

Administrative tasks such as grading homework, archiving outdated information, and creating design groups have a graphical user interface (GUI) which reduces these tasks to mouse clicks.

This structure permits professors with low computer skills to have an advanced course site. While a comparable system can be created by a professor from scratch, the time commitment that will go into developing one is very high, plus it will often be tailored for one specific course. The COMMAND system meets the needs of professors who want a high end site but do not have the time or skill necessary to develop a system on their own.

6.3 Central Management Model

6.3.1 COMMAND's Home Page

From COMMAND's home page the user can register, interact with a fully functional demo course, request a web site, and learn about the system. Once the user is registered, and courses have been setup, the homepage will continue to act as the gateway to system services. Such as student folder, course lists, and profile updates.

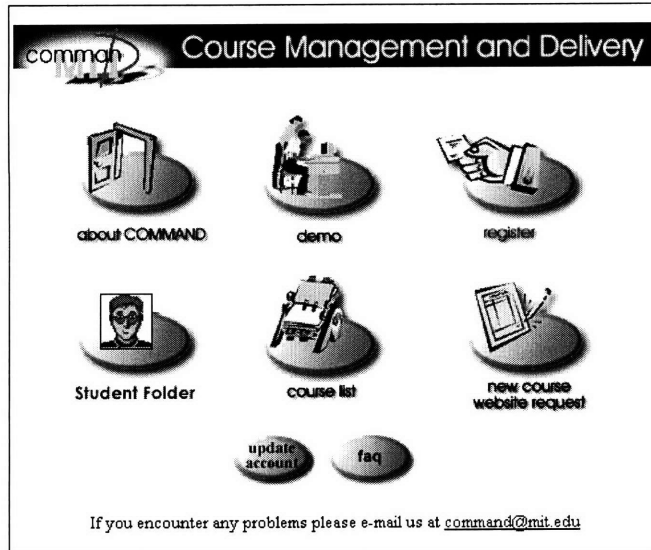


Figure 6.3.1-1 COMMAND's Home Page

6.3.2 Registration

The first step in becoming part of the COMMAND system is setting up an account (see below).

Registration is done only once and the user's identity is recognized through out all of COMMAND's courses and system services.

New Account

First Name: ✓
 Last Name: ✓
 Password: ✓

Phone Number: ✓
 E-mail: ✓
 Home Page URL:

Please select the courses in which you would like to participate. Use the <CTRL> key to select more than one.

1.120
 1.130
 1.201
 1.202
 1.204
 1.206
 1.231J
 1.432
 1.44
 1.441

1.120
 1.130
 1.201
 1.202
 1.204
 1.206
 1.231J
 1.432
 1.44
 1.441

✓= Required Fields

Figure 6.3.2-1 Registration

Once users are registered they can then make full use of the site services. Professors can request courses and students can go to their folders and link to their classes or review their upcoming homeworks.

6.3.3 Student Folder

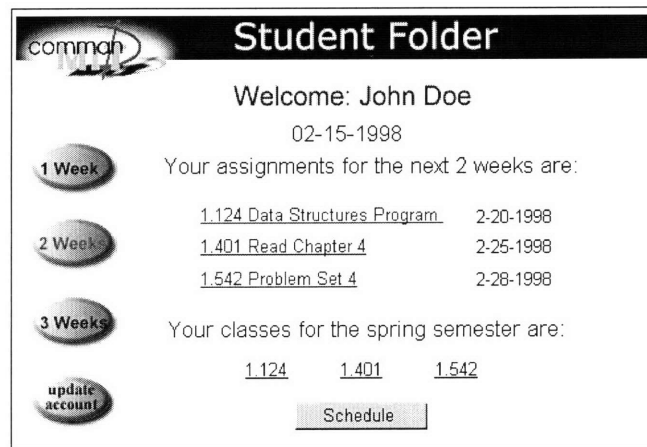


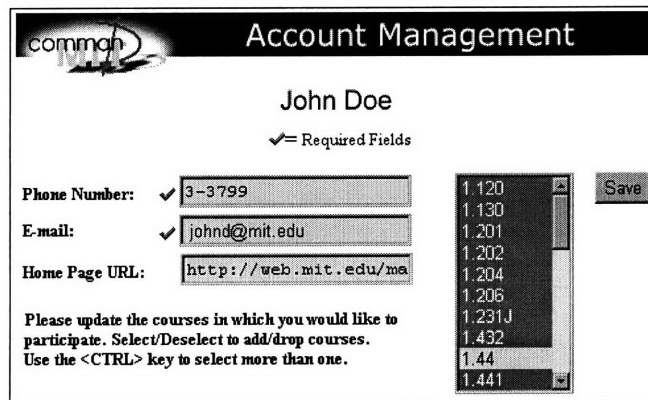
Figure 6.3.3-1 Student Folder

The system will go to each individual class and pull the assignments for the specified dates. Links to the description of each assignment are provided as well as the due dates.

One of the documents that is used most often by students is the course syllabus. It is checked week after week to check what assignments are coming up, when the next exam is scheduled, and term project deadlines. Thus, a web page that can extract and display all of this information at the click of a button, for every course a student is registered for, is very valuable. This feature is expected to save students considerable time spent looking for information.

The student folder also presents links to the classes the users is enrolled in. This feature provides a convenient access point to a course site for users who do not the URL to their class or students who want to know their class schedule.

The ability to update accounts is also present in the COMMAND system. As the semester goes on students add and drop courses. Hence, it is important to have a flexible system where updates can be made easily. Pressing the "update account" button, on the welcome page, provides the following interface to make changes.



command Account Management

John Doe

✓= Required Fields

Phone Number: ✓ 3-3799

E-mail: ✓ johnd@mit.edu

Home Page URL: http://web.mit.edu/ma

1.120
1.130
1.201
1.202
1.204
1.206
1.231J
1.432
1.44
1.441

Save

Please update the courses in which you would like to participate. Select/Deselect to add/drop courses. Use the <CTRL> key to select more than one.

Figure 6.3.3-2 Account Management

6.3.4 Help

Help for COMMAND is provided in various formats, including video instruction. For people who are new to COMMAND short videos describing how to perform routine COMMAND tasks are available. These videos, take the user step by step and screen by screen. Example, videos are "About COMMAND", "Registration", and "How Submit Homework online".

These videos have been optimized for transmission over the web. Due to small file sizes and video streaming, viewing can be done over typical internet service provider connections. Testing

has been done at 56 kbs, and T1 connections. With the exception of overseas connections all tested positively and user response was enthusiastic.

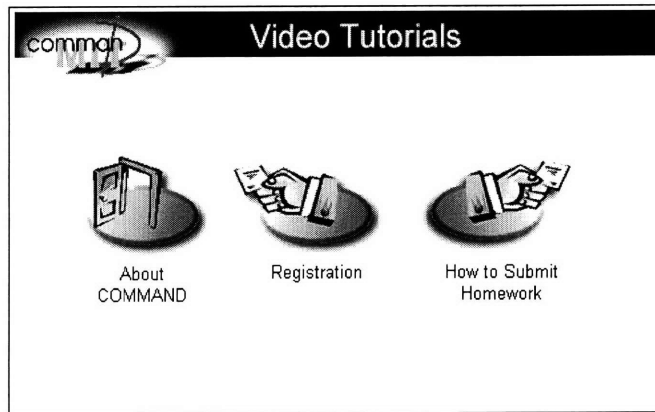


Figure 6.3.4-1 Video Tutorials

6.3.5 Central Structure

Due to the central structure of COMMAND, users can access all services through one URL. At the end of phase one, COMMAND had over 50 different screens (URL addresses) a user could access excluding content contributed by faculty to the site. Thus, a structure that presents all of this information in a manner that is easily processed by the viewer is essential for an effective system. After reviewing several models the development team found a central structure to work the best during testing.

The central structure is represented in the following tree diagram:

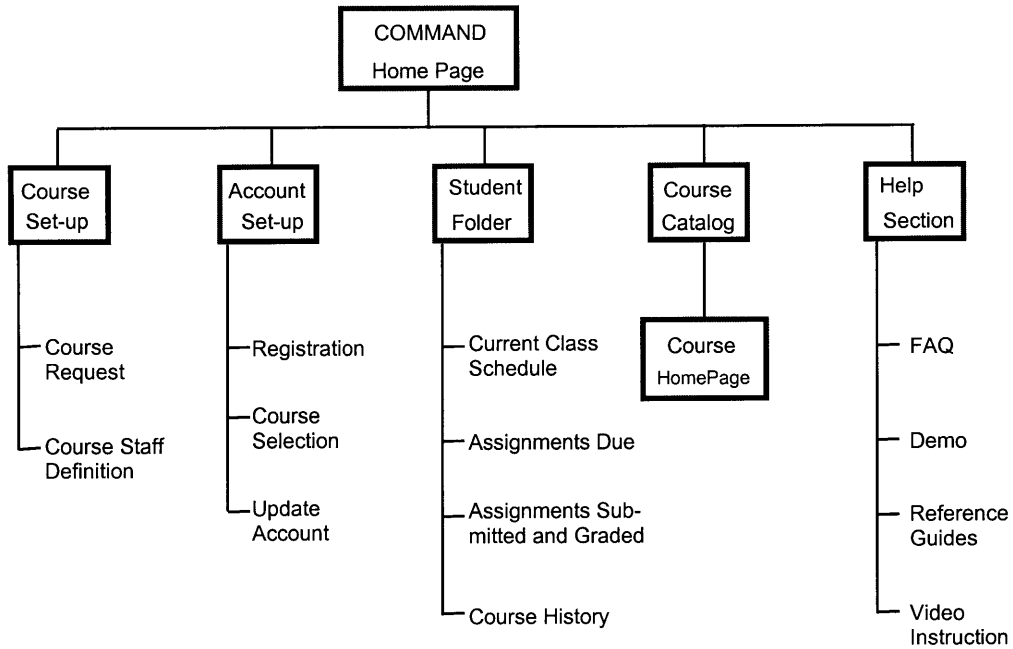


Figure 6.3.5-1 COMMAND Central Structure

6.3.6 Access Model

Access to COMMAND is divided into three levels: instructor (course staff), students, and system administrators. Access privileges are detailed in the following diagram:

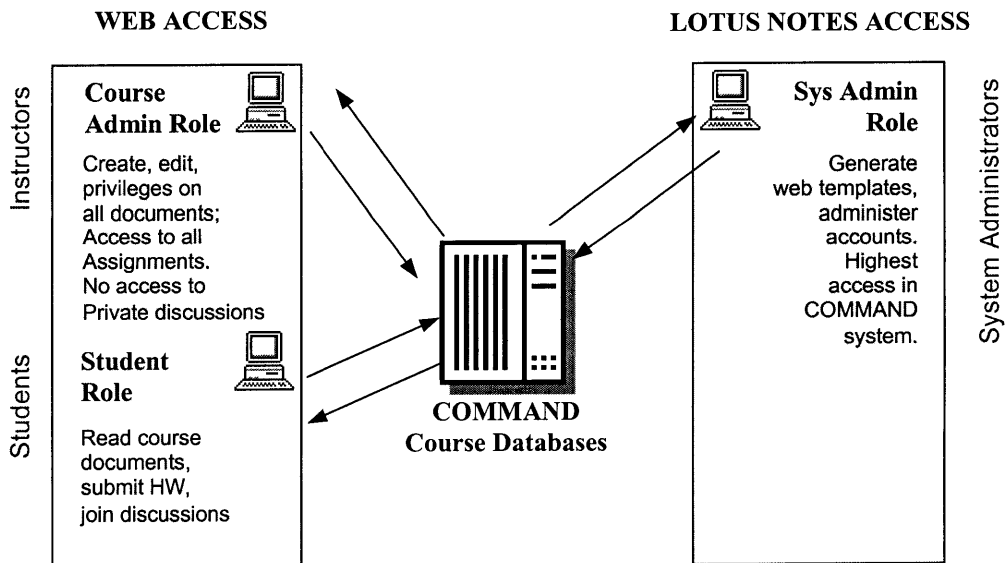


Figure 6.3.6-1 Access Model

6.4 Platform Independence

MIT is a platform diverse computing environment (see diagram below). Administrative groups clinging to the Mac OS even as market share continues to decrease. There is also Athena, MIT's UNIX-based campus-wide academic computing facility. The Information Systems (IS) office reports that "the Athena system provides computing resources to over 18,000 users across the MIT campus through a vast system of 1,300 computers in more than 40 clusters, private offices, and machine rooms, all connected to a campus-wide network". In addition, MS Windows networks started slow but are coming on strong, IS has started work on Athena NT, a system that would serve the current offerings of Athena on an NT platform. Currently, several Windows-NT labs have open across several departments and a large number of private NT domains have been created over the last year.

6.4.1 Testing Platform

The nature of MIT's computing environment made it a very good testing community. During the initial test phase incompatibilities with platforms were discovered quickly due to the diversity in testing platforms. Thanks to this, most of the incompatibility issues were solved in the early stages of development.

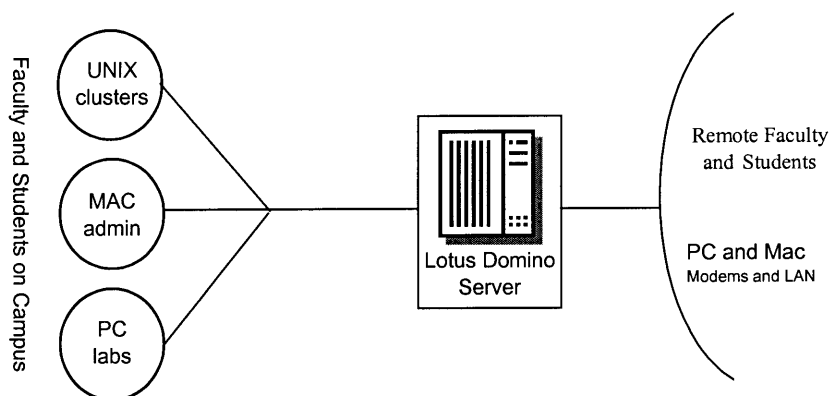


Figure 6.4.1-1 Platform Independence

6.5 Security

Security for the COMMAND system was planned in three phases. The first would be basic web authentication. The second, secure socket layer (SSL), and the third, the Kerberos system. For a discussion of Kerberos and SSL please see chapter 4.

6.6 Video Streaming

The promise of video on the web has been in the horizon for some time. The media presents glitzy commercials of video conferencing with crisp picture perfect quality. However, up to now this promise has failed to become reality. Video is often painfully slow to download and even when it does the quality leaves a lot to be desired.

Why consider it then? Well, the scenario of MIT is a little different than the average web user. To start with students are blessed with T1 connections to their dorm rooms. Also, most remote students are part of large corporations which have similar connections as part of their infrastructure. Thus, the higher bandwidth can make the transmission of video audio a functional reality.

Also, advances in the last two years, by companies like Vxtreme, Real Networks, and Microsoft's Netshow, have greatly improved the transmission of audio and video. During testing the author was able to encode 100-megabyte files at 2.5 megabyte with results that were not as good as the original but were more than satisfactory for lectures or system help.

The combination of higher bandwidth and advances in compression made video streaming an attractive technology. The uses planned were: 1) help clips; 2) record lectures; 3) live encoding.

Help clips and test lectures were created for the first phase. Live encoding was left for the second phase.

CHAPTER 7

The Course Database

This chapter goes into the specifics of the course web sites. It discusses the organization, course documents, homework administration, collaboration, and administrative tools.

7.1 Organization

7.1.1 Structure

Course web sites are broken into five major categories: course documents, homework administration, collaboration, administration tools, and video streaming.

The structure of the course web sites is presented in the following tree structure:

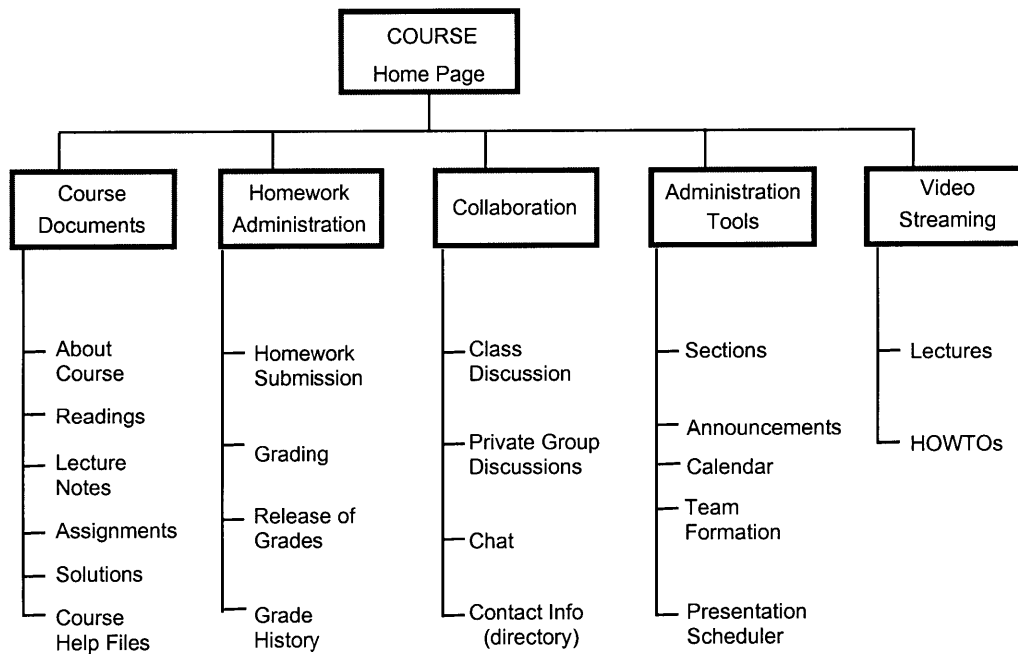


Figure 7.1.1-1 Structure

Each major category will be discussed in this chapter.

7.1.2 Course Home Page

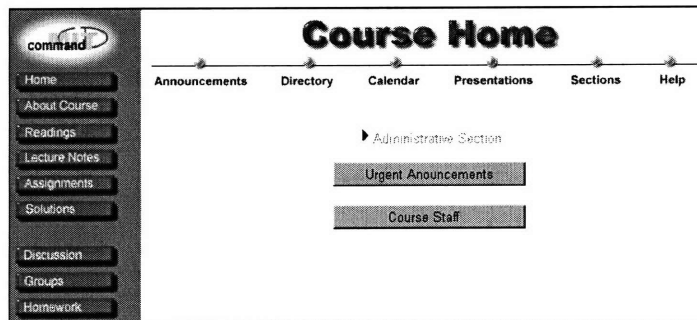


Figure 7.1.2-1 Course Home Page

The home page provides links to all possible parts of the course database. It also contains an administrative section which can only be viewed by the course administrative staff. When logged on as course staff the administrative section expands as shown below:

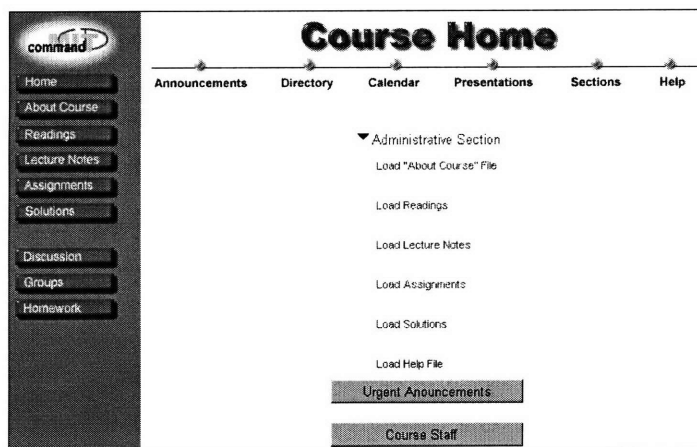


Figure 7.1.2-2 Course Home Page - Administrative

A heavily used feature of the home page is the "Urgent Announcements" button. Pressing on this button brings up a pop-up window (see below) with last minute announcements. This feature has been a popular one due to its easy access for students and ease of use for administrators. Entering information is simple, all they do is press the edit button, type in the announcement, and press submit. For students the effort is even smaller, one click of the button.

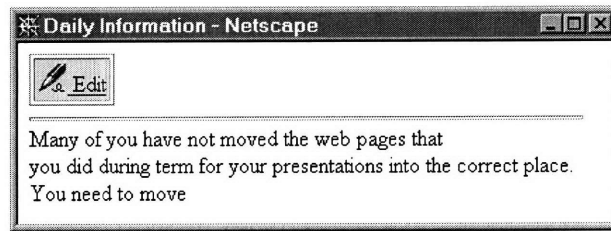


Figure 7.1.2-3 Urgent Announcements

The other button labeled “Administrative Staff”, on the home page, provides information about professors and teaching staff. It provides name, e-mail, room number, office hours, and home page if available. This information was placed on the first page as it is accessed very often during the semester.

Course Staff				
Name	e-mail	Room No.	Office Hours	Home Page
Dr. John Doe	johnd@mit.edu	1-999	MWF 1-2	http://web.mit.edu/johnd
Dr. Jane Doe	janed@mit.edu	1-888	T TH 1-2	http://web.mit.edu/janed
TA Jungle Jim	jungle@mit.edu	1-950	T TH 4-6	http://web.mit.edu/jungle
TA Joe Bow	joebo@mit.edu	1-950	T TH 4-6	http://web.mit.edu/joebo

Figure 7.1.2-4 Course Staff

7.2 Course Documents

Content uploaded by professors is divided into five categories: about course, readings, lecture notes, assignments, and solutions.

The development team expected to have requests for additional categories once the databases were in use. However, the provided categories proved to be sufficient at the current scale (31 courses).

7.2.1 About Course

This category contains documents that describe the course. The types of documents that have populated this area are syllabus, course objectives, and term project descriptions.

To upload a document into this section the user needs to click on “Load About Course Files” in the expanded administrative section. Once this has been done the following interface appears.

The screenshot shows a web interface for uploading course documents. On the left is a vertical sidebar with a 'command' logo and buttons for 'Home', 'About Course', 'Readings', 'Lecture Notes', 'Assignments', 'Solutions', 'Discussion', 'Groups', and 'Homework'. The main area is titled 'About Course' and contains a form with the following fields and instructions:

- Category:** Grading
- Title:** Percentage Breakdown
- To submit a file:**
 1. Press the "Browse" button.
 2. Choose your file from the dialog box.
 3. Press the submit button at the bottom of the page when you are done.
- File to submit:** c:\course_documents\grading.html
- Buttons:** Browse... and Submit

Figure 7.2.1-1 About Course Form

Subcategories can be created by making an entry under “category”. Once the title has been entered the “Browse” button will read the host machine’s hard drive and the user can choose a document to upload. For the given example the following entry will be made into the “About Course” category.

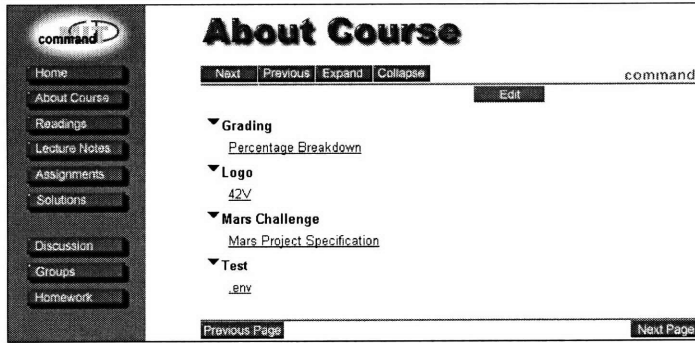


Figure 7.2.1-2 About Course Documents

Additional categories follow the same pattern. The only difference in the upload procedure is that a number is assigned to the uploaded document. For example, when uploading the reading “Sabre Case Study”, the number one is assigned, this then becomes reading 1 in the “Readings” section.

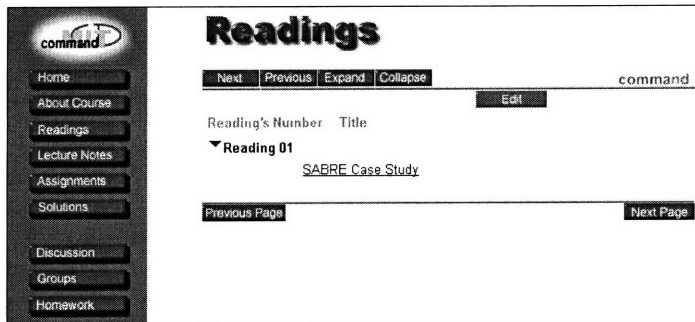
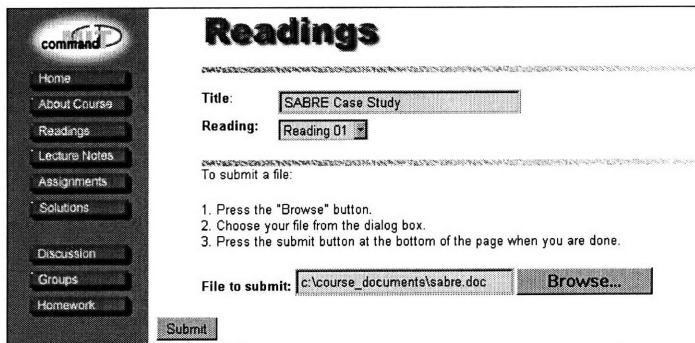


Figure 7.2.1-3 Readings

Screen shots for the remaining categories are shown below

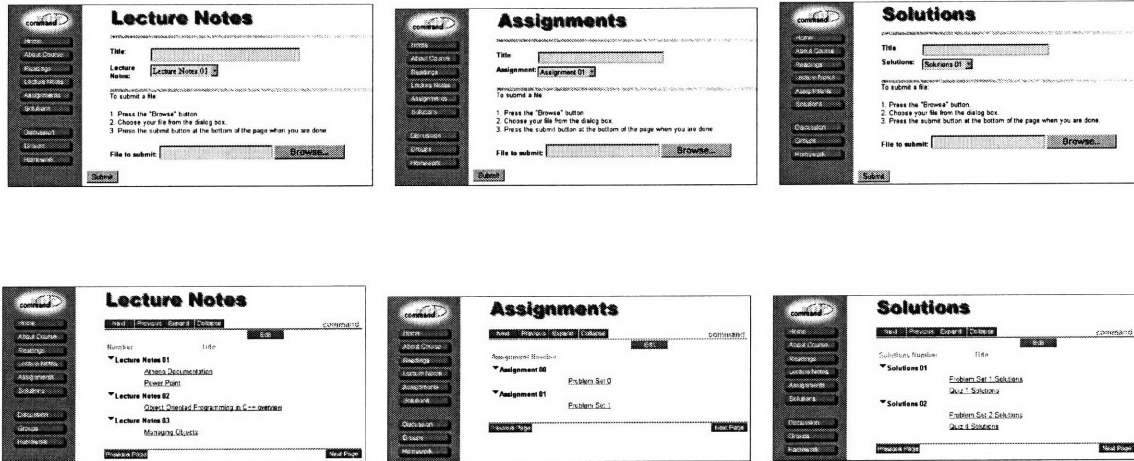


Figure 7.2.1-4 Lecture Notes, Assignments, and Solutions

7.3 Homework Administration

Homework administration provides features that permit homework to be submitted and graded online.

7.3.1 Interface

Homework can be submitted over the web using the following interface:

 The screenshot shows the 'Homework' submission form. It features a sidebar with navigation links (Home, About Course, Readings, Lecture Notes, Assignments, Solutions, Discussion, Groups, Homework). The main form area includes:

- Buttons for 'Submit' and 'Cancel'.
- Fields for 'Submitted by' and 'Evaluated by' with a timestamp 'on 4/06/08 10:58 AM'.
- A dropdown menu for 'Homework No./Title' with '<Select one>' selected.
- Fields for 'Status' (containing 'Submitted') and 'Grade'.
- Two options for submission:
 - 'Option 1. Enter text here, then press the Submit button.' followed by a large text input area.
 - 'Option 2. Attach a file from your PC' followed by a 'Browse...' button.

Figure 7.3.1-1 Homework Form

The student has two options: 1) attach a file; 2) enter text directly on to the form. The second option provides for the case when the format of the attached files between professor and student

becomes a problem. Anything entered in this box will retain the plain text format and thus can be read by any system, format, or word processing program.

7.3.2 Late submissions and submitting a new version of the homework.

Since all homework is date stamped, the grader can immediately determine if the homework is late or not. In the case where the student submitted an assignment early and wants to turn in a new version of the same assignment, the system will allow him the option of replacing the homework he had previously submitted.

Once the homework is submitted the grader can grade each homework and hold publishing the grades until the homework of all students has been graded. There are several types of holds, one is "hold for teaching assistant", another is "hold for professor", and a general hold. In each case, the graded homework is not visible by the student until the status has been changed to "visible by student". The separate holds are used when a grader will not get a chance to grade the homework right away, when the grader has a question for the professor, or when there is an honesty question. The grading interface is shown below:

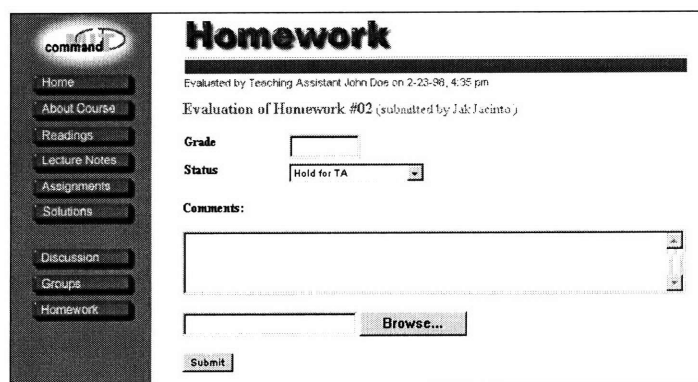


Figure 7.3.2-1 Homework Evaluation

Once the homework has been graded and made visible to the class, only the author of an assignment can read it, but the course staff can view all assignments.

The entire course assignments can be viewed by pressing on the button labeled “Homework”. A sample of this view is shown below.

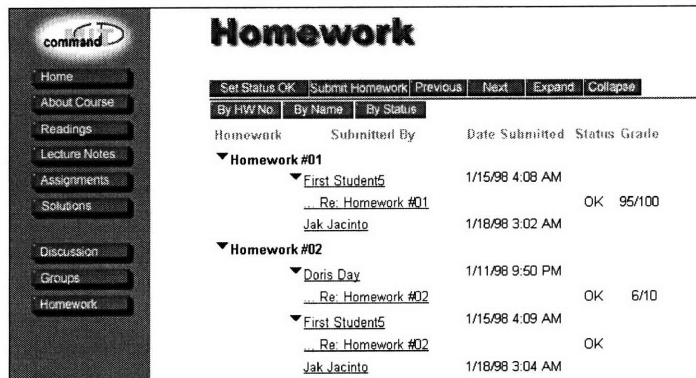


Figure 7.3.2-2 Homework Documents

7.4 Collaboration

Several features have been built into the first phase of the system to promote collaboration between students online. The most note worthy are class discussions, private discussions, chat, and contact information.

7.4.1 Discussion

Class discussion entries are made through the following interface.

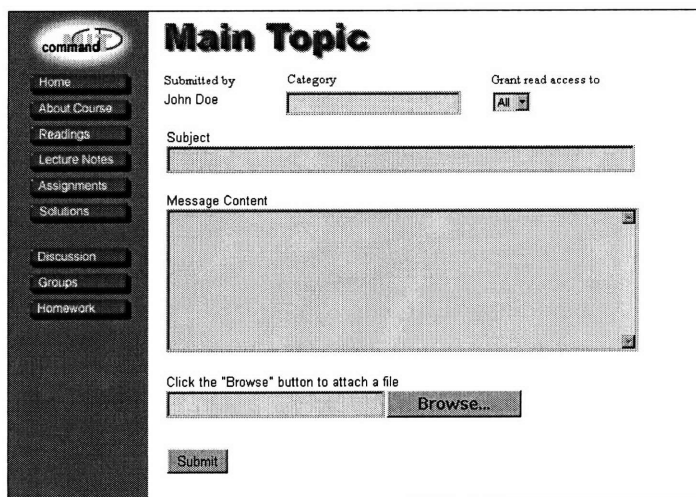


Figure 7.4.1-1 Discussion Entry

Discussion entries can be made to the entire class or to a private group (see above).

Often classes at MIT have design competitions at the end of the semester. These competitions are usually also the term project. The discussion sections serve as a history of the project and often help as a reference when the final write up is done. During the life of the project the discussion is also used to upload documents, images, or computer code that a user wants to share with the rest of his team.

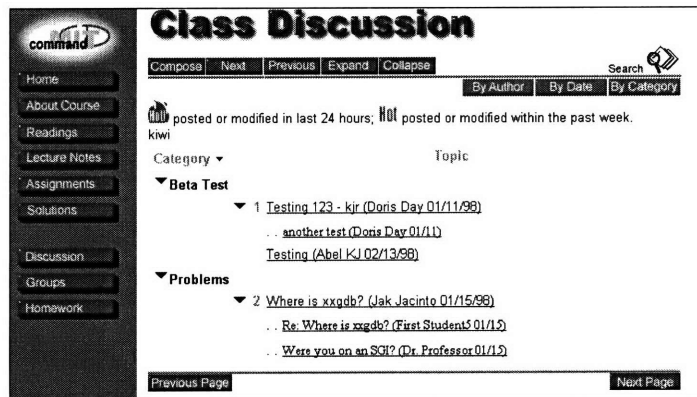


Figure 7.4.1-2 Class Discussion Documents

7.4.2 Chat

In addition to the class discussion, a live chat applet was written. The requirement for design was that it be small, quick to load (run), and have a transparent interface. The result was just that, a small applet that loaded quickly and was easy to use. The applet is shown bellow.

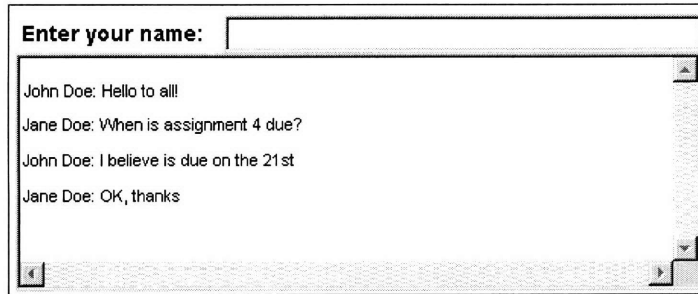


Figure 7.4.2-1 Chat

7.4.3 Contact Information

The last feature that was included to promote collaboration was a directory with everyone's contact information. This directory contained name, e-mail, and URL for all members of a course. The goal was to facilitate contact between classmates by facilitating information.

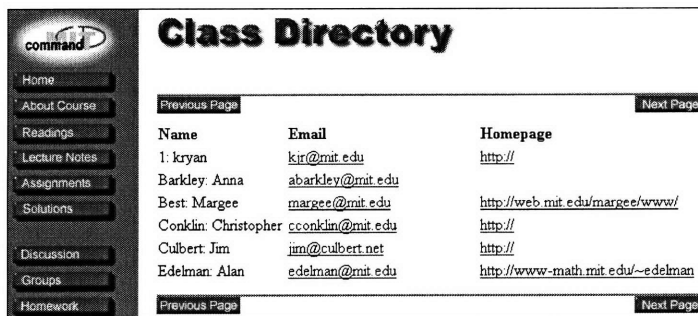


Figure 7.4.3-1 Class Directory

7.5 Administration Tools

Several administrative tools were provided to course staff and students. Some of the most notorious are creation of sections, announcements, team formation, calendar, and presentation scheduler.

7.5.1 Groups

One of the most highly requested features was the ability to create private group discussions. The following interface permits this function.

Figure 7.5.1-1 Groups Form

This interface shows the available students for private teams and the additional fields allow descriptive entries. Once a team is formed it can be viewed by clicking on the groups button on the left vertical navigation bar.


Group	Members	Send Mail	Web Site
Dinner with Neil Palm	Falker, Jay Greif, Irene Lerman, Steven	✉	www.us.ibm.com
Group Test - kjr	Day, Doris KJ, Abel	✉	
Jak & First	Jacinto, Jak Student5, First	✉	
Teaching staff	GradStudent, Struggling Professor, Dr.	✉	
The A Team	UnderGrad, Pretentious	✉	

Figure 7.5.1-2 Groups

As a security precaution, only a team member can give access to another student. Course staff was not given access to provide group members complete privacy.

7.5.3 Calendar

Presentation entries will also appear in the class calendar.

 Calendar		
May 1998	16	May 1998
4 Monday *** <small>..:..</small> <u>Presentations</u> *** <small>..:..</small> <u>Signal Estimation-Filter Banks and Neural Networks</u> *** <small>..:..</small> <u>White Noise Removal from an Audio Signal</u> *** <small>..:..</small> <u>Bring the Noise</u>	Monday 11 *** <small>..:..</small> <u>Presentations</u> *** <small>..:..</small> <u>Rhythm Perception using Wavelets</u> *** <small>..:..</small> <u>Object Analysis of Images using Wavelets</u> *** <small>..:..</small> <u>Morgan McGuire</u>	
5 Tuesday 6 Wednesday *** <small>..:..</small> <u>Presentations</u> *** <small>..:..</small> <u>Wavelets for Edge Detection in Images</u> *** <small>..:..</small> <u>Applying Wavelets to the Compression of 3-D Data</u> *** <small>..:..</small> <u>Multiresolution Computer Graphics Using Wavelets</u>	Tuesday 12 Wednesday 13 *** <small>..:..</small> <u>Presentations</u> *** <small>..:..</small> <u>Wavelets for Audio Compression</u> *** <small>..:..</small> <u>Singing voice feature extraction using wavelets</u> *** <small>..:..</small> <u>Detection of Musical Intervals using Wavelets</u>	
7 Thursday 8 Friday *** <small>..:..</small> <u>Presentations</u> *** <small>..:..</small> <u>Wavelets application in array signal processing</u> *** <small>..:..</small> <u>Wavelets for Object Detection</u> *** <small>..:..</small> <u>Eddy-Current Based Crack Detection Using Wavelets</u>	Thursday 14 Friday 15	
9 Saturday 10 Sunday	Saturday 16 Sunday 17	

7.5.4 Sections

As classes grow the ability of the professor to answer questions becomes more difficult. In a class with 300 students and 50 minutes of lecture this becomes impossible. For this reason large classes often have recitation (discussion) sections, typically this sections are only 10-15 students.

When classes are very large, scheduling all the students into recitations can become very difficult.

Up to now this process has been done by hand. COMMAND, in its first phase tries to make this easier by letting the course staff create sections on line. The second phase will extend this support, for more on this subject please see "Future Work", in chapter 4.

This interface allows the teaching staff to form sections from the course's students.

Sections

Section Profile

Section Name Required field

Description

MEMBER SELECTION

- Michael Jordan
- Magic Johnson
- Larry Bird
- Dennis Rodman
- Scotie Pippen
- Hugh Grant
- Kareem Abdul Jabar
- Charles Barkley
- Larry Johnson

Submit Cancel

Figure 7.5.4-1 Section Form

A view of the sections is also available

Sections

New Section Previous Next

New Group Previous Next

Section	Members	Send Mail	Description
A	Cadieuz: Matt Conklin: Christopher Culbert: Jim Edelman: Alan Falker: Jay Falker: John Goran: Jim	<input type="checkbox"/>	Recitation section meets MW 1-2

Figure 7.5.4-2 Sections

7.5.5 Announcements

An announcement system has been built into COMMAND. The course staff has author privileges while the rest of the class has reader access. All announcements are sorted by creation date, from new to old. Thus, as users enter the announcements screen the first message they see will be the most recent.

The announcement entry form and the announcements screen are shown below.

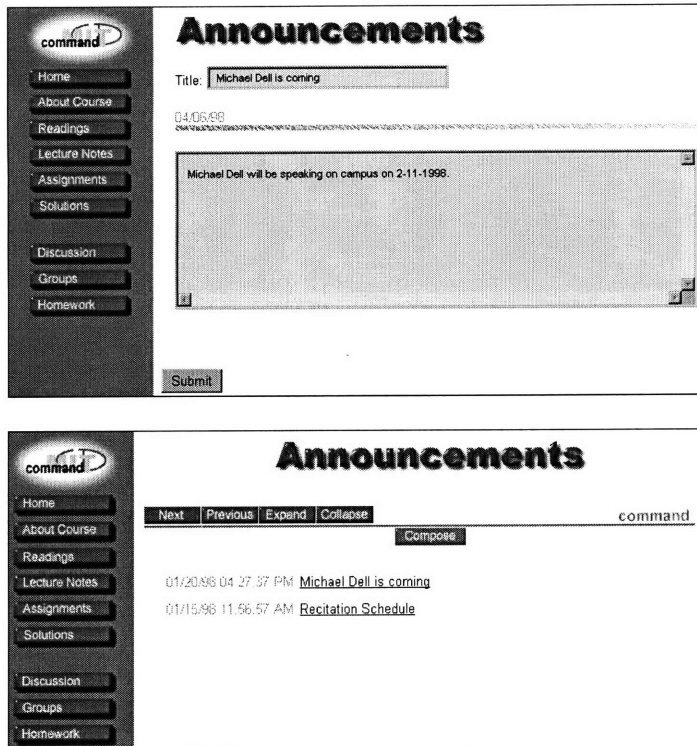


Figure 7.5.5-1 Announcements

7.6 Workflow

The following diagrams, created by Hidalgo [6], depict the workflow involved in performing the most common tasks.

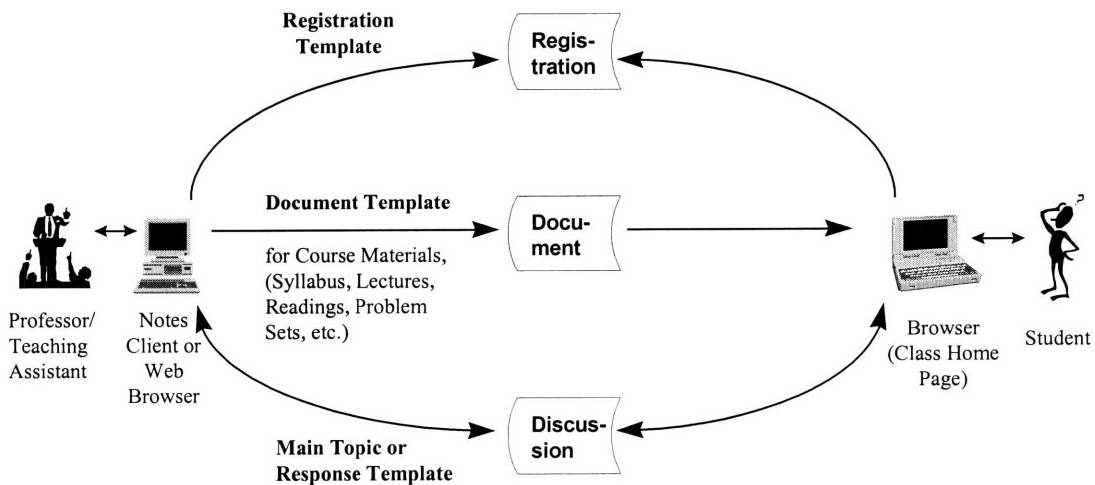


Figure 7.6.1-1 Workflow Showing Databases, Templates, and Documents that Support Registration, Course Material Management, and Asynchronous Discussion

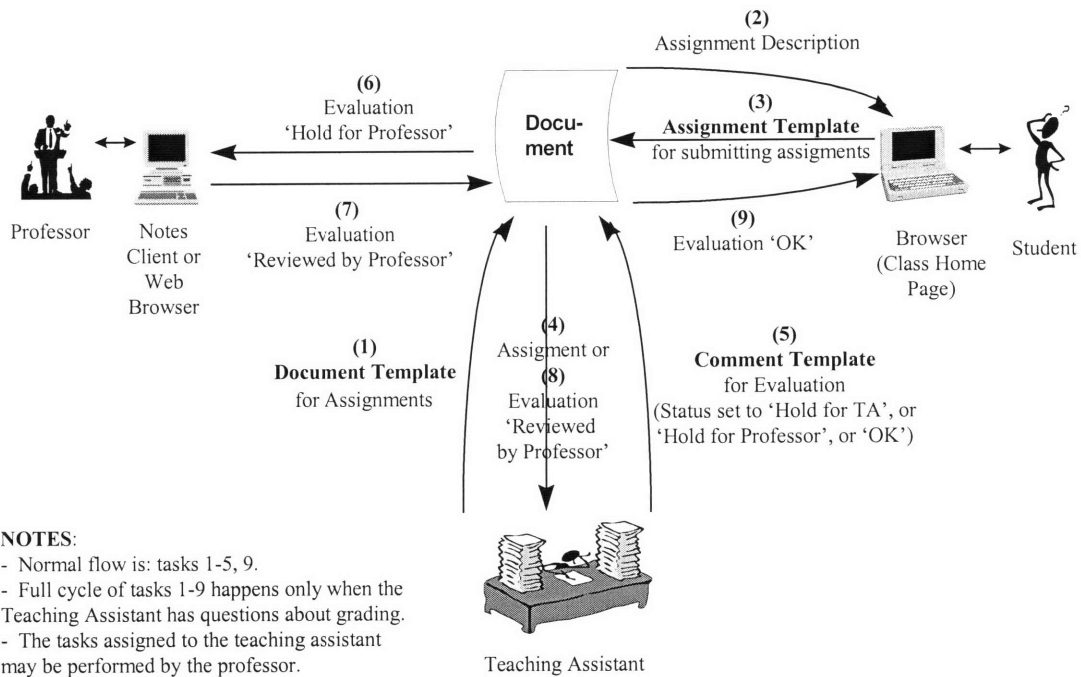


Figure 7.6.1-2 Workflow Showing Databases, Templates, and Documents for Managing Assignments

7.7 Video Streaming

Even though video streaming will not become a major part of COMMAND until the second phase of this project, most of the ground work will be done in the first phase.

Video delivery will be done mostly in the way of previously encoded help files and sample lectures. Through out phase one of COMMAND several instructional videos have been made to help students learn about the features of COMMAND. All of these videos have been placed on the web and are available to anyone with a web browser.

Sample lectures will also be created for testing purposes. The lectures made during the first phase will not be made open to the public. Experimentation with light settings, backgrounds, encoding formats, and streaming technologies will be done in the discovery process.

Lecture encoding and delivery to students will not be made until the second phase. For more information on this subject see chapter4.

CHAPTER 8

Closing Remarks and Future Work

Future work on COMMAND can be classified as follows: 1) additional features to the system, 2) additional applications of the current information model, and 3) improving the remote student experience.

8.1 Closing Remarks

Phase I of COMMAND was released January 1, 1998. Over 30 courses were requested from the schools of Management, Engineering, Science, and Architecture (for a complete listing of courses see appendix D)

The mass customization, and central management model, met the needs of classes across diverse disciplines and students at a distance. Also, designing for institutionalization made it possible for two persons to support all courses and close to 1000 users.

Professors maintained publishing control while the system shield them from complexity. Faculty feedback, indicated that ease of use, in the course web site creation, contributed to the high volume of courses requested. Also based on course staff inquiries the volume of courses is expected to double for the second phase.

8.2 Application of Information Model

The model that was constructed for COMMAND is very flexible and can be readily applied to other information intensive tasks. A possible application is laboratory research groups.

Discussion entries can be used as a history of experiments, documentation, and to bring new

research assistants up to date. The site directory can list everyone involved with a research project. Also, the ability to attach files can be used to collaborate with remote research participants.

An application, similar to the one described, has been made for the virtual engineering initiative (VEI). This project is a joint effort between MIT, Ford, Daimler-Benz, Motorola, and Siemens to build the car of the future. Since the participants are in different parts of the country and different continents, face to face meetings are difficult. Thus, the projects database has worked well as a central information bank that holds contact information for all participants' project history, attached files, and correspondence.

An advantage of having all this information in a database is that it can easily sorted and manipulated. For example, if at any point it became convenient to sort participants by company this can readily done. Even obscure data manipulations, such as how many files have been attached by a German member on Tuesdays can be done with the same ease.

8.3 Additional Features

8.3.1 Compiling online

Compiling on line is something the author wanted to develop, but could not, due to time constraints. For the remote student, the ability to write computer code and compile it online can be a great asset. The student no longer has to worry about installing a compiler or looking for a computer that has one. An online compiler will permit students to program anywhere there is a browser.

For additional features please see chapter four, “New Features”.

8.4 Improving the Remote Student Experience

Currently there are many research groups trying to find effective ways of delivering education at a distance. One of the toughest challenges facing all groups is motivating the remote student as well as building interpersonal communication.

Many distance education students feel disconnected and soon lose their motivation. Trying to overcome this problem, some programs have had success by personalizing their sites. They have included short bios of people working on the site and have let the site itself reflect their personalities. Over the length of the course, students come to view the content providers as real people and not just machines.

Another option may be to provide real time validation of the user. This may come in the form of taking questions from remote users or polling remote students for their opinion while the class in session.

Thus, suggested approaches to improve remote user’s experience are:

- Strive for a site tone that is inviting and not impersonal
- Real time user validation

APPENDIX A

Available Information Technology

This appendix introduces available information technology and possible configurations. It goes on to discuss several models to access database engines through the web. It also presents current tools for course delivery

A.1 Web Servers

The leading web servers at the time this paper is being written are:

- The Apache Group: Apache HTTP Server
- Sun Microsystems Inc: Java Web Server
- Lotus Development Corp: Lotus Domino Webserver
- Microsoft Corp: Microsoft Internet Information Server (IIS)
- Netscape Communications Corp: Netscape Enterprise Server
- Novell Inc: Novell Netscape Enterprise
- O'Reilly & Associates Inc: WebSite Professional
- StarNine Technologies Inc: WebStar
- Tenor Intersystems: WebTen

The market share that each company has on the web differs dramatically along two major categories: intranets and public web sites. For example, consider the Apache web server which holds almost 50 percent of the market in public web sites while only eight percent in intranets⁴.

Following is a list of the features that are common to most of the web servers listed above:

⁴ Netcraft Ltd, 2/98, and Zona Research Survey, 9/97

- Access allowed for both major browsers (Internet Explorer and Netscape Communicator)
- Scalable to multiple processors
- Proxy server
- Built in indexing
- FTP
- Remote management
- Logging, Statistics
- Web authentication, digital certificate authentication, SSL 3.0, X.509

Most servers can be setup for database connectivity. The caveat is that each is optimized for a particular database. That is not to say that they would not work with other databases but the configuration could be very messy and results not optimal. Operating system dependence is another characteristic that differs across servers.

A.2 Databases

There are many databases that can be tied in to the web. Some of the databases that corporations have built their information infrastructure on:

- Borland IntraBuilder
- Microsoft Visual InterDev
- Netscape Visual JavaScript
- Oracle Web Developer Suite
- Sybase PowerBuilder Enterprise
- Apple WebObjects Enterprise
- IBM VisualAge
- NetDynamics, NetDynamics 4.0

All of these applications permit a web application to access a legacy client/server database application. They will also fit into creating a web site and content from scratch.

A.3 Database – Web Server

There are several models that can be used to link a database to the web. The most common will be discussed with their pros and cons in what follows.

A.3.1 Web Server, Database Server, with Plugins and Native ODBC

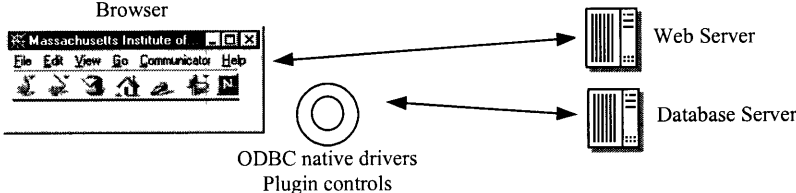


Figure A-1 Database - Web Server, Plugins and ODBC

Using plugins such as Visual Basic and ActiveX the client can access the database directly. A drawback to this setup is that the interaction with the database takes place on the client side. The result is a very heavy client, which is not a problem in the intranet but is impractical outside of it.

This structure is very attractive to many companies that already have a web server and a database in house. Many companies already have a web server and a database server working successfully independently of each other. For this companies linking the two requires minimal effort. An added bonus is not having to alter the company's internal information infrastructure. However, the client, due to heaviness, will only be practical in local area networks (LAN) and intranets.

A.3.2 ActiveX and Java

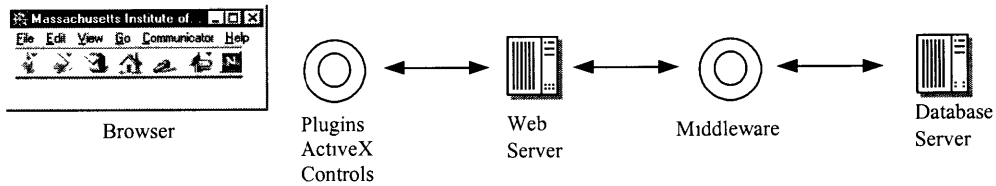


Figure A-2 Database - Web Server, ActiveX and Java

Another option is using ActiveX and Java to build forms for display in HTML. In this configuration no database drivers are needed but depending on the browser not all features (or any) will work. Also, Java is often painfully slow to download to the client. Thus, the speed problem remains.

The operation often takes the following form; as the browser requests pages the web server can retrieve the data using Java or ActiveX from the database and serve them to the client.

A.3.3 HTML only

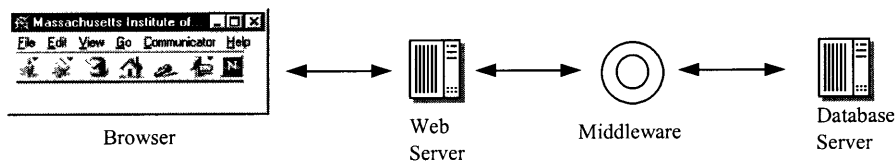


Figure A-3 Database -Web Server, HTML Only

A pure HTML model, on the client side, is likely to be the best model. This approach creates a very thin client where pages are formed using HTML or JavaScript.

In this configuration the server manages the entire interaction with the database server. The protocol to use between the web server and the databases can be CGI, Java, JavaScript, or Microsoft's Active Server Pages (ASP).

A.3.4 Overview

All things being equal, HTML only, thin clients, are the best configuration. Clients, intranet or other, need only a browser to access the database. Corporate can access from anywhere in the world; they need not be confined to the office in order to access the database. The database server can be upgraded with out the client having to be involved.

A.4 Web Browsers

The current leading browsers are Microsoft's Internet Explorer and Netscape's Communicator. The new features of these browsers are supported by hypertext markup language (HTML) 4.0.

At the time this project was being planned MIT's network system, Athena⁵, was using and supporting Netscape Navigator 3.1 for their UNIX system. Since the majority of users would be on-campus students, the development team choose to support the same browser.

A.4.1 HTML 3.2

The version of HTML, as defined by the World Wide Web Consortium (W3C), is closely related to the browser version. For example, Netscape's version 3.x supported HTML 3.2. This means that when the team choose Navigator it limited it self to HTML 3.2.

⁵ MIT's campus-wide academic computing facility

This limitation is not a necessarily a shortcoming in the system. The new features of DHTML are very appealing: layers of content, cascading style sheets, and animation. However, the problem in most cases is that the supporting code is not browser independent, also the load this pages puts on a machine are quite high, and in many occasions the browser becomes unstable and crashes. Thus, these new features are great for demos but it will take six months to a year, in the author's opinion, before they become stable enough for practical applications.

A.5 Real Time GroupWare Tools

Currently there are many real time tools for groupware on the market, among them are:

- Lotus Instant! Teamroom
- Netscape Virtual Office by Concentric
- HotOffice Virtual Office Service
- Netopia Virtual Office 2.0
- 3-2-1 Intranet!

However, there may be no need to look further than one's browser. Both Microsoft and Netscape browsers have built in collaboration tools.

Microsoft NetMeeting offers a whiteboard, chat, file transfer, video conferencing, and application sharing, while Netscape Conference 4.0 includes a whiteboard, chat, file transfer, and collaborative browsing to the mix.

For the first phase of COMMAND the browser's communication and collaboration tools will be sufficient for the students to collaborate. A great advantage of choosing browsers for collaboration is that they are free and most people already have them.

A.6 Video Streaming

In streaming video, the video player on the user's desktop continually requests video data from a server. The user does not have to wait for the entire file to download. Instead the video is continually sent in a stream.

Streaming video compresses video using a codec (Coder/DECoder). Coders shrink video by using mathematical algorithms such as wavelet, fractal, and discrete cosine transform (DCT). Decoders interpret this data and play the video.

As bandwidth decreases streaming techniques scale the amount of video data transmitted. A simple technique is stream thinning where frames are dropped. Wavelet encoding on the other hand will maintain the frame rate but send less detail.

Companies offering codecs are:

- Cubic VideoComm Inc, CvideoNow
- Microsoft Corp, Microsoft NetShow
- Progressive Networks Inc, Real Video
- Xing Technology Inc, Stream Works
- VDOnet Corp, VDOLive
- Vivo Active Software Inc, VivoActive
- Vosnic LLC, Vosiac MediaSuite

A.7 Security

Only two options were considered: secure socket layer (SSL) and Kerberos. SSL, originally developed by Netscape Communications Corp, is a security protocol layered between application protocols and TCP/IP. SSL works by using a private key to encrypt data that's transferred over the SSL connection. The encryption uses RSA public key cryptography from RSA Data Security Inc. Both Netscape Navigator and Internet Explorer support SSL. For a more thorough discussion of SSL please see chapter 4.

Kerberos is a security system developed at the Massachusetts Institute of Technology (MIT). Kerberos is a network authentication system for physically insecure networks. It works by assigning a unique key, called a ticket, to each user that logs on to the network. One of its best features is that users do not send passwords over the network. For an in-depth discussion of Kerberos please see chapter 4.

A.8 Lotus Notes

As the reader may have concluded there are many ways of creating a web site. There is a rich variety of configurations and features to choose from. There are applications that will fit every need or conform to most established information infrastructures. Why use Lotus Notes?

This project was not the first effort into web based course delivery by the author. Over the past three years the author has worked on internet development and over the last year and a half on course delivery. Over this time the author has had the opportunity to become familiar with most of the tools in this area.

During this time the product proved to be robust and scalable. Also, the feedback from students and system administrators was a positive one.

During this time Lotus Development Corp. developed a friendly relationship with the researchers. They provided free software, accepted internship students, and displayed interest in collaborating further with MIT.

What follows is an outline of Lotus Notes and Domino features.

A.8.1 Typical Database File

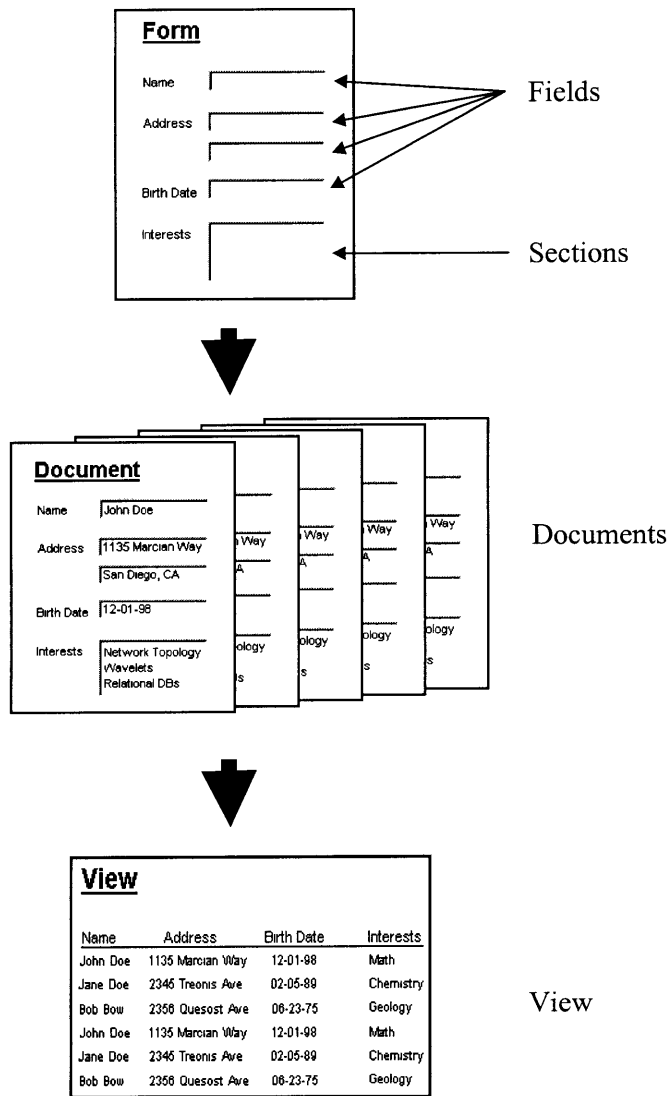


Figure A-4 Notes Database

Forms are the interface that is used to enter information to a database file. The components of forms are fields (blanks to be filled) and sections. These can be arranged in any manner to meet the developers needs.

Once a form has been created, every time a user fills in form a document is created. For example, a form, which holds the contact information for a user, will be completed by every member of an organization. This in turn will create a separate document for every member.

In order for documents to be viewed they must be arranged in views. Views are very flexible and they can display information from one or many forms. In the previous example, the documents for every member would be listed, see the previous figure.

All of the forms, documents, and views are stored in a single file called a database file.

Regardless of how many records the database comes to have, there will be only one file.

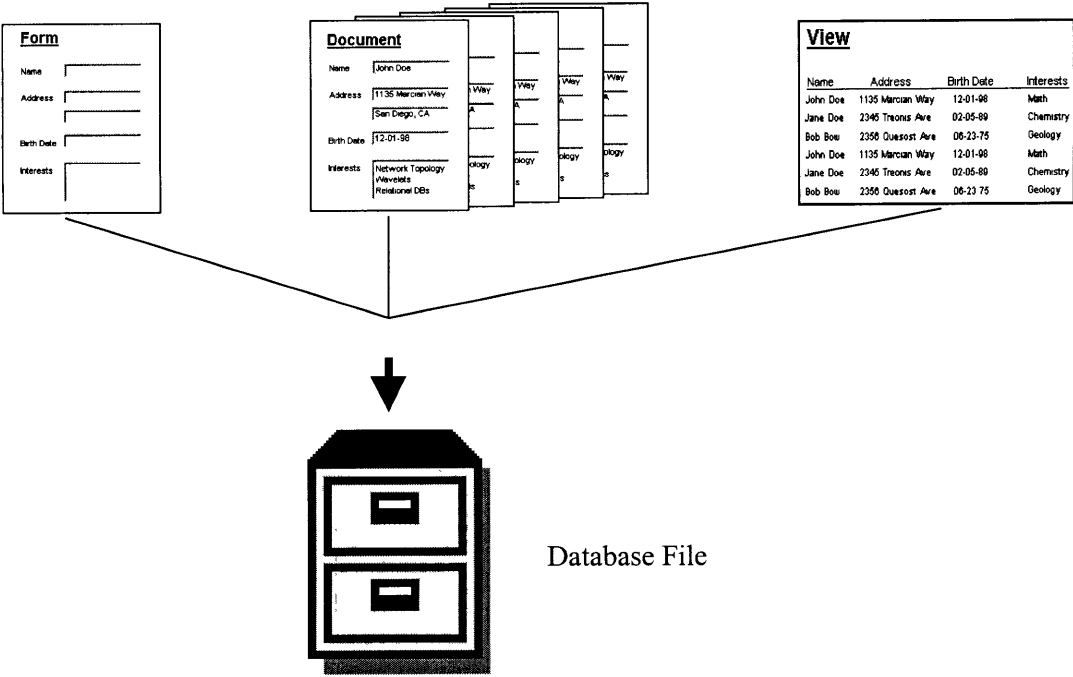


Figure A-5 Notes Database Structure

A.8.2 Domino Architecture

Lotus Notes and Domino present an HTML only relationship between the browser, web server, and database.

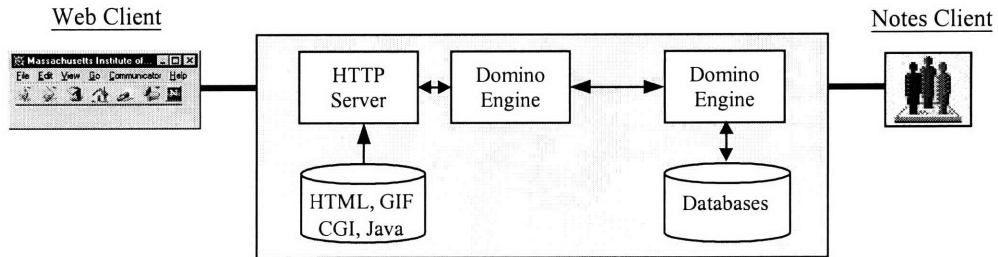


Figure A-6 Domino Architecture

A.8.3 Notes Security

Lotus Notes is a groupware system used by many corporations, for this reason and others it has a rich array of security features. Conceptually, the levels of security can be represented by a funnel, where the lower the level, the higher the security (see below).

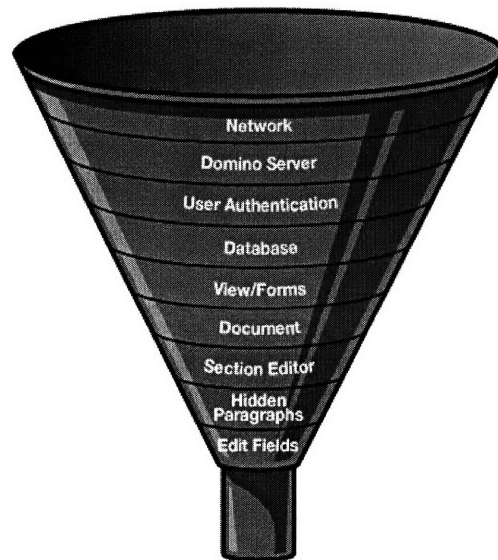


Figure A-7 Security Model⁶

⁶ Figure from "Designing a Secure Application", Howard Greenberg, Iris Associates Inc, 1997

A.8.3.1 Network

The first line of defense is access to the network. Easy access to a network will promote collaboration but will also invite hackers.

For highly sensitive information the best solution is to implement a firewall. However, for course delivery this is impractical.

A.8.3.2 Domino Server

The domino server has the ability to refuse any unwelcome request. This is achieved by denying anonymous access. This means that anyone that wishes to retrieve information must first prove that he is a system user.

A.8.3.3 User Authentication

In addition to requiring user access for the server, it is also possible to request user authentication for individual databases.

As part of the authentication procedure an SSL connection and encryption of the information exchange can be initiated. For a more thorough discussion on SSL please see chapter 4.

A.8.3.4 Levels of Access to a Database

The access control list (ACL) defines 7 levels of access:

1. No Access
 - Access to the database is denied
2. Depositor
 - The user can only deposit new documents
3. Reader

- Reader access
4. Author
 - Can author documents
 - Can read documents created by author
 - Can read other documents as defined by the designer
 5. Designer
 - Author
 - Can read any document
 - Can change database design
 6. Manager
 - Designer
 - Can change ACL
 - Can delete database

A.8.3.5 Views and Forms

In addition to the restrictions mentioned it is also possible to restrict access to individual views and forms.

A.8.3.6 Documents, Sections, Fields

A similar technique to the one used to deny access to a database, view, or form can be used to restrict access to documents sections or fields.

A.8.3.7 Relevant Features to the Course Delivery System

At a high level the most important features of the Lotus notes application were:

- Reliability
 - a) Proven history at MIT
 - b) Multi-threaded routing and SMP support
 - c) Clustering, partitioning
- Scalability
 - a) Several thousand users per server
 - b) Centralized administration
 - c) Replicated directory
- Legacy Integration
 - a) Integration with MS Windows NT
 - i. Directory synchronization
 - ii. Single login for NT and Notes Domain
 - iii. Password synchronization
 - iv. Single user creation for both systems
- Security
 - a) See previous section
- Internet Standards
 - a) SMTP/MIME
 - b) HTTP,HTML,NNTP
 - c) POP3,IMAP,LDAP
 - d) SSL,SMIME

At a lower level the important factors were:

- Strong Points:
 - a) Dynamic Translation to HTML and self management
 - b) Developer can write code to extend HTML. Thus, when a user enters data the system can answer (run) with a series of processes.
 - c) 100,000+ Documents per database
 - d) User Friendly development platform

- Why are they important:
 - a) In order to scale up with out increasing staff a dynamic translation to HTML was needed. This way users can interact with the system in plain text and the Domino Server can take care of the translation. In other words, once the logic is in place the administrator can “forget” about the database (web site) while users continue to interact with the database and add content (discussion entries, turn in homework, registration entries, post lecture notes, etc)
 - b) The code behind fields will allow users to search, upload documents, and manipulate the data on the system.
 - c) By the end of a term a course has thousand of documents. Thus, the management system must be robust or chaos will incur.
 - d) Notes offers a friendly WYSIWYG interface. The low-level aspects must still be coded but high level components can be created quickly.

APPENDIX B

B.1 Specifications for COMMAND Phase I⁷

The Command (Course Management and Delivery) System is intended to facilitate the delivery of course materials over the WWW. Course participants may use the system to collaborate with each other and/or communicate with the instructor. Its design seeks to incorporate the needs of the administrator, the student and the instructor.

Command will be optimized for Netscape 3.0 and higher. Command does not seek to deliver course content in any way, therefore it is up to the individual instructors to set standards regarding presentation software (i.e. PowerPoint/Freelance Graphics/MS Word versions, etc.) Should the instructor wish to post a file which has a viewer, he/she should instruct the students on how to download and install the viewer for their browser. Command accepts and delivers all file types.

The following is a list of the features which will be included in Phase I of the development cycle, to be released in January '98.

Command System Home

This area is the main page for all users. It includes links to the Administrative ,Course Listing, Contact Information and Course Selection area, Student Folder, Hot Topics and Demo Course

⁷ Written by Kimberly Ryan, Graduate Student, MIT, October 1998

area. It also features a log-on button where if one should use to subsequently log-on as an administrator, icons for the system log area is available as well.

Administrative Area

This area contains a workflow enabled form for the instructor to request that a course include an on-line component, it is here that he/she selects the course template features desired. Upon submittal, the administrator finds the request in the "Pending" area, and creates the databases using the appropriate templates. The naming convention is to place each course in a directory called by its coursename. (The administrator also needs to set-up a redirect so that the URL will look like <http://server.mit.edu/coursename> and not <http://server.mit.edu/coursename/home.nsf?open>). The administrator then marks the request as processed which triggers an e-mail to the instructor (and TA (s) if listed) . The e-mail indicates that the course templates have been set-up and are now ready to be populated. During Phase I, a course setup request will be manually verified by the administrator as coming from a valid source (i.e. the course instructor or TA). Additionally, during Phase I, the instructor and listed TA's for the course will be placed into a group with the convention CourseNumberADMIN. This group will be manually created and added to all the databases of the particular course. This database shall consist of one request form, one e-mail agent and two views (pending and processed).

Course Listing

The course listing area is the same database as the administrative area, however it is a peek into the "Processed" view. It is simply a listing of all courses which are available on-line, with a link to their "Home" area. The document itself, includes a link which has been computed to the registrar's course description. The link to the description is automatically generated via a formula which follows the registrar's convention.

Contact Information and Course Selection

This area is available for edit at any time during the school year. It is here where all system users (including instructors and TA's) begin. Users may add/delete contact information, change their password, and select courses. During Phase II, the Command system will incorporate the use of MIT's Kerberos certificates in order to securely authenticate users.

Each semester, an agent will be run on all documents to archive historical data on course selection and to leave the document in a clean state. This agent will also archive the courses which the student was registered for during the term. This database shall include one form, once archiving agents and several system views. The system views will be coordinated with the course template areas so that a class list may be generated.

Hot Topics

This area is where the system administrator posts important system wide information. It is a stand alone database, which contains an archive of all system messages, and a view which displays only the last 5 messages. These messages may or may not be links, based upon the information they contain. It is wizard-like in the sense that the administrator fills out a form, which includes a question about whether or not the document is a link. Once in the view, the document is automatically "clickable" or not. From the Command System homepage, there is a computed graphical display, which links to this area only if a new message has been posted in the last 5 days. This database consists of one input form, and two views (current and archived messages)

Student Folder

Initially the student folder will simply be a link to a computed page, which includes links to the courses the student is registered for. Ultimately (sometime post Phase I) the student folder will list things such as the daily schedule, cross-course assignments due, etc. This areas will be a computed page contained within the System Home area database. The computations will cross reference the contact information database with standard views in the course template databases for the courses the student for which is registered.

System Log

By capturing CGI variables using LotusScript agents, we capture information about system usage. We are able to tell where a user entered the system, how they moved through the system and further how long they spent on each page which we track. During Phase I, the agent and the logging database will be created, when it is determined which types of reports are required, the appropriate views will be built to display that information. This database is for administrators only.

Every document/form/view which we want to track must contain the logging subform, and each database must contain the logging agent. These will be a logging database where information for each tracked page is contained.

Course Templates

The course templates area is divided into many pieces. There is one template for each possible permutation of features, so that the administrator may generate the databases without any custom development. Ultimately, each course will contain at most two databases with all of the features. The databases are divided according to archiving requirements.

Documents

This database is where the instructor posts syllabus, lecture notes, etc. Each term, the instructor may choose to move all of the documents to the "background" and then on an individual basis move them back into the foreground where they are available to the students. This will be achieved by setting a flag field on the "course document" form and then placing the document in the view based upon the flag.

Discussion

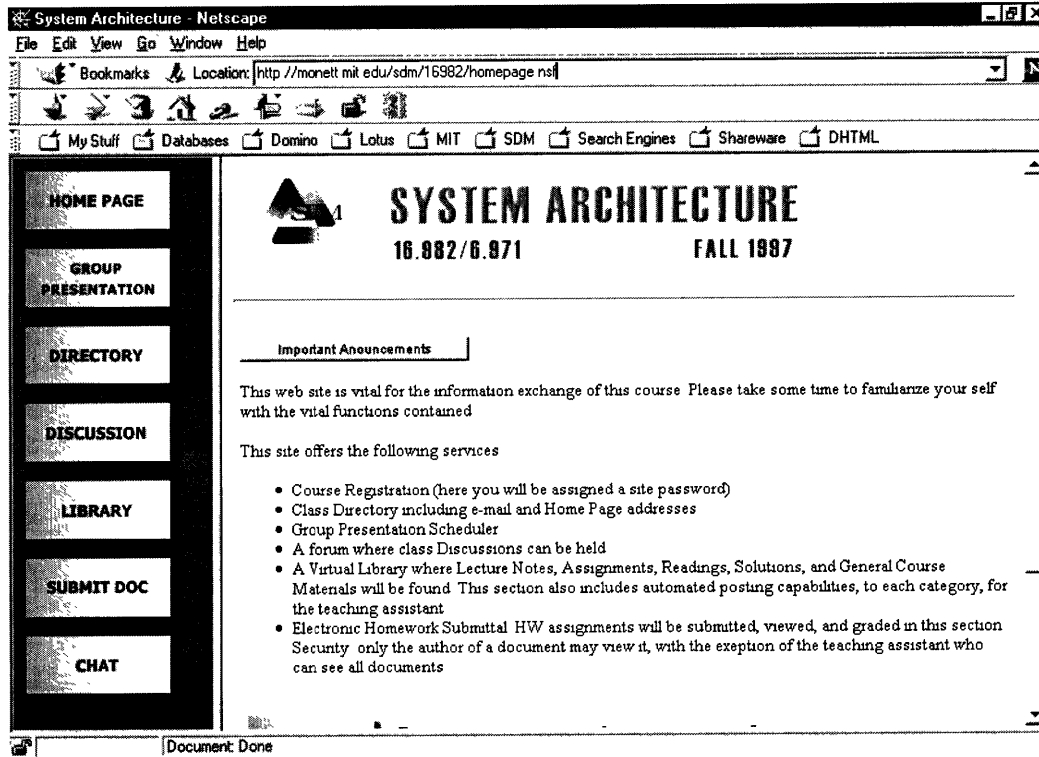
This area will be modeled after the standard discussion template that comes with Domino 4.6. It will also feature a ReaderNames field, which will allow the author of the main document to select from the list of students in the class. All responses will inherit this information, thus allowing for "private" discussions within the database. The list of students must be computed from a system view in the contact information/course selection area database.

Homework Upload & Grading

This area will incorporate Reader Names fields, so that students can only see their own submission and grades. It will also include views which display the grades for each homework, so that the instructor can easily export the information and import it into a grading roster. It includes timestamp information to ensure timely submission of homework.

APPENDIX C

C.1 Systems Architecture – 16.982/6.971



C.2 System and Project Management – 15.965

System & Project Management - Netscape

File Edit View Go Window Help

Bookmarks Location: http://monell.mit.edu/sdm/15965/homepage.nsf

My Stuff Databases Domino Lotus MIT SDM Search Engines Shareware DHTML

HOME PAGE

GROUP PRESENTATION


DIRECTORY

DISCUSSION

LIBRARY

SUBMIT DOC

CHAT

 **SYSTEM & PROJECT MANAGEMENT**
15.965 FALL 1997

Important Announcements

This web site is vital for the information exchange of this course. Please take some time to familiarize your self with the vital functions contained.

This site offers the following services:

- Course Registration (here you will be assigned a site password)
- Class Directory including e-mail and Home Page addresses
- Group Presentation Scheduler
- A forum where class Discussions can be held.
- A Virtual Library where Lecture Notes, Assignments, Readings, Solutions, and General Course Materials will be found. This section also includes automated posting capabilities, to each category, for the teaching assistant
- Electronic Homework Submittal HW assignments will be submitted, viewed, and graded in this section. Security: only the author of a document may view it, with the exception of the teaching assistant who can see all documents

Document: Done

C.3 Computer Aided Engineering – 1.124

1.124 Home Page - Netscape

File Edit View Go Window Help

Bookmarks Location: nst/5d793982ffc27d38525650000757e53/0907e402045d445852565010014089a?OpenDocument

My Stuff Databases Domino Lotus MIT SDM Search Engines Shareware DHTML

HOME PAGE

GROUP PRESENTATION


DIRECTORY

DISCUSSION

PRIVATE DISCUSSION

LIBRARY

SCHEDULE

 **COMPUTER AIDED ENGINEERING I**
1.124 FALL 1997

Important Announcements

Lecturers:
Prof [John R. Wilkams](mailto:john@iesl.mit.edu), 1-250, john@iesl.mit.edu, Office Hours: TU after class
Prof [Kevin Amaratunga](mailto:kevin@mit.edu), 1-274, kevin@mit.edu, Office Hours: TH 2-3:30, Rm 1-131
[T. A. Petros Komodromos](mailto:petros@mit.edu), 1-245, petros@mit.edu, Office Hours: W 3-5:00, Room 1-131
Lecture Tues, Thurs 9:00-10:30 A.M. Room 1-390, **Recitation** Fri 9:00-10:00 A.M., Room 1-190

[Goals and Content](#) [Assignments](#)

[Java Tutorial](#) [Solutions](#)

Application Design & Webmaster [Abel Sanchez](#), MIT-IESL

This page has been accessed 9976 times since 09/01/97

Document: Done

APPENDIX D

D.1 List of Courses on COMMAND



▼ Course 01 - Civil and Environmental Engineering

1 201	<u>Transportation Systems</u>	Instructor: Joseph Sussman TA: Christopher Conklin
1 441	<u>Public Infrastructure Development System</u>	Instructor: John B. Miller TA: Unknown
1 44	<u>Law and the Construction Industry</u>	Instructor: John B. Miller TA: unknown
1 231J	<u>Planning and Design of Airport Systems</u>	Instructor: Richard de Neufville TA: Jay Falker
1 130	<u>Wavelets and filter banks</u>	Instructor: Kevin Amaratunga TA: unknown
1 206	<u>Airline Operations Research</u>	Instructor: Cynthia Barnhart TA: Amy Manville Cohn
1.77	<u>Water Quality Control</u>	Instructor: Eric Adams TA: Ling Tang
1 432	<u>Project Control (Revised Content and Un</u>	Instructor: Femosky Pena-Mora TA: nobody
1 202	<u>Demand Modeling</u>	Instructor: Moshe Ben-Akiva TA: Scott Rammung
1 51	<u>Design of Steel Structure</u>	Instructor: Shi-Chang Wooh TA: donotknow
1.120	<u>Information Technology M Eng Project</u>	Instructor: Femosky Pena-Mora TA: idonotknow
1.204	<u>Computer Algorithms in Transportation</u>	Instructor: chabun TA: NA
1 691	<u>Surface Wave Dynamics</u>	Instructor: Chiang Mei TA: Chuang Mei
1 999	<u>Theme Seminar</u>	Instructor: Salvatore Di Bernard TA: Salvatore Di Bernardo

▼ Course 02 - Mechanical Engineering

2 882	<u>Principles of Axiomatic Design</u>	Instructor: Nam Suh TA: Yodhdhot Sanyal
2 739J	<u>Product Design and Development</u>	Instructor: Dan Whitney TA: Unknown



COMMAND Courses

▼ Course 04 - Architecture

Thresholds	<u>thresholds</u>	Instructor: mark jazombek TA: james o'brien
4.189	<u>Preparation for MArch Thesis</u>	Instructor: Hasan Khan TA: Bjoy Ramachandran
4.156	<u>Architectural Design Level III</u>	Instructor: Michael Dennis TA: Darren Brathwaite
4.401	<u>Introduction to Building Technology</u>	Instructor: Vince Cammalleri TA: Acharawan Chutarat
4.361	<u>Dimensions of the Body</u>	Instructor: Ed Levine TA: Michael Rakowitz
4.199	<u>Charrette Ford/MIT Project</u>	Instructor: William Mitchell TA: Susan Yee
4.195	<u>Designing Electronic Commerce and Online</u>	Instructor: Dan Greenwood TA: Ewan Branda

▼ Course 06 - Electrical Engineering and Computer Science

6.046J	<u>Introduction to Algorithms</u>	Instructor: Charles E. Leiserson TA: Harald Prokop
6.011	<u>Communication, Control, Signal Processing</u>	Instructor: George Verghese TA: Chns Hadjicostas

▼ Course 15 - Management

15.311	<u>Organizational Process</u>	Instructor: Thomas Kochan TA: Loan Ngo
15.840	<u>Marketing Management</u>	Instructor: Melvyn Menezes TA: Merritt Maxam
15.057	<u>systems optimization</u>	Instructor: Jim Orlin TA: joel sokol, Yi Wang

▼ Course 16 - Aeronautics and Astronautics

16.982b	<u>System Architecture</u>	Instructor: Crawley TA: Scott Rasmussen
---------	--	--

▼ Course 18 - Mathematics

18.337	<u>Parallel Scientific Computing</u>	Instructor: Alan Edelman TA: Parry Husbands
--------	--	--

▼ MAS - Media Arts and Sciences

MAS964	<u>Interdesign Workshop</u>	Instructor: Hiroshi Ishii, John Maeda TA: Paul Yarin, Tom White
--------	---	--

Bibliography

- [1] Steiner, J, Neuman, C, Schiller, J "Kerberos: An Authentication Service for Open Network Systems" *USENIX Mar 1988*
- [2] MIT Bulletin. *Courses and Degree Programs Issue 96/97.*
- [3] "Lotus Notes Programmers Guide", Lotus Development Corporation, Cambridge, MA 1995
- [4] Doyle C., Stuple S. *Microsoft Windows NT* Microsoft Press, Redmond, Washington 1996
- [5] Brackett, G. "Course Notes - Spring 1998" T525 - *Designing Educational Experiences Using Networks and Webs* Harvard Graduate School of Education, Cambridge, MA 1998
- [6] Hidalgo, C. *Educational uses of the Web : extending a teacher's communication and mediation capabilities through the Internet* Thesis (M.S.), Massachusetts Institute of Technology, Dept. of Civil and Environmental Engineering, 1997
- [7] Best, M. "Electronic Mail" *Program Administrator and Corporate Liaison* Systems Design and Mangement, April 29, 1998
- [8] McWilliams, B. "PC World News Radio" *Can Government Stop Hackers?* PC WORLD online, March 2, 1998. URL: <http://www.peworld.com>
- [9] Calabria J., Kirkland R., *Professional Developers Guide to Domino* Que Corporation, Indianapolis, IN 1997