1

PM-1 3 1/2"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

PRECISIONSM RESOLUTION TARGETS

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Qualité de votre document

Qualité de votre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

MULTINODES - A Multimedia News-on-Demand System

by

Benjamin W. Falchuk, B.Math

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Science

Ottawa-Carleton Institute for Computer Science

School of Computer Science

Carleton University
Ottawa, Ontario
August 1, 1995

© Copyright
1995, Benjamin W. Falchuk

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file / Votre référence

Your file / Votre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-612-09003-5

Canada

Name Kenneth F. L. Hunt

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

Computer Science

SUBJECT TERM

0984

SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION

General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0777
Guidance and Counseling 0319
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523

Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language 0679
General 0289
Ancient 0250
Linguistics 0291
Modern 0291
Literature 0401
General 0294
Classical 0295
Comparative 0297
Medieval 0298
Modern 0316
African 0591
American 0305
Asian 0352
Canadian (English) 0355
Canadian (French) 0593
English 0311
Germanic 0312
Latin American 0315
Middle Eastern 0313
Romance 0314
Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
Religion 0318
General 0321
Biblical Studies 0319
Clergy 0320
History of 0322
Philosophy of 0469
Theology 0323

SOCIAL SCIENCES

American Studies 0323
Anthropology 0324
Archaeology 0326
Cultural 0327
Physical 0310
Business Administration 0272
General 0770
Accounting 0454
Banking 0338
Management 0385
Marketing 0501
Canadian Studies 0503
Economics 0505
General 0508
Agricultural 0509
Commerce Business 0510
Finance 0511
History 0358
Labor 0366
Theory 0351
Folklore 0578
Geography 0578
Gerontology 0578
History 0578
General 0578

Ancient 0579
Medieval 0581
Modern 0582
Black 0328
African 0331
Asia, Australia and Oceania 0332
Canadian 0334
European 0335
Latin American 0336
Middle Eastern 0333
United States 0337
History of Science 0585
Law 0398
Political Science 0615
General 0616
International Law and Relations 0617
Public Administration 0814
Recreation 0814
Social Work 0452
Sociology 0626
General 0627
Criminology and Penology 0938
Demography 0631
Ethnic and Racial Studies 0628
Individual and Family Studies 0629
Industrial and Labor Relations 0630
Public and Social Welfare 0700
Social Structure and Development 0709
Theory and Methods 0999
Transportation 0453
Urban and Regional Planning 0453
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture 0473
General 0285
Agronomy 0475
Animal Culture and Nutrition 0476
Animal Pathology 0359
Food Science and Technology 0478
Forestry and Wildlife 0479
Plant Culture 0480
Plant Pathology 0817
Plant Physiology 0777
Range Management 0746
Wood Technology 0306
Biology 0287
General 0308
Anatomy 0309
Biostatistics 0379
Botany 0329
Cell 0353
Ecology 0369
Entomology 0793
Genetics 0410
Limnology 0307
Microbiology 0317
Molecular 0416
Neuroscience 0433
Oceanography 0821
Physiology 0778
Radiation 0472
Veterinary Science 0786
Zoology 0760
Biophysics 0760
General 0425
Medical 0996

EARTH SCIENCES

Biogeochemistry 0425
Geochemistry 0996

Ceodasy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0768
Health Sciences 0566
General 0300
Audiology 0992
Chemotherapy 0567
Dentistry 0350
Education 0769
Hospital Management 0758
Human Development 0982
Immunology 0564
Medicine and Surgery 0347
Mental Health 0569
Nursing 0570
Nutrition 0380
Obstetrics and Gynecology 0354
Occupational Health and Therapy 0381
Ophthalmology 0571
Pathology 0419
Pharmacology 0572
Pharmacy 0382
Physical Therapy 0573
Public Health 0574
Radiology 0575
Recreation 0575

Speech Pathology 0460
Toxicology 0383
Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences 0485
Chemistry 0749
General 0486
Agricultural 0487
Analytical 0488
Biochemistry 0738
Inorganic 0490
Nuclear 0491
Organic 0494
Pharmaceutical 0495
Physical 0754
Polymer 0405
Radiation 0605
Mathematics 0986
Physics 0606
General 0608
Acoustics 0748
Astronomy and Astrophysics 0607
Atomic 0798
Electronics and Electricity 0759
Elementary Particles and High Energy 0609
Fluid and Plasma 0610
Molecular 0752
Nuclear 0756
Optics 0611
Radiation 0463
Solid State 0463
Statistics 0346
Applied Sciences 0984

Applied Sciences

Applied Mechanics 0346
Computer Science 0984

Engineering 0537
General 0538
Aerospace 0539
Agricultural 0540
Automotive 0541
Biomedical 0542
Chemical 0543
Civil 0544
Electronics and Electrical 0348
Heat and Thermodynamics 0545
Hydraulic 0546
Industrial 0547
Marine 0794
Materials Science 0548
Mechanical 0743
Metallurgy 0551
Mining 0552
Nuclear 0549
Packaging 0765
Petroleum 0554
Sanitary and Municipal ... 0790
System Science 0428
Geotechnology 0796
Operations Research 0795
Plastics Technology 0994
Textile Technology 0994

PSYCHOLOGY

General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

The undersigned hereby recommend to
The Faculty of Graduate Studies and Research
acceptance of the thesis,
MULTINODES - A Multimedia News-on-Demand System
submitted by
Benjamin W. Falchuk, B.Math
in partial fulfillment of the requirements
for the degree of Master of Science

Director, School of Computer Science

P. Karmouch

Thesis Supervisors

Carleton University
August 1, 1995

Abstract

Currently, maturing networking and computing technologies are making possible a wide range of applications and services that aim to empower users with information and make everyday tasks and chores simpler and more efficient. Applications on broadband networks (and even the Internet) such as home shopping, teleconferencing and video-on-demand are rapidly gaining user acceptance.

In this thesis we discuss MULTINODES, a multimedia news-on-demand system intended to service users on broadband networks, that has been developed in the Multimedia Information Research Laboratory at the University of Ottawa and beyond. The main results of this thesis are: i) the conception and implementation of a news service for the future which makes use of text, graphics, audio and video to present timely news information, and ii) the conception and implementation of an effective tool called video-tiles that allows users to achieve content-based browsing through stored news videos.

MULTINODES represents a number of achievements. It is a new type of news service that incorporates qualities of television and newspapers. In addition, unlike television and newspapers, this system has the characteristics that it is personal, intelligent and powerful while remaining familiar by making use of some standard ways of presenting information such as the *headline/story* model. Although MULTINODES can make use of twisted pair networks like the Internet, it is intended for the fiber networks of the future which will become the backbone of the information superhighway. The system makes an original contribution to the growing field of video representation and browsing with one of its tools called the video-tile video-browser. This tool allows for compact, low-bandwidth, content-based browsing of news videos which play a large role in the system. It uses small icons extracted from the video itself arranged in a hierarchic fashion.

Acknowledgments

I would like to thank my supervisors, Professors Ahmed Karmouch and Ekow Otoo for their motivation and support. Their work ethics and feedback made this thesis a much better piece of work. In particular, I would like to thank Professor Karmouch for opening his laboratory to me and making me feel at home and also for his comments and suggestions which, in the end, have made me a better report writer and researcher.

I would also like to thank the researchers of the Multimedia Information Research Laboratory at the University of Ottawa. Dr. Omar Megzari for his advice, input, and effort into support code; Glenn Warnock for his feedback and software development which saved me some time; and Dr. Bala Brahmanandam for his patience and tutoring when I was still new to the laboratory.

Finally I wish to acknowledge my parents for their constant support through all stages of this degree - without them it would not have been possible.

Contents

Abstract	iii
Acknowledgments	iv
List of Figures	vii
1 Introduction	1
1.1 Traditional and Electronic news	2
1.2 Terminology Related to Multimedia News	4
1.3 Outline	5
2 Current and Emerging Technologies	7
2.1 The World-Wide Web Project	8
2.2 Introduction to Network Protocols	9
2.3 Emerging Multimedia Standards and Forums	12
2.4 The MEDIABASE Project	14
2.4.1 Structuring Multimedia Information	15
2.4.2 Database Architecture	17
2.4.3 Multimedia Storage Subsystem	19
2.4.4 Synchronization Management	19
2.4.4 How MULTINODES Fits into MEDIABASE	21
2.5 Technical Challenges of Multimedia News	21
3 MULTINODES - Feature Set and Architecture	23
3.1 Goals of MULTINODES - Defining News Systems of the Future	25
3.2 User-Oriented Features	26
3.2.1 The Look of the New Medium	27
3.2.2 Continuing Traditional Features - The Follow-Up	29
3.2.3 A Personal Experience	30
3.2.4 An Intelligent Medium	32
3.2.5 News as it Happens	33
3.2.6 Compressing Time with Temporal Proxies	34
3.2.7 Power to the User	35
3.2.8 Encouraging Learning and Interactivity	35
3.2.9 Video Browsing	36
3.2.10 Advertising	36
3.2.11 Interacting with Databases Through Keyword Trees	37
3.3 Author-Oriented Features	39

3.4	Architectural Overview of the System	42
3.4.1	Using the Production Server for MULTINODES	43
3.4.2	Using the Content Server for MULTINODES News Production	45
3.4.3	Using the Multimedia Database for MULTINODES	46
3.4.4	Using the Synchronized Document Architecture for MULTINODES	46
3.4.5	Using Network Communications for MULTINODES	48
3.4.6	Distributing User Sites in MULTINODES	49
3.5	Distinguishing Aspects	49
3.5.1	The Hypertext Transfer Protocol	50
3.5.2	The Nested Server Scheme	50
3.5.3	Database Schema and Other Issues	52
3.6	Functionality and Examples	53
3.6.1	Example 1	54
3.6.2	Example 2	54
3.6.3	Example 3	55
3.7	Related Work	57
3.7.1	Internet and W3	57
3.7.2	Other News Systems	59
3.8	Examination of MULTINODES Architecture	62
4	Representing and Browsing Videos - The Video-tile	63
4.1	The Need for Video Tools	64
4.2	Representation - Visual Indices to Video	66
4.2.1	Related Research	67
4.3	Video Browsing	69
4.3.1	Related Research	70
4.3.2	Determining Camera Operations	71
4.4	A Novel Tool - The Video-tile	72
4.5	Video-tile as a Tool for News Scenario Visualization and as a Generic Tool	82
4.6	Remarks	85
5	Implementation and Results	86
5.1	Sample Multimedia Sessions	87
5.2	Graphical User Interface Construction	94
5.3	User Reactions	96
5.4	Some Test Results	97
5.5	Notes on the Video Browser	99
6	Conclusions	101

6.1 Suggestions for Future Research	102
References	104

List of Figures and Tables

T1	Size and Bitrates required for different Media	7
2.1	The OSI Reference Model	10
2.2	Generic Architecture of the MEDIABASE System	15
2.3	Multimedia Document Model	17
2.4	Multimedia Database Architecture	18
2.5a	Synchronization by Retrieval Algorithm	21
2.5b	Synchronization by Delivery Algorithm	21
3.1	A Follow-Up Initiating a Database Search Resulting in a List of Hits	30
3.2	A Multimedia Profile	31
3.3	A Hypermedia Web	39
3.4	The Basic Logical Structure of the Multimedia Document	40
3.5	The Result of Mapping the Web to the Database	41
3.6	The Generic Architecture of the System	43
3.6	Groupware News Production	44
3.7	A Traditional Timeline and a Portion of a MEDIADOC Document	48
3.8	The Architecture and Protocols of the System	54
3.9	An Author Creating a Hypernews Web on the Database	55
3.10	User-Profiled News- A List of Hits Followed by Article Retrieval	56
3.11	A Search Followed by a Video Browsing Session	57
3.12	Simplified Architecture of the Nynex Dow-Jones System	61
4.1	Accessing Archived Video Streams	66
4.2	A Graph and a Possible Fisheye-type View of the Same Graph	73
4.3	A Video Stream and the Fisheye-type View of the Same Stream	74
4.4	An Insecteye	74
4.5	The Basic Shape of the Video-tile and Temporal Ordering of Sub-Tiles	75
4.6	Browsing with Video-tiles	76
4.7	Initial Tile of a Browsing Session	77
4.8	Focusing on a Part of the Video	78
4.9	Zooming in Further into this Part of the Story	78
4.10	Temporal Structure of a News Story Video	79

4.11	Choosing Frames for the Video Browser	80
4.12	Initial Tile of a Browsing Session	81
4.13	Focusing on a Part of the Video	81
4.14	Zooming in Further into this Part of the Story	82
4.15	A Simple Scenario for a Multimedia News Article	82
4.16	A Video-tile-type layout for a Multimedia News Scenario	83
4.17	Sub-tile Overlays Which Reveal Camera Information	84
5.1	The News Application's Front-end Interface	87
5.2	Creating a Custom Profile	88
5.3a	Example of a News Article from the Hypernews Database	88
5.3b	Example of a News Article from the Hypernews Database	89
5.4	Entering a Search Criterion	90
5.5	A List of Documents Matching the Search Criterion	90
5.6	Viewing the Document which Matched the Search Criterion	91
5.7	Following-Up the Current Document	92
5.8	The Key-word Tree for the Database	93
5.9	Viewing the Contents of a Scrapbook	93
5.10	The News Applications Front-end Interface	95
5.11	The Instance-Hierarchy for the Front-end	96
T2	Bitrates while Transferring media over ATM using TCP/IP	99

Chapter 1

Introduction

Recent developments in network and computer technology have increased optimism that the world, in the future, will be connected by an information superhighway on which many daily activities can be accomplished. These include education, information gathering, voting, shopping and entertainment. It seems inevitable that the phone, television and computer will merge into a multi-purpose machine that considerably empowers its users. Services such as home shopping, banking, video-on-demand, video-telephones and educational programming are already being beta-tested in markets such as Orlando, Florida by Time-Warner among others [5][13]. This is all being accomplished on broadband communication networks which, unlike telephone lines, make possible the kind of data-rates required for most of the above-mentioned services. One of the most important services will certainly be news - that is, some kind of timely electronic replacement *or complement* for the printed newspaper and television news show. It is this news service that we focus on in this thesis. In this chapter, we discuss some pros and cons of electronic news, some basic terminology that will be used throughout the thesis, and provide an outline of the remainder of this thesis, clearly identifying our contributions.

1.1 Traditional and Electronic News

As more and more people use the Internet to retrieve and browse information and broadband services are on the verge of breaking out, it would be interesting, from a news point of view, to know what it is that the *people want* in terms of services. In a recent survey, adults indicated that the top three desired services are: voting in elections, searching reference books and taking courses [9]. News was in ninth place just behind video-on-demand and just in front of searching legal, scientific and medical journals. This relatively low placing of the news application and the variety of responses is a testimony to the challenge of predicting which applications will be in demand and to the competition that will exist amongst these applications.

Electronic news will bring with it many features which empower users more than they could ever be with paper ; for instance, users could perform searches and interact directly with advertisers. However, there are a number of reasons why the traditional newspaper is good and should not be replaced or phased out. Some of these reasons are as follows:

- they are **portable**; they can be read in bed, at the breakfast table or on the train.
- they are **durable**: coffee or crumbs can be spilt on them with little effect.
- they offer **high-density** news viewing; there can be many stories on a single page.
- they are **organized** into sections allowing for fairly easy information retrieval. A newspaper's **physical layout** is well known and effective for browsing.

Certainly as wireless communications become more prevalent the portability issue may be overcome. Several companies, such as Mitel have plans for small, wireless, portable devices which will receive news information from satellites on a constant basis. This is a direct compromise, however, with the news density issue. The smaller the device, the

smaller the viewing screen and hence the smaller the amount of data that can fill that screen. As far as durability, it may be a long time before manufacturers decide to make a computer that is coffee-resistant. The physical layout of the newspaper into independent sections which may be taken and read separately is an advantage that is not impossible to mimic in an electronic system. Good browsing tools and layout can make users almost forget that they are browsing electronic news - also, one can offer browsing capabilities that are not possible with paper. Newspaper layout is so standard that picking up and reading a paper from any other city (of the same language) poses no problem whatever. In particular, the front-page metaphor is adopted by almost all the newspapers in the world and that will not likely change soon since it has been around for over 300 years already [2].

One of the challenges for electronic news systems is to offer enough functionality to overcome the advantages of newspapers. It seems unlikely that newspapers will ever be completely phased out though. There will always be the *have's* and *have-not's* of computer technology and thus a large portion of the community will either not have access to or will not be able to afford the technology that provides electronic news. Newspapers, we feel, are here to stay, though as exciting new media are introduced, more and more people will realize that they lack the dynamism of video and film and thus make for a less effective learning medium [1].

The way that electronic news will differ dramatically from newspapers and television is as follows. Traditional news, in particular television news can be thought of as being directed, or pushed, at the user. If the user is not present to receive it or it is not the particular information that is wanted then the information is wasted or lost on the user. In the new medium of electronic news, the user will grab whatever information he/she wants, when he/she wants it. This makes the medium user-centered. Further, the medium itself will be interactive, unlike television which must be watched passively, allowing the user to drive the information in the direction he/she wishes. This is already the case in some full-blown video-on-demand systems. A distributed electronic news system may involve data

servers from different countries. These servers will not only provide stories relevant to that part of the world but will also serve information in foreign languages. So we see that electronic news can easily be multilingual. In the next section we discuss some terminology that will be used throughout this thesis.

1.2 Terminology Related to Multimedia News

In this section we will introduce some important terminology. Many authors attach different meanings so the following is how we view the definitions of these terms in the context of this thesis:

- **media object**; A unit of data which is homogeneous and is typically considered atomic. It belongs to only one media type - for example it could be an image or a passage of text but not both. Here we do not discuss composite media objects which may consist of both text and image.
- **multimedia**; A general term implying the representing, storing, retrieving and disseminating of information expressed in multiple media such as text, graphics, images, animations, video, and audio.
- **multimedia news or synchronized news**; News presented on a computer screen that has a temporal dimension and some temporal relationships between media objects (i.e. a song plays in the background just as the video ends). By temporal dimension we mean that we think of this type of news as occurring through time in that it has a begin-time and an end-time. This type of news *plays itself back* to the viewer in multiple media according to some temporal script.

- **news document** or **news article**; General terms for a self-contained media object or objects which have some semantic meaning together in that they form a coherent story or convey some piece of information. Documents and articles may also have some logical and/or physical structure which define them; for instance a document or article may have logical structures which partition it into paragraphs, etc.
- **hypertext**; The linking of text objects together via *anchors* and *links*. Anchors are hot-spots embedded in the text which, when clicked upon, traverse a link which points to some other piece of text. A web of textual information can be created and browsed in an arbitrary non-sequential fashion. Hypertext is store-and-forward; that is, data is not retrieved until a user makes some action such as clicking on an anchor at which point the data is retrieved in its entirety. There is no temporal dimension to hypertext.
- **hypermedia**; As hypertext but links may point to media objects of *any* media type.
- **hypernews**; Hypermedia information in the context of news - i.e. the end result of making a hypermedia web of information from newspaper data such as stories and photographs. Hypernews may also include video, audio and animations as media objects - media not typically associated with newspaper data.

These terms are important since MULTINODES encompasses many of these terms and the issues that are associated with them. Multimedia is a relatively new area of communications and computer science and this thesis proposes a news service that uses technology and expertise to create a new type of news medium.

1.4 Outline

The major contributions of this thesis are found in chapters 3 and 4 with implementation details and results given in chapter 5. The main contributions are:

- the conception and feature set of a multimedia news service

- an architecture for this news system which is robust and supported by technological and editorial expertise
- a tool for representing and browsing videos which allows for quick grasping of the video-content and for non-sequential browsing through the stream

In **chapter 2** we provide an overview of the current state of technology; the Internet, the World Wide Web, network protocols and multimedia standards. We then discuss the MEDIABASE project and how this work is related to what has already been accomplished. This provides a good introduction to many areas of multimedia. In **chapter 3** we present MULTINODES, the multimedia news-on-demand system. We describe the conception phase and how and why we were able to make the system powerful, personal, intelligent, innovative, familiar, and user-friendly. The system represents a new type of news service and its feature set and user interface are unlike those in other existing systems. The architecture of MULTINODES takes advantage of broadband networks as well as being compatible with networks such as the Internet. In **chapter 4** we describe the challenges associated with representing and browsing digital video. We introduce an original contribution to the field of video representation and browsing called video-tiles - a tool which allows users to i) form a conceptual image of a video stream without viewing the entire video and ii) perform content-based hierarchic video browsing on news videos. Video-tiles can be generalized to be effective on other types of temporal data. Video-tiles have advantages over other browsers and query languages since it is graphical and content-based. This maintains a good mapping - a visual browser for a visual medium. Also, video-tiles is effective since it is hierarchic and at all times provides global context while displaying local detail. In **chapter 5** we give some implementation details and results and show sample sessions with our system. We also provide some code samples and algorithms involved. In **chapter 6** we conclude and suggest directions for future research.

Chapter 2

Current and Emerging Technologies

Currently, although technological advances are being made faster and faster and it is possible to achieve network bitrates of up to 155 Megabits per second (Mbps), the fact that these networks are still expensive and not widely available means that the common user is typically *not* sitting in front of a high-speed workstation on a high-speed network. Different types of information have different storage and data-rate requirements. These can be summarized in table 1.[47] (note that Mbps means megabits per second):

	Text	Image	Audio	Animation	Video
Object type	Standard ASCII or EBCDIC encoded text	Bit-mapped graphics, photographs, faxes	Noncoded stream of audio	15-19 frames per second of audio and video	TV image at 24-30 frames per second
Size and Band-width	2 kB per page of text	Undetailed: 64KB/image Detailed: 7.5 MB/image	Voice/phone: 64 kb/s CD quality: 1.4 Mbps	for 320x240x16 pixel/frame: 20 Mbps	for 640x480x24 pixels/frame: 221 Mbps

Table 1. Size and minimum bitrates required for different media

In this chapter we discuss the technology that is the backbone of many current systems. In particular we introduce the Internet World Wide Web project, and compare two basic but important network protocols. We also briefly introduce some of the emerging international standards and forums for multimedia. Finally, we discuss the MEDIABASE project at the University of Ottawa which was the background work which provided a platform for MULTINODES to be built upon.

2.1 The World-Wide Web Project

The World-Wide Web (W3) project is a type of Internet information service. With it, users browse hypermedia documents which allow them to click on highlighted words or images and receive related documents retrieved from somewhere else on the Internet. In effect, the W3 creates an enormous, seamless, distributed document. Born at the European Center for Nuclear Research (CERN) it is a client server system consisting of W3 browsers (clients) and information servers. Browsers and servers can use a number of existing Internet protocols. Typifying the text-based Internet protocols is Gopher. Gopher is a distributed information service that makes hierarchical information available to users on the Internet. Textual information is passed to the client as he/she browses through menu-type choices. The Hypertext Transfer Protocol (HTTP), the native protocol of the W3, can handle many types of data including images, text, video and audio. It is intended mainly for transferring hypertext documents which create links to other resources on the Internet. It absorbs many existing Internet protocols (such as Gopher) so that *existing* information databases and resources are browsable on the W3 [16]. To serve information to the mass of users on the W3 then, one must run a server. To serve hypertext/hypermedia information it is necessary to run the HTTP server. However to serve Gopher data, for example, it is not necessary to run HTTP since W3 browsers

inherently know the Gopher protocol. A W3 hypermedia document has the look of a glossy magazine page. Text of varying sizes and fonts coexist on the page with drawings and photographs. Clickable spots in the document may download a new such page, an audio clip, or a video.

From one point of view W3 is the ultimate in information liberation - the W3 has given the world easy access to (often) beautifully formatted information and has given the common user the ability to make information available to the world. From another point of view W3 has congested the Internet with users who browse thoughtlessly, not unlike television viewers who impulsively change channels, and download movies and images for amusement purposes. Between January and August 1993, the traffic due to the W3 multiplied by a factor of 414 [16]. Also, as there are no rules for information providers, everything from pornography to advertising is being made available.

2.2 Introduction to Network Protocols

In this section we will briefly summarize and compare two basic network communication protocols that come into play in applications such as multimedia news. Whenever information is sent over a network it is sent in the context of some protocol. A protocol is a set of rules or procedures governing communications between computing environments. The International Standards Organization (ISO) has established the Open Systems Interconnection (OSI) reference model. The model has the seven layers shown in figure 2.1. The physical layer co-ordinates the actual transfer the bits of information, the data link layer performs error-checking and groups information into organized units of information called frames, the network layer groups the data into larger units called packets and performs some network routing of the information, the transport layer makes transport links, the session layer establishes the session and parameters, the presentation layer

performs transformation of data, and the application layer is the high-level communication between the participating applications such as e-mail.

Figure 2.1. The OSI reference model

Local Area Networks (LAN's) are commonplace for connecting computers in the same laboratory or building; two LAN protocols are Ethernet and Token Ring. In the Ethernet protocol, a station wanting to send data waits until the network (or bus) is relatively traffic-free and then sends the data. It continues to monitor the bus and if a collision is detected it waits a random amount of time and attempts to retransmit. In the Token Ring protocol, workstations are configured on a unidirectional ring-shaped network. A special bit pattern (a token) circulates around the ring and if a workstation wishes to transmit data it must seize the token first. When the data is transmitted the token is released. LAN protocols can offer an approximate *shared* bitrate in the range of 4-16 Megabits per second (Mbps) but since it is a shared rate the effective throughput may only be a fraction of those. As opposed to a LAN, the Internet network connects computers and networks all around the world and allows business, universities and home-users to exchange information. The transport protocol used by the Internet is called the Transmission Control Protocol/Internet Protocol (TCP/IP) which covers the transport and network layers of the OSI model. Broadband Integrated Services Digital Networks (B-ISDN's) now exist which offer much larger bitrates than the Internet and often use fiber-

optic cables as opposed to the coaxial cables of the Internet. The transfer mode for B-ISDN is called the Asynchronous Transfer Mode (ATM) which resides in the data link layer of the OSI model.

TCP/IP was originally developed by the Department of National Defense as a network-routing protocol. It assures reliable transfer of *frames* of data in a full two-way communication. Reliability is achieved as the TCP frame format contains both sequence numbers and error-checking. Sequence numbers let the receiver keep track of the frames being sent. An acknowledgment number is sent by the receiver to the sender so that the sender may keep track of successfully received frames. A control field indicates the type of frame being sent, and a checksum is used to detect errors in transmission and of course the data is also included in the frame. In TCP, the frame size is variable. The IP part of the protocol routes messages across an integrated system of networks such as the Internet. It may, for example, break frames into smaller frames for the purpose of routing and then reassemble them at the destination [15]. It should be noted that it is possible to run TCP on fiber-optic networks but that the TCP frames will be a bottleneck to the bitrate.

The ATM protocol or the “cell relay” protocol enables high-speed transport and the multiplexing of signals from different media (i.e. text, graphics, video etc.). ATM can be run on any medium but was intended for a fiber-optic medium such as the OCRInet ATM network in the Ottawa region. Like TCP, ATM is a frame protocol but differs in that ATM frames, or *cells*, are of *fixed* length - namely 53 bytes. The fixed size simplifies switching, reduces network latency and allows for better multiplexing. The ATM cell consists of six codes and the payload. Of note is that logical connections for unidirectional data transfer between endpoints of a link may be established, called *virtual connections*, which guarantee a certain quality of service. Users on an ATM network may be guaranteed a non-shared bitrate of 150 Mbps which is sufficient for constant bitrate applications such as video-on-demand and video conferencing.

2.3 Emerging Multimedia Standards and Forums

As the importance of multimedia grows, more and more communications forums, committees and special interest groups emerge to attempt to bring some kind of standardization to this quickly growing field [10][11]. Three such groups, and their goals, are summarized below [12].

- **Multimedia Communications Forum (MMCF)** ; to accelerate market acceptance of business-oriented networked multimedia applications. The applications group logs multimedia standards while the applications group generates service descriptions for applications. This group includes players such as IBM, AT&T and National Semiconductor.
- **Multimedia Communications Community of Interest (MCCOI)** ; to accelerate global acceptance of multimedia applications which use broadband switched digital networks and to promote standards and tariffs. Some key players are France Telecom, IBM and INTEL.
- **Interactive Multimedia Association (IMA)** ; to promote the development of interactive multimedia applications and reduce barriers to this technology. Focus groups examine such topics as scripting languages and multimedia data exchange. Key players include Apple, DEC, HP and IBM.

Not only is there a need for committees to promote multimedia but there is a hard and fast need for standards on some important aspects of multimedia. Recall that multimedia may make use of text, images, audio and video.

Text is not an information-heavy media. Images and most importantly video, on the other hand, are. If one is to store large amounts of image or video data it must be compressed in some way. Standards for image compression exist - the most prelevant one

is the Joint Picture Experts Group (JPEG) standard. JPEG can compress an image to a ratio of anywhere between 10:1 and 50:1 [13]. For motion video the Moving Picture Experts Group (MPEG) standard is the most widely accepted. This compression scheme can achieve compression rates between 50:1 and 200:1. Other standards include QuickTime and Digital Video Interactive (DVI). Fractal compression is a relatively experimental compression scheme which can, in some cases, achieve phenomenal compression rates by exploiting the self-similarity of images.

Multimedia documents are another area where standards are fighting to emerge. The following is a brief summary of two such standards [14].

- **Hypermedia Time-Based Structuring Language (HyTime)** : suited to time-based documents such as music scores. It is essentially a markup language that allows the definition of structured document types and allows linking between media. Coordinate systems can be established as well as layout and timing.
- **Multimedia and Hypermedia Information Coding Experts Group (MHEG)**; focuses on the structuring and linking of different media types and interactivity between them. It allows special *objects* to be built, using a special coding, from any media type. MHEG is suited for situations where information has been prepared and is stored and ready to be viewed but is *not* considered ideal for document authoring or multimedia editing.

Document standards are important in the context of multimedia applications. If a standard structure emerges it will allow application developers to develop exchangeable documents which present information on daily news, university courses, etc. This will make it easier to establish an information highway which can distribute information documents to a variety of platforms and users.

2.4 The MEDIABASE Project

In this section we discuss the existing platform (not part of this thesis) upon which the much of the work on our multimedia news system depended. We provide an overview but refer to other publications for further details since this section is not meant as a detailed description of all the research components.

The purpose of MEDIABASE was to develop an advanced high performance multimedia information and communications system with a particular focus on document architectures, database models, high-level communications and synchronization protocols, and real-time physical storage of multimedia data. Objects manipulated in MEDIABASE are composed of text, graphics, still image, motion video, audio, and voice.

Applications such as video on demand were designed with a specific objective in mind which is to provide remote delivery of audio and video data stream to the home or to the office by using a high speed network such as ATM. An interactive multimedia information can be much more complex. Such a system can be thought of as composed of the following main components as illustrated in figure 2.2:

- A powerful document model capable of structuring multimedia information as well as representing the spatio-temporal relationships that may exist between different media objects. Examples of multimedia documents are stories, courseware, a video clip, a textbook, electronic news, movies, medical reports etc.
- A production service to acquire the media from different sources and create, modify and manipulate multimedia documents. Typically this component is equipped with advanced electronic devices to capture and display various quality of audio and video media. Documents produced by these modules may combine different media sources such as camera, VCR, CD-ROM and on-line multimedia databases and may be stored locally or transferred over the network to the multimedia database for their storage.

- **Multimedia database** to store and manipulate multimedia documents. It includes the specification of the meta data (e.g. data model, indexes, etc.), the data structure that is used to store information in the database, as well as the information content.
- **Multimedia communications system** to transfer multimedia documents over the network and to handle the remote interaction between the user and the multimedia database. Among the requirements are appropriate bit rates for each type of media, synchronization (of mono media and inter-media) and time constraints (real-time communications, acceptable user-feedback). Further, protocols which promote error-free and jitter-less transmissions are desirable.
- **Multimedia user interface** to present multimedia information to the users and to support the interaction between the users and the multimedia system. Different multimedia application profiles may have various specific features that the user interface should support.

Figure 2.2. Generic Architecture of the MEDIABASE System

2.4.1 Structuring of Multimedia Information

The first step towards the design of a multimedia information system is to provide an integrated and homogeneous way to describe, organize, and structure multimedia

information objects, and to represent their temporal relationships in a single entity called multimedia document. In the MEDIABASE project we have developed an architecture, called MEDIADOC, for structuring multimedia documents. It also includes a rendering synchronization scheme that enables the specification of temporal characteristics for multimedia objects. MEDIADOC had to include:

- *a powerful architecture* that adequately describes multimedia documents to the extent that is required by a document creator.
- *an all-purpose architecture* that allows the creation of a wide range of multimedia document classes and types of applications.
- *a simple architecture* in the sense that it should not be overly difficult to create multimedia documents, and that their structures should be clear and concise so that they can be easily interpreted, understood, and modified by a user.
- *an effective rendering synchronization scheme*, that provides a useful set of synchronization specification types for the objects in a multimedia document.
- *a simple rendering synchronization scheme*, so that the task of creating rendering schedules for multimedia objects is not unnecessarily complicated or cumbersome.

The basic functional document model is illustrated in figure 2.3. In multimedia documents, information is divided into elementary units which are indivisible fragments that contain information. Examples of multimedia document units are images, video frames, voice segments, and text segments. The *content* of a multimedia document is determined by the content of each unit involved, as well as the spatio-temporal relationships that exist between these units. In fact, spatio-temporal relationships are important components of content, because the meaning of the overall information can be altered by changing these relationships. In order to describe temporal relationships in the content, a schedule must be created. This schedule, referred to as a *scenario* or script, describes when and for how long each media object should be rendered.

Figure 2.3. Multimedia Document Model

The content of a document is organized according to two different structures : logical and layout structures. The principle concepts used to express the structures of documents are the notions of abstract objects, hierarchical links between these objects, ordered and unordered objects, and shared components. A complete description of the multimedia document architecture is given in [17].

2.4.2 Database Architecture

The main idea the design architecture of the system is the use of the multimedia document architecture for the conceptual data model. Multimedia document architecture describes the objects, their types, and their relationships in both time and space. Once a multimedia document standard has been defined and widely adopted, it can be employed in a multimedia database as a conceptual model in a way similar to the entity relationship model. Operations (i.e. transactions) are defined on a document as a whole or on different parts of the document and they include storage, retrieval and updates.

Figure 2.4. Multimedia Database Architecture

The architecture of the multimedia database is described in Figure 2.4. It consists of three levels:

- **Physical level** is composed of an object oriented database management system and a high performance storage system. The former is used to manage the multimedia document structure and the latter is dedicated to management of the media content objects. Note that this separation is transparent to the users (e.g. applications).
- **Conceptual level.** Application classes define document structures which can be stored and retrieved using a set of global class schemas that describe the document. A set of operations is associated with the global class schema allowing the manipulation (retrieve, playback, store, update, link, synchronize, copy, move, select etc.) of the multimedia document structures as well as the manipulation of document content objects.
- **External level.** The multimedia database management system acts as a server for one or more different clients. A set of primitives are provided by the client according to the application view and profile (two applications may have two different views of the same document). The user interface is built on top of this client system.

2.4.3 Multimedia Storage Subsystem

One of the most important and challenging aspects in multimedia technology is the design of a real-time recording and playback system that will write and read continuous media directly on and from a secondary storage device. In fact multimedia information requires a new physical data structure for their efficient storage and retrieval. This data structure must meet the real-time constraints required for the rendition of a media (e.g. audio and video). It should also provide facilities for the handling of synchronization between two or more media. In the MEDIABASE project, a sub project called MEDIAFILE exists whose purpose is to provide an integrated storage and (continuous) retrieval system that physically stores and retrieves multimedia information from a dedicated secondary storage device with respect to the synchronization properties. The system constitutes an independe

nt multimedia file system with (modified) traditional file features (open, close, write, read, append, directory, etc.). It is used in conjunction with the object oriented database system to form a multimedia database management system.

2.4.4 Synchronization Management

MEDIABASE should guarantee the real-time retrieval and delivery of multimedia information. This information must be ordered and scheduled in order to maintain the desired service quality defined in the multimedia document. The real-time aspect is characterized by the temporal relationship that may exist between multimedia information being retrieved or delivered to the end user. The multimedia document is a self-contained entity that can be created, transferred over the network, stored or retrieved from the

database and presented (playback) to the end user. In each of these activities the components (e.g. database, client, network etc.) involved in the manipulation of the document have to ensure that the real-time actions on the document are performed with respect to the temporal relationships defined in the document. The temporal relationships together with the actions on the document are referred to as synchronization. There are three levels of synchronization in MEDIABASE: document definition and presentation level, storage level, and network level which we briefly introduce next.

Due to the random network delays and different data processing speeds and clock errors, the transfer of a multimedia document on the network for a remote presentation to the user requires a synchronization control algorithm. The algorithm must maintain the temporal relationships defined within the document. Because each media type has a different bit rate, a network with variable bit rate such as ATM is suitable.

MEDIABASE uses an architecture that defines two algorithms that can be used by the synchronizer to achieve the presentation of multimedia documents over the network - retrieval and delivery algorithms. The former is depicted in Figure 2.5.a and the later in Figure 2.5.b.

- Retrieval algorithm uses the logical structure of the multimedia document to perform a progressive retrieval of the content from the database according to the scenario located in the document structure. In this algorithm, the synchronizer is located at the database client side (e.g. at the display workstation).
- Delivery algorithm assumes that media are stored in a physically separate machines. In this case the synchronizer and the scenario are replicated in each server. The synchronizers engage in a collaborative negotiation to decide when to start sending the document to the workstations for its playback to the user. Further detail can be found in [44][45].

Figure 2.5. a. Synchronization by Retrieval Algorithm

Figure 2.5. b. Synchronization by Delivery Algorithm

2.4.5 How MULTINODEs Fits into MEDIABASE

MULTINODEs is a multimedia application built upon an existing technological platform - namely MEDIABASE. MULTINODEs is an application layer on the top of MEDIABASE. Software at production sites and user sites implements client routines and graphical user interfaces which facilitate the creation, transfer, manipulation and presentation of multimedia news information.

2.5 Technical Challenges of Multimedia News

The challenge of determining technological requirements for a multimedia news application is that they largely depend on the feature set of the system; a multimedia news service could come in many variations. If the call is for news which resembles W3 documents then Internet bitrates are sufficient, as is TCP/IP. Note that this is only if video is downloaded in its entirety upon its request. Media such as video cannot be sent

sequentially (bit by bit) over a network such as Internet since it cannot support the kind of bitrates needed for this activity.

It is clear that if fully synchronized multimedia news documents or video streams are to be transferred over a network then fiber-optic networks are preferred - as is the ATM protocol, for reasons suggested in the previous section. The biggest challenge then, may be deciding how to create a hybrid multimedia news systems that allows viewing of both types of news on either Internet or OCRInet. That is the challenge that we answer in the remainder of this thesis. In the next chapter we discuss MULTINODES, our multimedia news system which was developed upon the existing MEDIABASE platform.

Chapter 3

MULTINODES - Feature Set and Architecture of a Multimedia News-on-Demand System

During the search for the answer to the question, "*What should multimedia news look and feel like?*", we came to the conclusion that we did not want to fall into the trap that others have fallen into [6]. That is, we did *not* want to try to create a system that exactly duplicated the experience of reading newspapers. It was determined that would not only be an incompatible mapping but also an injustice to the technology that we had available to us. We decided that, to best create multimedia news for computers, we should let our imaginations run free in order to break free of the old, formal, familiar ways in which newspapers are browsed and read. Not that newspaper-reading techniques should be discarded altogether- in fact it is important to preserve some of the newspaper metaphors so that this new medium will not alienate the old readership.

Previous attempts at on-demand news systems have revealed several of the key aspects of architecture and functionality that should be integrated into the system; in this work, we attempted to provide at least those functionalities as well as including some unique and novel features. Since the objective of a system such as this is to provide daily

information in a user-friendly way, the application running at the user site should be a unique blend user-friendly screens and powerful search, query, and browsing mechanisms while maintaining the transparency of data distribution. Also, it should offer novel features that will facilitate users *in the context of multimedia news*. Note that video presents particular challenges in this area and current research is striving to solve these challenges. This will be discussed in the next chapter.

Creating and defining the architecture for a project of this scope was also challenging; among other reasons, the system consists of a number of participating groups (not including users) not all at the same location. These participants include the Ottawa Citizen, Algonquin College and the University of Ottawa. Users, distributed on the participating network, OCRInet or Internet, include students at Carleton University or Algonquin College, employees at Bell Northern Research (BNR), Newbridge or the Communications Research Center (CRC). Our architecture is novel because it not only introduces new protocols but lets users use standard and widely available protocols and networks to communicate with the database server. Further, the way that the corresponding servers are nested is novel and effective.

In this chapter we describe the key features and tools that are available to the *users* of our multimedia news while explaining why they are important features to have. We then describe the unique functionality available to *authors* and *document creators* (however, authors may also make use of some of the former tools). Next we describe the architecture of the system and all the components involved. We then discuss our novel ideas about structure and protocol which helped define it. Finally, we offer some examples of user interactions with the system and explain how each uses the architecture and protocols.

3.1 Goals of MULTINODES - Defining Multimedia News Systems

After studying many implementations of multimedia news or systems related to news [1-10][13][23][31][33] and independently analysing what we thought were important issues, we reached some conclusions on how multimedia news of the future will behave, what it will look like and who and what will support it. These goals form the basis of the construction of MULTINODES, a system which emerged directly from these early goals. These goals also form the basis for comparison when we studied other systems. In our view, multimedia news of the future will be:

- **powerful**; consumers must be able to perform complex operations involving large amounts of data in a simple way
- **intelligent**; the system not only does what you tell it but also performs actions on your behalf based on what it thinks you will appreciate
- **familiar**; as a blend of existing media, multimedia news should maintain some of the effective features of traditional newspapers
- **innovative**; new and interesting features which exploit technology and bring a new quality of service to the news service should exist
- **personal**; unlike television and newspapers, news of the future should be transmitted at individuals. That is, the news depends on who you are and what you like.
- **user-friendly**; user interactions should be easy and non-threatening
- **distribute the expertise**; communications networks should allow the geographical distribution of the parties involved in a news system. This means that journalists, artists, and advertisers need not occupy the same physical setting.
- **promote learning**; multimedia news as a new medium should be informative as well as fun and should promote learning and exploration of knowledge.
- **archived**; news should be archived - just as traditional news is put onto microfilm, large databases should make available old news articles
- **a new type of medium**; it is time for the television and newspaper to merge - that is, news of the future should offer both dynamic (the news "plays back" in time like television) and static (like traditional newspapers) types of news

- **supported by a robust dependable architecture;** multimedia news requires that the technological support be available. This includes databases, communications, and document architectures.
- **on-demand;** this means that the information that they want, when they want it, as opposed to it being broadcasted at a certain time and/or place.
- **for news-producers and news-consumers;** news-producers will have to be supported with technology and innovation in order to create a multimedia version of the news - these tools and feature should be shared by both news-producers *and* consumers.

Also, the system needed to:

- **involve specialists;** a news service must be a collaborative entity between technologists and people with journalistic expertise. One should not expect technologists and scientists to create the news content.
- **use ATM networks;** this is necessary to achieve real-time transmission of multimedia news information - e.g. videos, movies, etc.

We feel that these criteria will lead to a news service that is far more powerful than traditional newspapers and more effective than television news. Later in this chapter we discuss how MULTINODES has each of the qualities listed above and why.

3.2 User-Oriented Features

In this section we discuss tools and features that we feel should be available to users of the new medium. For clarity, we discuss these separately from those offered to authors but in fact some of the tools may be used by both types of users of the system. Each tool builds upon one or more of the qualities listed above which define our view of multimedia news.

3.2.1 The Look of the New Medium

Since our goal was to create a new medium that resembled both television and traditional newspapers the resulting system has these characteristics. MULTINODES has the ability to incorporate two types of news. The first type is called hypernews which is essentially hypermedia news. In this type, much like a newspaper, the emphasis is on reading the stories and examining photographs. A large number of details can be presented in this kind of medium, arranged into a web-like structure of information. Webs of this sort can dramatically increase learning and fun [46]. The second type of news, synchronized news, more resembles television news in that it is played back to the user continuously when the user clicks on the play button on the control panel. Also on this panel are at least: stop, pause, rewind, and fast-forward. Associated strongly with this type of news are video, closed-captioning, and any media that need to be synchronized together through time.

We have selected a heirarchic *headline-metaphor* for presentation of news stories. Headline level detail is displayed first. After selecting a headline, the user is presented with the story, viewed either with the hypernews browser or the MediaPlayer, depending on whether it is hypernews or fully synchronized news. The viewer is spawned automatically so that this choice is transparent to the user. We feel this is an appropriate method since it is the method used by newspapers, radio and television (often, particularly in radio, *only* a headline level newscast is provided).

When hypernews articles are to be read, a Mosaic-type application is launched in order to display them. The article is then *active*. It may be browsed like any other hypermedia document; hot-spots, or links, are embedded in the text or images in the article and may point to some other object of any media type - text, image, audio, video, or synchronized news document - in any database. When the user selects the link, the appropriate media object is retrieved. In this way, the hypermedia news document is browsed *on-the-fly* from the multimedia database since each link initiates a database request

which yields the desired object. Since hypernews articles are written in the Hypertext Markup Language (HTML) they are parsed and formatted upon arrival at the user's application. As an authoring tool, this language allows the news articles to be physically formatted in an agreeable way upon presentation. It offers many different font styles and sizes, the ability to in-line images on the pages, and the ability to create links to any other type of media object.

When fully synchronized articles are to be viewed, it is necessary to launch the MediaPlayer [38] - an external player not part of this thesis. This player takes care of all network synchronization issues by establishing connections directly with the database server, requesting the appropriate media objects and displaying them to the user. The MediaPlayer knows which objects are needed since synchronized news articles are stored as multimedia scenarios which specify all objects and the temporal relationships between them. Further, its interface is an intuitive GUI with large push buttons to start, stop, pause, rewind or fast-forward the presentation.

The news application's front-end interface is a unique, graphical, highly-functional tool. It contains routines which enable the user to do all of the administrative-type chores that go along with news viewing as well as begin a viewing session. The interface consists of three main components: a menu-bar, text-fields for document and profile selection and a tool-bar. The menu-bar is of the pull-down sort and allows the user to perform anything that the tool can do; document specification, profile manipulation, search, follow-up, view the key-word tree, initiate a video-browsing session, or scrapbook manipulation. The tool-bar offers the same functions as the menu-bar but presents them in the form of easy-to-read and find push-buttons. The text-field area allows users to specify the database, document, and profile names. Any of these names may be entered directly or, to save keystrokes, selected from scrolled lists. The largest and most prominent area on the interface is the *Daily News* button. This is because by a single click on this button the specified news document is retrieved over the network and displayed. Finally, the interface

has a hierarchic *Help* menu should the user become confused or unsure about what he/she should do next. We believe that this interface is unique in the sense of interface functionality. Other comparable news systems do not have interfaces which offer such easy ways to manipulate user data.

3.2.2 Continuing Traditional Features - the Follow-up

The follow-up is a feature that can effectively be mapped into the new medium. This serves to both illustrate the power of technology and to add familiarity to a news service that some users may consider very different from traditional news.

Through the use of keywords, the reader may *follow-up* any article or media object that he/she has just viewed. By selecting the follow-up button or menu choice, the reader is presented with articles and media objects related to the current article. This is accomplished by a search of the database in which we seek articles and objects which have the same keywords as the current article. The articles and objects that are found are presented in a list of hits, ordered in descending order by strength. Strength of hit means that if article A has keywords *politics* and *Chretien*, and articles B has only *politics*, and our current article has both *politics* and *Chretien* as keys, then the follow-up will list article A as a stronger hit than article B (because it matched on more counts than did B). The range of the follow-up can be the document or an entire database.

We feel that this feature is important in that it partly mimics a key element of traditional newspapers. Very often in newspapers, stories and editorials are linked in a similar way. For instance on the front page there are often small areas which describe the lead stories in the interior sections followed by a page identifier, e.g. "*Garth Brooks releases greatest hits CD - Story on page G1*". Alternatively, a story may refer to an editorial on the same subject matter - e.g. "*see editorial on page E2*". Our electronic follow-up is even more robust since it searches for related articles in the entire database and

thus may return a particularly relevant article from *last week's news*! This can certainly *not* be achieved in a traditional newspaper. Figure 3.1 illustrates the semantics of a follow-up.

Figure 3.1. A follow-up initiating a database search resulting in a list of hits

3.2.3 A Personal Experience

A subtly different but more interesting way of presenting information, is to do so by *user profile* or preference. We believe that each user should have his/her own customizable profile. At creation time, authors and editors must somehow be presented with a number of keywords from a fixed news vocabulary ; for instance: *cinema, sports, hockey, business, stocks*, columnist names etc. They must associate any number of these to the article. Also, free-form (random) words may be associated with the article (this is important for names such as *Chretien*). Analogously, users create profiles by selecting words from the news vocabulary and are also given the option to add their own words to the profile. Users should be able to easily create, retrieve, edit and remove custom profiles. Pre-made profiles are also available for downloading so that, for instance, instead of having to create a profile that covers many general areas of business, the user may choose to just download a pre-fabricated profile and modify it to his/her needs. When a user requests multimedia news filtered through his/her profile, only articles that match the current profile are retrieved.

Consider now an enhanced user profile which contains more than simply text - e.g. a *multimedia profile*. The traditional way is to supply key-words such as *Gretzky* and

hockey, but consider the possibilities of supplying an image. Let us say that a user likes hockey and is a fan of the player Wayne Gretzky. The user would not only like articles regarding these topics but also *pictures*. Why not then let the user supply an image of Wayne Gretzky (perhaps a close-up of his face) and a number of images of hockey players in action in his/her profile? In this way, articles and images are first image-processed to determine their relative closeness to the supplied images; if a “good match” exists then the image is accepted. This could be accomplished using advanced pattern matching (to find other close-up’s of Gretzky’s face) or pattern and colour matching to find other images of hockey matches. But why stop at images? Consider a film student who wants to find other videos which match the filmmaking technique of a film that he/she has digitized. The user should be able to add this video to a profile and let the system find related videos. This would involve first analyzing the video content and extracting camera data such as pans, zooms, etc. This data would have to be compiled into a kind of sub-profile describing the camera techniques of the video. This sub-profile could then be compared to other videos in the database so that matches may be found. Figure 3.2 illustrates what is meant.

Figure 3.2. A multimedia profile

Ultimately, the profile should be smart enough to figure out what the user likes. This can be done by initiating a watch-dog program that watches what type of data the user views, for how long, etc. and hence slowly learns user-preferences. These are called

intelligent user profiles [4][5]. For example, if a user consistently selects articles about *automobiles*, the intelligent profile adds this topic automatically to the user's profile.

What we consider important in all of this is that we have chosen to add *random stories* to profiled news to "liven up" the presentation. In this way, even though users are getting highly filtered news articles, they also get the occasional random article - this helps break the potential monotony of filtered news. In fact, when browsing or reading a newspaper, we are often caught off-guard by a story which grabs our attention - this story is often *not* one that was necessarily being searched for, or for that matter, even be related to any of our *usual* interest areas in the newspaper.

3.2.4 An Intelligent Medium

Intelligent agents are receiving a lot of attention in a wide range of applications including USENET news [8]. An agent performs some action or makes some decision on behalf of its user. It makes this decision or action either because it was supplied with a rule by the user which prompted it, or it has a knowledge base from which it deduced that the action was appropriate. Agents can also learn by watching their users use application programs and by examining user satisfaction with the choices or actions they have made [22]. In general, there are two main issues; *competence* and *trust*. Competence addresses how the agent acquires and uses salient information, and trust addresses the level of faith the user has in his/her agent to make decisions and actions on his/her behalf.

In the news context, we feel that users should be able to dispatch agents, or *software robots*, to scout the distributed database servers, communicate with other users' agents, and search for particular products and events in the news. One might instruct an agent to *keep an eye out* for articles on the prime ministerial visit, or, if one is involved in stocks and investments to watch for articles on a particular stock or company. If articles of interest are found by the agent, the user is notified with an alert explaining what was found.

As competence and trust increase we see the agent browsing through multimedia classifieds, looking for deals at electronic grocery stores, skimming through multimedia real-estate offices and even initiating the purchase of a house, all on behalf of the user. Agents introduce social and ethical questions about privacy and security.

3.2.5 News as it Develops

News stories go through an *aging* process as they develop in time. These aging stages should be available to the user so that if desired, a story's development may be re-viewed. The documents that include these stories are themselves archived so that old news editions are available to the user for browsing and searching.

We feel that this is an important feature. Consider an official visit by Chretien to Paris. As the tour progresses the story will do the same. For instance, the morning edition of the newspaper may reveal what Chretien will talk about with the French leaders and what Canadian concerns will be tabled and why. The afternoon edition however, will go into more detail - largely because the meeting has now taken place and the results are known. It will reveal the results of the meetings, the leaders' reactions, and the analysis and ramifications. So we see that articles can change on an hour-by-hour basis. For this reason, our system attempts to maintain stories for as long as possible so that a reader may re-live the entire progression of news, see the pre-analysis, read about the event, and finally view the results and ramifications. Other types of stories change largely in a day-by-day fashion. An example of this would be a running story on the World Series. This is a 7-game 10-day event which changes dramatically every night. These changes should be available to users.

3.2.6 Compressing Time with Temporal Proxies

We believe that *proxies* should be incorporated into the news context. Proxies are a way to scan a large amount of temporal information quickly and graphically [26][27][36]. It is important to try to discover how to present news-at-a-glance. Multimedia news scenarios are indeed temporal - they are scripted on a timeline and rendered in strict accordance to this timeline. We are investigating proxies for multimedia news scenarios and for video. Of course temporal proxies apply only to temporal media such as multimedia news articles - hypermedia news articles are not temporal but can be represented in ways which allow users to view global context as well as local detail. One such way is called a fisheye view [36].

We have seen that one way to preview a news document is to actually play the document over the network to the user, but in a compressed, rapid manner. The idea is to send just enough information that the user *gets the drift* of the article in order for him/her to establish whether or not it's relevant. It is non-trivial however to determine *how* compressed and *how* fast to preview the document. Another way to accomplish news proxies is to map the scenario onto the video-tiles metaphor that we will discuss in the next chapter. By doing so, a scenario is represented in different levels of granularity by sub-tiles, or icons, which represent the key moments - e.g. when a new media object begins to be rendered. The sub-tiles may be browsed by the user if desired, hence we introduce the concept of scenario browsing.

The challenges of creating an graphical object which represents a video stream will be discussed in the next chapter. The key idea is that we do not want to create a textual proxy for video as this would be an incompatible mapping. We want a representation which reveals content as well as temporal information graphically.

3.2.7 Power to the User

The news systems of the future will empower users with the ability to manipulate, rearrange, assemble, download, search, query, and order large amounts of information. Querying and searching are particularly important in the system. Together, they should support a user who wishes to search for an article or media object relating to something *specific*- for instance a video clip of Jean Chretien playing golf- or something *general*- for instance a video clip about war. Queries and searches are available in our system within media objects, stories, or the database and can be accomplished in similar ways to user profiles. Media objects such as video and images may be used as search criterion, or, simple key-words may be used. Ultimately the goal is to realize *multimedia perception* - a multimedia query specification language with a seamless interface. This creates a powerful news searching engine which other comparable news systems do not have.

3.2.8 Encouraging Learning and Interactivity

As interactive multimedia technology infiltrates more schools and homes, the masses will see that it is a very effective tool for learning and education. Students are not clean slates onto which educators must put their knowledge. Learning is a two-way process where meaning is constructed progressively. Interaction and learning-by-doing is very crucial [46].

These ideas are manifested in an interesting feature - that is, the idea of *multimedia scrapbooks*. That is, with certain arrangements, a user can be allocated a certain portion of the archive for his/her own use. Alternatively, and perhaps better, the user could simply store a pointer to the article of interest. Then, at any point during browsing, if an article is of particular interest, the user may arrange to have the media objects included in his/her personal scrapbook where personal annotations or custom documents may be created. This feature is not only interesting to casual news viewers but is particularly interesting in

an educational setting. Students who must do course projects can browse and search through news databases discovering articles and media that are relevant. Then, by storing references and appropriate assembling, students may create multimedia project presentations and documents to be shared with other students.

This introduces the idea of giving the news-consumers the same tools for creation and assembly of multimedia presentations as the the news-producers. We feel that this will create a dynamic learning environment for students and news-consumers.

3.2.9 Video Browsing

Since the database will collect more and more news video clips for users to view as part of synchronized documents, we have made sure to include in our project a content-based video browsing tool. This tool will allow users to preview large video clips before watching them or browse them down to a low level of granularity in search of particular events. We discuss this concept in detail in the next chapter.

3.2.10 Advertising

What would a news service be without *advertising*? Multimedia advertising is an issue that raises some interesting questions. In an interactive multimedia news system, should advertisements be intrusive or should the user be able to avoid them altogether? In one way, the presence of advertisements in the news of the future will imply that the user is able to reach the vendor directly through the communications medium to discuss and order products. This is very desirable assuming that when the user becomes tired of seeing the advertisements in his/her news, he/she may "turn them off", or disable them. Already, in today's on-line world, companies have recognized the difference between the demanding net-surfers who if not happy will simply click the mouse and "fly away" and the passive television viewer who watches almost anything. Because of this, on-line advertisers

attempt to deliver their message in successive layers as a part of a sort of *dialogue* with the consumer. By making consumers make selections and answer questions about themselves as they go the message can be tailored to the individual. In Club Med's hypermedia pages on the W3 project, users are separated into categories by first answering a few questions. Then, specific information may be dealt to them depending on whether they are a family, single, etc. [23]

We believe that solutions can be found such as; if the user pays more, then more advertisements and links to sponsors are provided. We feel that advertisement is best handled with profiles and intelligent agents - that is, give the user control over what he/she must view. The profile should serve as a filter to eliminate advertisements which are of no interest and the agents can be directed to search for advertisements (including searching through multimedia classifieds) which may be of interest.

We believe that advertisements should also be linked to editorial content. Imagine a video clip of a new movie by Sylvester Stallone in which he portrays a motor-bike racer. A local motorcycle dealership that carries that same line of bikes or boots that we saw in the clip may want to provide a link to their dealership directly from or after the movie clip. In this way, zealous fans can immediately purchase the same items that the actor wore or used in the clip simply by initiating an ordering session. This should apply to all types of products - large or small. For instance, products from stereos to grocery items could be linked to the originating vendor for ordering or for more detailed information.

3.2.11 Interacting with Databases through Key-word Trees

Key-word tree visualization is a feature that can be very informative to users. Recall that every database has a hierarchy of key-words that serves as indices to documents in the database. The tree is not unlike a directory tree - at the top is ROOT; at level one we could have; SCIENCE, TECHNOLOGY, SPORTS, and ARTS; at level two under SCIENCE we might

have; PHYSICS, CHEMISTRY, BIOLOGY etc. Associations to the key-word tree are made at authoring time - new documents are associated to words in the tree, or, new key-words are added to the tree and the documents are associated to them.

When one considers the size and scope of multimedia news databases it is apparent that a feature such as this will be useful. A multimedia news database may contain news documents for an entire week or month or even longer. The scope of the articles, as in traditional newspapers, will of course be very broad. By enabling users to view the keyword tree in text format or in a graphical tree structure we give the user the power to perform a high-level browse of the content of the database since the keywords are associated to content data. Further, assume that there is a branch in the tree that hits the following key-words; ROOT, SCIENCE, PHYSICS, EINSTEIN. By simply locating this branch and then clicking on EINSTEIN, the system automatically retrieves the document associated with this key-word - an article about Albert Einstein. We believe that this allows for powerful high-level database browsing. When one considers that the best solution for a high-level browse in a traditional newspaper would probably be to browse headlines in some organized, hierarchic way, the key-word tree feature becomes appreciated. What it effectively does is to create the order and hierarchy necessary for browsing and it effectively chops headlines down to a single word (e.g. the key-word) for compact browsing.

The key-word tree visualized is also used at production centers. Recall that when an article or document is created it must either be associated with an existing key-word in the tree or a new key-word must be created. This process is sped up dramatically by letting authors simply view the tree and select the node, or word, to which the article will be associated. The link is then made automatically.

3.3 Author-Oriented Features

More and more, traditional newspapers are taking advantage of computers and networks to increase the efficiency of news production. One would expect that multimedia news would do at least the same. In this section we discuss the ways in which our system is unique as an authoring, assembling and maintenance tool, focusing on hypernews as opposed to synchronized news. Not only are time-saving features included but all the functionality is available over the participating network whether it is an ATM network or the Internet.

In our view of multimedia news, creation of articles and information should be as effortless as possible with editorial content not necessarily created and edited at a single center. As discussed in previous chapters we see content creation as a distributed process. In terms of hypermedia news which consists of text, graphics, images, video and audio it is worth recalling that hypermedia, and it's precursor hypertext, is a web-like structure of information. The web can be thought of as consisting of nodes and directed arcs - nodes correspond to media of *any* type and arcs to links between them. If nodes A and B are connected then B can be reached by some selection at A.

Figure 3.3. A hypermedia web. Nodes represent media, arcs represent links

Figure 3.3 represents a hypermedia news story. Node A is the initial screen. It contains three links to other media; B, C, and D - that is, any of these may be reached by a particular user action. A selection on B allows viewing of node E etc. Hypermedia news originates

at production centers as a number of nodes or media objects with links to other nodes. The important distinction is that all of the nodes reside on a file system.

In MULTINODES, we archive hypermedia news articles in a multimedia database, so users browse articles *from the database not from a filesystem*. The benefits of using a structured database which supports queries, versioning, etc. as opposed to the file system are obvious and have been discussed previously. It remains then to map the news document created by authors and editors on the file system into the database. Documents in the database have the basic logical structure shown in figure 3.4.

Figure 3.4. The basic logical structure of the multimedia document

Our novel feature then is the automatic hypermedia web traversal and mapping to the multimedia database structure. Note that the mapping is incomplete since hypermedia news documents have no concept of time whereas the multimedia document has all necessary ingredients to specify a synchronized document. The algorithm is a depth-first traversal through the web starting at the home page or front page. Paragraphs are created based on links from the initial node (i.e. we assume that the front page partitions the information into logical sections using the links) and nodes are successively put into the appropriate paragraph as they are hit in the depth-first search:

```

A:=first_node;                                /* the front-page or home-page */
For (each child Ci of A){                      /* create a paragraph in the document */
    create a paragraph P(Ci)                  /* this is analagous to creating sections */
    create Media object Ci in P(Ci)
}
For (each child Ci of A){                      /* put all related media into appropriate sections */
    perform a Depth-first traversal putting all nodes into paragraph P(Ci)
}
  
```

The document structure created by mapping the hypermedia web from figure 3.4 into a database is shown in figure 3.5.

Figure 3.5. The result of mapping the web to the database

This feature makes hypernews creation very simple for authors. They need only create their web of information, specify a database and document name, and ask the application to perform the mapping for them. In this way all objects are moved to the database where they may be queried and browsed.

Updating news information is a very important feature to have - particularly for articles and objects which concern highly volatile information such as stock prices or sports scores. News articles may be updated several times daily so an effective interface for editors and authors to make these updates is necessary. In our system authors may interact with documents over the network. From any site they may select a database for updating, choose the appropriate document and update one of its media with new data or an alternate media. All of this is done using a simple, graphical, list-based interface. For example, if stock prices are coming in hourly over the wire from a Dow Jones server then the list is simply captured into some text file with any ordinary editor, and then swapped into the database replacing the stale text object which is now embedded in a multimedia document.

Removal of media objects is accomplished in a similar easy way. Authors may browse through the structure of a database and then through the structure of a document, choose a media object and remove it from the document. Similarly, a paragraph in a document or an entire document itself may be removed when it becomes too stale to keep

archived. Of course, removal privileges are given only to authorized personnel so that the news database is not vandalized or accidentally changed by users.

These features necessarily make hypernews updating and removal effortless. This is extremely important so as to minimize production time and get updated news articles onto the database server in a timely fashion. Other comparable systems do not have such tools that make these processes flow smoothly.

Of course as documents are being created and updated it would be useful for authors to be able to pre-view their documents and browse through the content database in search of other media objects to include. MULTINODES provides a simple interface to accomplish both of these tasks. Searching through the database is a simple matter of mouse clicks on appropriate buttons and pre-viewing a document is accomplished by simply using the retrieve functions to fetch the article - it is then displayed automatically.

3.4 Architectural Overview of the System

The main architectural goals for the MULTINODES prototype were as follows: utilize ATM networks for the bandwidth advantages and to allow the distribution of the expertise involved in the project, utilize components of the existing MEDIABASE platform, get the prototype in operation quickly to allow for testing and demonstrations.

After studying existing news systems [1-10][13][23][31][33] we realized that very few of these systems have the technical foundation found in MEDIABASE. Our goal then became to implement a news system involving: a main production center, a news content server and database from which production may draw from during document creation, the news database for completed documents encompassing all aspects of document aging, archive management etc., communications protocols, and user sites distributed over OCRI-net (see figure 3.6).

Figure 3.6. The generic architecture of the system

The remainder of this section describes in more detail the architecture of our multimedia news system. We also discuss how the existing platform can be used as a part of the MULTINODES system.

3.4.1 Using the Production Server for MULTINODES

The backbone of the production center is the production server. It can be the source of all of the data needed for the creation of a multimedia document. Examples of how the production server is used follow. First, if an editor deems that a particular news scenario or article requires a video clip to make it complete and the video clip resides on a VHS tape (perhaps it was captured from CBC Newsworld), then the video clip must be acquired. The production server is used to digitize the clip directly from the VHS tape and store it in a database. It does this by presenting a graphical user interface to the video capture software. A second example is that if the previous video clip is deemed to need its corresponding audio track then the server must also capture the audio, making sure that it

matches exactly with the video just captured, and then store it in the database so that the two media may be included in a scenario and synchronized upon playback.

Attached to the server in hardware are both video and audio capturing devices which allow for the capture and editing of these data types on a per-sample basis. Support for editing of graphic and text objects is also provided. Alternatively, any type of file may be transferred over the network to the production server if it is needed and existing somewhere else on the network.

The production server caters to the stations participating in production which are typically operated by the news authors (e.g. columnists and reporters) and editors. Other hardware and devices available to these users include scanners, video-cameras, microphones, and PC-VCR's. Production is greatly enhanced by the existence of network groupware and collaborative software. In this way, news stories may be jointly authored - for instance two journalists and an editor may collaborate on a story using shared applications and video conferencing as shown in figure 3.6.

Figure 3.6. Groupware news production

Multimedia news can then be created distributedly. Columnists and editors need not occupy the same physical building to have face-to-face meetings and perform collaborative

editing. Related to the production center is the news content server which we will discuss in the next section.

3.4.2 Using the Content Server for MULTINODES News Production

The content server provides production with a large repository of media objects. For example if an author requires a short video clip of the prime minister, he/she could query the content server for all clips having "Jean Chretien" or "prime minister" and "Canada" in the keyword descriptors. Most of the content must obviously be updated or expanded on at least a daily basis so that the documents reflect the day's news. The content server acts as a kind of multimedia clip-art database from which media can be seized. It is necessary then to accompany the server with a simple and effective database query engine to find media quickly or to browse the content. We discuss this in the next chapter.

News production is done mainly at two sites; the Ottawa Citizen and Algonquin College. In fact this is one of the strengths of the system; by distributing production, it can be spread to parties with superior skills instead of compromising on skills and centralizing production. The Ottawa Citizen provides the editorial expertise in the creation of the text and content of the news documents. They also have the photographers, VTR operators, and reporters who gather news on location. Algonquin College has the technical expertise in multimedia production. Audio and video editing or photograph scanning can be done at the college, or computer animations can be produced by skilled animators. Files are exchanged freely between the two sites via OCRInet, arranged into documents using scenarios, and then stored in the multimedia database for future retrieval and playback.

3.4.3 Using the Multimedia Database for MULTINODES

An application such as MULTINODES brings with it certain requirements for the database, or news repository. Frequent updates of information, user-initiated queries, and profiles are just some of the features that influence the structure of the database.

The database supports navigation and querying of multimedia objects. Media objects are decomposed into segments which represent either video frames, segments of frames, segments of audio, or portions of text. The architecture of the database is three-tiered: external level, conceptual level, and physical level (as discussed previously).

Data is stored in both high-speed active storage devices and tertiary storage devices creating multimedia news archives. The database system handles document aging- for example a news article may go through several editions as the story evolves; each should be available. It also handles frequent updates of volatile information such as stock prices and sports scores. Each database also has a key-word tree. A key-word tree is a heirarchic structure similar to a directory tree for a file system. The root of the tree is called ROOT. It may have any number of children, each a key-word for the database, as can each of these children. Each key-word in the tree can refer to, or point to, a document or media object in the database. In this way the key-word tree is in fact an index tree which is browsable and powerful which is a critical characteristic for both news-producers and consumers.

3.4.4 Using the Synchronized Document Architecture for MULTINODES

The structure of synchronized news documents is important to understand if one wishes to understand why synchronization is needed. Also, since authors create one of two types of documents, one of which is this type, we provide an overview in this section. A more detailed description may be found in [17] or [21].

The conception of MEDIADOC, a multimedia document architecture, is not a result of the work of this thesis but its function and integration into the context of MULTINODES is important to the system. The biggest contribution to the field is MEDIADOC's ability to specify temporal relationships between media objects in a document in a graphical way. Traditionally, multimedia documents are composed on a timeline where one axis of the timeline represents time and the other represents media object. Media objects are simply placed on the timeline and one can determine which objects will be rendered at any one moment by finding all media objects intersected by a vertical line at that time. Figure 3.7a. shows a timeline but note that it is, for example, impossible to know whether video1 and audio1 are to be tightly synchronized or not. Authors of MEDIADOC documents may specify temporal relationships which cannot be expressed using a simple timeline approach. These include synchronization control, offset delays and user interactions. Figure 3.7b. shows a portion of a MEDIADOC document specifying continuous synchronization between a video object (V1) and an audio object (A1). A1 is synchronized to the start of V1 ($=V1$) and V1 is delayed od_1 units after G1 ($od_1 > G1$), a graphic. Circles represent media object, rectangles represent begin or end-time constraints and dashed lines represent synchronization constraints. MEDIADOC also creates a rendering schedule for its documents so that they are ready to be synchronized and played back over the network.

Figures 3.7a. and 3.7b. A traditional timeline document (left) and a portion of a MEDIADOC synchronized document (right)

By using a synchronized document architecture in MULTINODES, it facilitated us in creating the new medium. That is, since MEDIADOC documents play themselves back, they represent the sit-and-watch type of news that was one of the goals of the system. Documents can now be created with tight synchronization (e.g. lip-sync) and captioning as well as a variety of other types of synchronization. The other type of news, hypernews, has already been discussed.

3.4.5 Using Network Communications for MULTINODES

MULTINODES is currently operational on the OCRInet ATM network. OCRInet is the world's first metropolitan area research network driven by asynchronous transfer mode technology. OCRInet, which employs fiber-optic cables to link over twenty nodes, connects technology companies, telephone companies, government laboratories and educational institutions. Network operations commenced in January of 1994 and it is a partnership between industry, government and educational institutions, that includes service and equipment providers as well as the user community in its membership. For full details on the network and how MULTINODES uses ATM protocols, see [18].

Communications over the network are achieved using the client server architecture.

This communication architecture is designed to ensure transparency in the interchange of information so as to screen end-users from the underlying architecture of the application. Distributed applications complying to the model may consist of a number of independent programs running on remote hosts scattered over the network, without the users' knowledge.

The use of the OCRInet is critical in delivering the MULTINODES goals of power and distribution. OCRInet allows communications at rates of up to 150 Mbps and connects all parties with reliable networks that can guarantee quality of service.

3.4.6 Distributing User Sites in MULTINODES

User sites are distributed over the ATM network. User sites all have client routines installed in software so that searches and queries (undirected searches) on the database, using the client-server model, may be performed.

For a hypermedia news browser, users may actually use the publicly available *Mosaic* to display hypermedia news. Users may also use our modified version of the Mosaic application which is more stream-lined and less cluttered. In this fashion, news can be created in the Hypertext Markup Language (HTML), the simple markup language that this software supports. The regular flavor of Mosaic cannot display true synchronized multimedia but we have developed a hybrid system for ATM that incorporates the features of Mosaic and HTML as well as a true multimedia player. In this way, synchronized news can also be retrieved and played on the *MediaPlayer*. The MediaPlayer plays back a multimedia news scenario over the network and ensures that media objects are synchronized together in the fashion that they were authored. For more details see [18]. To display the hypernews from the database (not the synchronized news), any hypermedia viewer that parses HTML could have been used. At the time of this implementation,

HOTJAVA [39] was not yet emerging but we see it becoming a very prominent tool for hypermedia - particularly the W3. HOTJAVA has the advantage that it is very flexible and new capabilities can easily be added; for instance if an object is requested by a HOTJAVA browser which it does not know how to deal with then a second request is made - this time for JAVA code which supports this object [39]. In the next section we discuss novel aspects of our architecture and protocols.

3.5 Distinguishing Aspects

When we chose to use the publicly available hypermedia browser Mosaic as the hypernews browser for the first prototype, we had to then operate within the constraints and protocols that this created. The reason that this software was chosen was that it is fast becoming a standard Internet tool, its look and feel are well known and it is easy to use. At the backbone of the architecture of our system then are *two* servers which respond to user requests for information. One server is the database server which serves media objects directly from the database and the other is a hypertext server to serve requests for hypernews. The remainder of this section discusses the server architecture, the database server and some issues related to the database and the schema.

3.5.1 The Hypertext Transfer Protocol

Mosaic is typically used as an Internet browser to browse hypermedia documents on the World Wide Web (W3). Recall that W3 is a project developed to create a pool of human knowledge allowing remote collaboration in a common project over the Internet [19]. W3 *client* programs (i.e. Mosaic) run locally and when links are selected, another object is retrieved from some other computer - i.e. a *server* - from somewhere on the Internet.

W3 servers use the Hypertext Transfer Protocol (HTTP) [20], a protocol which runs efficiently enough to handle the overhead of making hypertext jumps. This is required since even small documents may use data from many sources on the network so that a number of round-trips may be required. For this reason HTTP is a simple request/respond protocol [19]. HTTP can transfer many types of data. Clients in fact send a weighted list of formats that they can handle so that the server can respond in an appropriate manner. The main type of data sent is Hypertext Markup Language (HTML) documents. HTTP is a stateless protocol in that the Transmission Control Protocol (TCP) connection is only held for the duration of one operation. This is appropriate because linked documents are equally likely to be on the same server *or* somewhere else on the Internet.

HTTP requests from clients start in the form of one of three basic methods, or operation codes; GET, PUT or POST, and are often accompanied by a Universal Resource Identifier (URI) - for example `http://info.cern.ch/hypertext/WWW/Info.html`. URIs are the network-wide unique textual addresses of objects on the Web. They consist of a prefix which identifies the protocol and encoding (e.g. `http:` for hypertext, `ftp:` for file transfer etc.) and a set of parameters which form a network address and directory (e.g. `info.cern.ch/hypertext/WWW/Info.html`). The GET method is used by all browsers when an object is desired - it must preserve the state of the Web. PUT is used when an update is made to some document and POST is used for submission of fill-out forms to some server. PUT and POST are currently not used extensively although the latter will become more important as more and more forms are used as information exchange units. When an object is transferred over the network meta-information about them (i.e. *not* information contained *in* the object) is put into a set of headers extended from the Multipurpose Internet Mail Extensions (MIME) set. This information can include: content-length, content-type, date, last-modified, expiry, etc. Unlike email where only textual information may be

exchanged, the extended headers allow transfer in binary and nonstandard formats so long as they are agreed upon by both client and server.

In summary then, HTTP is a seamless, effective protocol for hypertext and hypermedia information. The program Mosaic acts as a client to hypertext servers and clients and servers communicate using the HTTP protocol. HTTP was created because existing protocols did not provide, for example, automatic format negotiation or the ability to refer a client to another server.

3.5.2 The Nested Server Scheme

Our architecture utilizes a novel method of *nesting* the servers that play a role in the system. The HTTP server sits and waits for any requests for hypernews. If a request comes, it comes in the HTTP protocol and it requests a particular article or object from the database. The HTTP server reads the header to determine this information and then executes server-side code that effectively initiates client routines for the database. This is necessary because the HTTP server knows how to retrieve objects from the filesystem but *not* from any arbitrary database. The client routines effectively form a request to the database server and send it. The database server uses low-level routines to fetch the object from the database and pass it back to the HTTP server. The HTTP server then attaches meta-information to the object and passes it back to the user wherever he/she may be. The object is typically an HTML document so it is parsed and formatted by the browser application that the user is running (typically Mosaic).

There are critical times when the process described above does not hold exactly. The most important is when a user, running Mosaic, makes a selection that is to initiate a fully synchronized multimedia news article. When this is the case, the following takes place. The HTTP protocol passes the request for the article to the HTTP server. The HTTP server then requests *only the scenario* from the database server (i.e. *only* the timeline

or script - *not* the actual data). This scenario is passed back to the user with a particular flag in the header. When the user's Mosaic application receives data with this flag it knows to spawn a *different* application and pass the scenario to it. This application is the MediaPlayer. It is spawned, passed the scenario and it establishes parallel connections directly to the database server. In this way the required bit-rate can be achieved - the HTTP protocol could not send real-time data at a fast enough rate. This is also the case when a user requests a video browsing session with a particular video. A video browsing application is spawned which communicates directly with the database server, by-passing the HTTP server completely. There is only a need for the HTTP server when hypernews is desired.

3.5.3 Database Schema and Other Issues

The database requirements for multimedia news were analysed and some important adjustments had to be made to accomodate them. For instance, as a result of the follow-up feature where a user may ask for a list of related articles to a current article, the database schema had to include a list of key-words for every document. This is because when a follow-up is requested on a document, first the documents list of key-words is checked. Then, for each key-word the we find the names of *other* documents pointed to by them. Of course this also requires that our schema include a list of documents for each keyword. The above also holds for the profile feature so we see that the feature set impacts directly on the architecture and database schema.

Figure 3.8. The architecture and protocols of the system showing the HTTP server as a server and also as a client to the database server

3.6 Functionality and Examples

In this section we will consider three examples. In the first example an author creates a hypernews web and then stores it in a database. The second example will be a user retrieving hypernews filtered based on a profile. In the third example a student searches for information on Michael Jackson, and browses the resulting data. We explain the protocols for each in turn.

3.6.1 Example 1

Consider an author creating a hypernews web for users. The author (either a reporter or editor) uses the production server to aid him/her in this task. A production server exists at every site and is connected to the authors workstation via a network (likely Ethernet or local ATM). The author assembles all the components of the news that he/she wishes to include. This means capturing video clips, recording audio sound-bites, editing text stories

and scanning images. Once done, the hypernews web is created by making links between related media so that when a user browses, he/she is free to follow links and choose a *direction* through the information. The author then starts the recursive program which creates a new, empty document in the database, traverses his/her web and creates each object in turn and then closes the connection.

Figure 3.9. An author creating a hypernews web on the database

The recursion is requested in a message using the HTTP protocol but any and all creation requests must come from a request to the database server. Figure 3.9 illustrates the task of creating hypernews on the database.

3.6.2 Example 2

Let us suppose now that a user wishes to view the contents of a hypernews database filtered through his/her personal profile. First the user starts the front-end application which resides locally on his/her machine. Using this application a database is selected and then a profile is created or selected from existing ones. The user then selects the "View News" button and the following happens. The profile and database name are sent directly to the database server with the client-server protocol. Along with this is a flag which indicates that this is a *profile* request. The server then uses its lists and data-structures to determine which articles in the database match the profile, and then the list is passed back to the client. The front-end application formats these article names into a scrolled list and

waits for a selection. When the user selects an article, the hypernews browser is spawned and it makes a connection to the HTTP server asking for the article. The HTTP server receives the request and in turn asks the database server for the article. In this way the article is finally passed back to the user's hypernews browser where it is formatted and displayed. Figure 3.10 illustrates the semantics of these actions.

Figure 3.10. User-profiled news - a list of hits followed by article retrieval

3.6.3 Example 3

Finally, let us suppose that a student has entered a search criterion for Michael Jackson. This is accomplished at the front-end by simply entering the search criterion in a scrolled list. This criterion is sent to the database server, bypassing the HTTP server since hypernews is not involved. The database server returns a list of media objects matching the criterion - let us assume that this includes a music video by Michael Jackson. To initiate a browsing session the user simply selects this object and the video browsing application is passed the object name and is started locally. The browser then establishes a connection to the database server since it must retrieve video icons based on user interactions with the browser. Figure 3.11 illustrates.

Figure 3.11. A search followed by a video browsing session

3.7 Related Work

There are currently many systems which offer news information to the public. In this section we will discuss those of the multimedia nature, ruling out, for example, USENET newsgroups. Also, we focus on systems which have at least some editorial expertise or some affiliation with a regional newspaper or newswire that provides the expertise. Newsgroups, for example, are often unmediated uncensored forums for information; as such they may make incorrect or offensive information available. We distinguish between Internet and World-Wide Web (W3) projects and more research-oriented broadband projects which tend to be more robust and forward-thinking. There are, in fact, surprisingly few systems whose mandate is to create a full-blown architecture and system such as ours.

3.7.1 Internet and W3

There are currently many news systems which reside on the Internet. The Internet was born over 20 years ago as a United States Defense Department experiment to enable scientists to share information. Since then 40 million people from 160 countries have gained access. The software Mosaic (among others) allows easy viewing of magazine-type pages of Internet information. Almost all news data is created for W3 access in a

markup language that allows users to view pages of information that look very much like pages from a glossy magazine. Commercial Internet providers such as CompuServe and America-On-Line provide access to this independent news information. CompuServe currently hosts 200 magazines and 55 newspapers [7] so we see that there is no shortage of interest by newspapers to go on-line. The magazines on-line include everything from the high-tech *Wired Magazine* to *Time*. Many of these projects provide very timely information. For instance, in a joint venture between ESPN and Satchel Sports, sports information is put on-line for users to view at any time. Users can view standings, scores, and statistics for any of the major league sports. If viewed in the evening, partial scores of games not yet finished are displayed and updated many times per hour. Also, stories from the Associated Press are included on a daily basis. The staggering growth of W3 information and the easy access to W3 browsers such as Mosaic and Netscape make W3 projects formidable. Most of these projects, however, are weak in several areas and some boil down to nothing but glorified corporate advertisement. They all rely on the *wow* factor - that is, make the pages as glossy and as colorful as possible to keep the users' attention for as long as possible. Often, editorial skill is compromised for flashy graphics and icons. For instance, the Buena Vista Movieplex service presumably offers information about movies distributed by this company (such as the Lion King, etc.) However, the service is rendered very slow since it consists mainly of large images from the movies and very little other types information about plot or production notes etc. and thus to some extent resembles a child's book. Further, few of these projects include archiving of any kind; that is, last week's news is gone and cannot be browsed. For instance, the Globe and Mail offers information on-line but only special stories and does not offer access to anything except that week's special stories. Few of these projects offer any kind of personalization factor; that is, the news is the same no matter who you are or what you like. Finally, tools such as searching are generally not offered at all.

3.7.2 Other News Systems

Jacobs et al describe their System for Conceptual Information Summarization, Organization, and Retrieval (SCISOR), a system which uses natural language text processing to extract information from USENET news [8]. News stories can be processed at about six stories per minute and to accomplish the extraction the following tasks are performed: the character stream is lexically analysed, the news is separated into components, the topic is determined, the story is analysed, the information is placed into a knowledge base. SCISOR allows viewer to browse headlines or ask free-form questions such as, "What was offered for Polaroid?". Since SCISOR does not really deal with multimedia information it does not address the user interface and focuses on natural language and knowledge bases. It proposes a very primitive user interface and relies strictly on USENET as a pseudo-database of information - hence there is no real editorial expertise nor is there a mandate for news creation or tools.

Ohkubo et al describe a system called the Information Skimming Space (ISS) that serves as a window through which news articles may be acquired and viewed [6]. They emphasize enhancing the efficiency of the *presentation* of large amounts of news information in order to make cognitive understanding and acquisition easier and faster. They focus on the initial presentation of a large amount of news articles and the provision of navigational paths to find other articles deemed interesting by some metric. News reports consist of the following parts; presentation, content/keywords, profile, and logical structure. Keywords and summaries are used extensively to facilitate quick article location. While the ISS is an interesting achievement in graphical layout of news and of intelligent article selection, it does not address database issues, media synchronization or how and where news is to be produced or stored.

Koons et al have created an information system called the Computer Sciences Electronic Magazine (CSEM) using elements from computers, televisions, and printed

documents to form a new communications medium [3]. They started with a magazine metaphor and adapted it to multimedia. The structure of magazine content was discovered by examining many magazines. CSEM looks like a magazine, acts like a VCR, is simple and gives feedback to users. They also performed usability experiments. The experiments showed that CSEM failed when it did not remain true enough to the magazine metaphor. Problems such as a lack of table of contents and screens which users thought were too busy made information gathering tasks more difficult. While CSEM provides good background to what users want in multimedia applications such as news, it focuses mainly on the graphical design and interface of the system. It does not address the architecture required for such an application, the production of multimedia data, or the storing of news or magazine data into a repository.

Hoffert et al describe EDUCOM, an on-line news system which features hypertext-type navigational functionality [1]. The goal of EDUCOM is to let users drive the news, bypassing items that do not interest them. Also, the ability to zoom into more detail in a particular area is available. Users can click on hyperlinks to view another part of the news. EDUCOM begins to address the challenges of producing electronic news and of delivering the news over a network. EDUCOM does not, however, address media synchronization or the database issues of storing news. Also, what EDUCOM calls user-driven news is simply a user making appropriate selections - there is no concept of user-profiling or modeling.

Lippman et al have proposed Newspeek, an *intelligent* application which gathers and presents daily news information from remote sites based on reader profiles and presentation preferences [4]. In this system news broadcasts are directed at a computational agent acting on the user's behalf. This agent then edits the incoming material and presents it in a way that the user will appreciate. Newspeek helps readers interactively explore a large news database in a non-directed manner. Articles and information are layed-out in a front-page metaphor, closely mimicking the front-page of North American newspapers - important stories are big, related stories are off in the corner,

etc. Touch screens and gestural input drive browsing in Newspeak; for instance a horizontal gesture across an article brings that article forward so it may be read. Although the idea of wide-spread distribution of touch screens is probably not feasible in the near future, Newspeak is an ambitious project that lays down the ground-work for many important issues in multimedia news. It does not, however, discuss document or database architecture at all.

Finally, Miller et al describe a news system involving Dow Jones and NYNEX over wideband networks. The system consists of a production center, a media service center, and user sites, featuring news alerts, updates, archiving, profiles, and distribution control [5]. The NYNEX system has the capability to multicast live video feeds to customers for video information which is too critical to delay. Live video is fed to a bandwidth allocator which allocate channels according to the needs of the video. Customers then receive this data at their sites (see figure 3.12).

Figure 3.12. Simplified architecture of the Dow-Jones Nynex System incorporating live video feeds.

This system is an excellent example of how both technology and know-how have evolved since the creation of some of the above-mentioned projects - it shares many goals with our system.

3.8 Examination of MULTINODES Architecture

Our architecture and utilization of the MEDIABASE platform makes MULTINODES quite unique. At the conception of this project not only did very few full-blown news architectures exist but few were supported by the technology available to our system.

The use of the HTTP server as a way into our multimedia database was novel at the time of conception of this thesis. Since that time, the explosion of W3 services and the huge amount of attention focused on the W3 have meant that other corporations and institutions use similar techniques to allow remote users to access their databases. These databases are primarily relational - often Oracle™. Further, some systems on the W3 now allow remote users to perform Structured Query Language (SQL) queries on the databases. It should be noted that the introduction of the HTTP protocol into MULTINODES helped bring the system to life very quickly since it allowed for the use of W3 browsers to view news and request articles. It is not the best solution in terms of efficiency though. We will show that it would be better to build a custom browser that can run more efficiently over the ATM protocol and take advantage of the greater bandwidths.

The distributed architecture of MULTINODES is one that we think will help define news systems of the future. Video conferencing to exploit the network bandwidth and remote document creation are all small but important parts of the whole. In the next chapter we discuss the problems associated with video representation and browsing and introduce our original tool to solve these problems.

Chapter 4

Representing and Browsing Videos - The Video-tile

One thing that is certain is that multimedia news articles will and should make extensive use of video. Whether it is news reports, sports highlights, video classifieds, or movie trailers, digital video objects will be available to users. Video is appearing more and more in other applications too - video conferencing, video-on-demand, video editing, interactive course-ware, geographic information systems, to name just a few, all include video streams in one way or another. But in applications where users are given the power to browse and manipulate these streams, if poor representation techniques and browsing capabilities are offered then the effectiveness of the application can be diminished. VCR-type browsing features such as play, stop, rewind, etc. are not considered effective for non-sequential browsing. Since the purpose of an application is to empower users with information, it is vital that the video tool not impede this process and that the representation, browsing, and manipulation methods map well into the application's context in order to maximize information transfer while minimizing cognitive overhead and time.

There are two main issues or problems to consider when dealing with video streams. One is how to graphically represent the stream as a whole or in part to the user, and the other is how to support non-sequential content-based browsing through the stream. As of the writing of this thesis, research in other laboratories had yet to provide definitive solutions to these problems or yield a browser that is effective on all counts. The remainder of this chapter then is organized as follows: first we discuss the need for video tools - particularly in the news context, then we discuss the two issues mentioned above and other related research in these areas. Next we introduce the tool we have devised to address both of these issues and argue that it maps well into the news context and into other contexts as well, emphasizing the visual and qualitative properties of the tool.

4.1 The Need for Video Tools

In a news system such as the one described in previous chapters, it is clear that the presence of tools supporting media retrieval and browsing is necessary. The type of media that typically has the least amount of good user-support is video. Research has paid a relatively large amount of attention to the search for effective and efficient search and query techniques for textual databases. More recently, image databases are becoming supported with robust query engines supported by pattern-matching and shape recognition which allow users to, for example, provide an image of a car and demand, "Give me all other images that are similar to this one" [40][41]. Video, on the other hand, is different than text and images because it is continuous; by this we mean that it not only has a content dimension, but also a temporal dimension. Frame-rate and duration are two temporal qualities that are unique to video; images and text do not have analogous qualities. Querying and browsing video is difficult, mainly because video content is temporal, possibly vague, and very difficult to specify textually. It has been shown that if two users are asked to label a video with a number of keywords that these words seldom match up

on more than a few counts [33][42]. Also, video content can be thought of as comprised of both *content data* and *camera data* where content data refers to what is happening in the shot, and camera data refers to how the camera itself is being manipulated in the shot - e.g. wide-shot, close-up etc. It is important to note that the way that these two types of data are combined can create different meanings and that visual indices can indirectly supply camera operation information. This is because the first frame of camera shots often makes the camera data self-evident. In the context of our news system we saw the need to supply both users *and* authors with video tools. Users will benefit from them in the following way: suppose that a student, doing a project on prime minister Chretien's tour of Paris, has a session with our system. She wishes to search through the archived news videos for a segment in which Chretien is delivering a speech and the Eiffel tower is visible in the background. How to do this? Keywords are not sufficient here since it is not likely that all items in the background will be identified - particularly if they are not integral parts of the content. At best the keywords *Chretien* and *Paris* will be associated with the clip, but not *Eiffel*. The best way to find the clip is to use some kind of visual index into the archive.

A second way of illustrating the need for video tools is to consider authors at production centers. When authors are assembling multimedia documents they need to query the content server to find and select appropriate media objects. When retrieving images or text, it is typically not a problem to retrieve and then display the object to verify its appropriateness. When considering video streams that may be up to 10 minutes in duration, it would slow down the entire process if authors had to view the whole video stream to judge whether or not it is relevant. Further, since a keyword approach to video is sketchy at best, one needs to use a visual index into the video stream that allows authors to quickly grasp the content and temporal nature of the stream. Figure 4.1 summarizes what is needed.

Figure 4.1: Accessing archived video streams using a graphical interface and specialized tools

Finally, to illustrate the need for a tool that allows non-linearity of motion within a video stream consider a video of a one hour soccer match that ended in a 1-0 score. Suppose someone who has not yet seen the match wishes to find and watch the only goal. Using only their VCR remote, how to do this? Since the VCR is a linear device the user would have to start from the beginning and begin scanning forward using the shuttle button, or, continually use the fast-forward button while hoping not to skip over the goal. This may take quite some time and in the worst case where the goal occurs in the final minutes, it will take about twenty minutes (assuming a 3-times shuttle speed). At best it will be a tedious and frustrating experience.

4.2 Representation - Visual Indices to Video

Let us first consider some clearly *inadequate* ways to represent video streams. One could simply represent the stream with a generic icon, meant for all objects of type *video*. Clearly the only piece of information that this provides to users is the type of media. Another poor way of representing the stream is to apply a title under the icon, much like a Macintosh icon. Again, this reveals very little about the content. Consider an icon labeled "New York street". Although one knows the media type of the object, this label does not answer the most basic questions such as, "What time of day?", "What season?", "How much traffic?", or "How long is this video stream?". The problem with *textual* indices into a *visual and temporal* medium is easy to see.

Now consider a method of representing video streams with a key frame. Often the key frame is the first (or often the tenth) frame of the sequence. The problem with this is two-fold. One, if the stream begins with some special effect whose duration is one second (30 frames), like a fade-in, the first and tenth frames will both be quite black. Furthermore, in a typical news story if the stream consists of a number of camera shots, the first shot typically sets a general context for the story (e.g. an aerial shot, or a skyline) and is not revealing in terms of the main thrust of the story.

Intelligently selecting a key frame from the stream can be expressive, but in doing so one takes a snapshot of the action at one moment in time; thus the temporal dimension of the stream is flattened. It would be much better to attempt to leave behind more of this temporal dimension to aid the user in visualizing the entire stream. Furthermore, automating this process is very difficult and selecting key frames manually is labor intensive. The following section briefly describes some of the most interesting research in the area of video representation.

4.2.1 Related Research

Zhang et al have developed algorithms to automatically parse television news and create 3D visual indices often called *micons* [31] (also proposed by Mackay et al [25]). Micons are iconic cubes with 3 faces showing; the front face shows an iconic version of the indexed video frame and the top and side faces are striped in different colors representing the presence of camera breaks in the video. The depth, or thickness, of the micon is proportional to the relative length of the video.

In a similar approach, in [32] the top and edge faces of the micon consist of the pixels from the top and right edges of each of the frames in the video, thus creating a stack effect and providing what the authors call the visual rhythm of the video. The micon can therefore convey content and temporal information.

Ueda et al use an array of micons in a video editing context. A large amount of video data is represented by using a matrix of such micons, each representing a scene or camera cut withing the movie [30].

In a similar approach, Cruz et al present a stack of frames called *video streamers* [34] as a part of a system meant for capturing multimedia information from video conferencing sessions and meetings. They also apply the stack concept to audio streams and to the user interface itself to show how it has changed over time. Video streamers are meant for a video conferencing application and make it easier to, once having captured a conference, go back to find and replay particular events that happened during the conference.

A different approach is taken by Teodosio et al [28]. By pre-processing videos and using image processing algorithms, they create what they call *salient stills* - single graphical images as indices to videos. The salient still reflects temporal changes that occurred in the video stream and preserves critical aspects of the content. It may feature any or all of: variable resolution patches, composited objects from the scene, or a field of view greater than that of any single frame in the video; it is constructed from the sum of all of the frames in the video. Hence the salient still is an image which does not appear as a single frame at any point in the stream. An example given in [28] is a salient still which represents a solo violin concert video and in which what looks like a ghost image of a person is present. This ghost image is the result of a stage-hand walking across the stage at one point during the presentation.

Tonomura et al developed the *VideoSpaceIcon* to represent spatial and temporal aspects of video [29]. By first detecting the camera operations in a video segment using image processing techniques, the VideoSpaceIcon can be built, allowing viewers to grasp the physical space of the segment at a glance. Incorporated as well are X-Ray images and camera movements are made apparent through icon shape.

Finally, Arman et al create *Rframes*, or representative frames, to display to users [35]. An Rframe is used to represent a video shot and consists of a central body, motion tracking regions, and time indicators. The central body is a key frame from the sequence, the motion tracking regions are around the perimeter and serve to reveal camera and object movement in the shot, and the time indicators are small representative squares which give a rough indication of the length of the shot.

Some of these methods of video representation may also be used directly for browsing through the stream but most may not. The next section introduces the concept of video browsing and discusses related research.

4.3 Video Browsing

Let us say that a user wishes to browse a video and assume that the video has been indexed at some granularity. Here, browsing is distinguished from passively sitting and watching the video as you would a television show. We take it to mean intelligently interacting with the video. Typically, basic VCR-type features comprise interaction capabilities (e.g. play, stop, pause, rewind, fast-forward). But users should be able to do more; for instance replay video, browse scenes, play at variable speeds and directions, and have quick, easy, random access to the smallest granularity of video that is indexed - e.g. a shot, a frame, etc. To understand why, recall the soccer match example given previously.

In order to discuss the type of access that users have, it is useful to discuss what we can call the *basic browsing unit* [35]. The browsing unit is important in the sense that it defines the granularity of browsing actions. For instance, a browsing unit of a scene indicates that the video is seen as being composed of a sequence of main scenes. On the other hand, a browsing unit of a frame means that the video is seen as being composed of individual frames which may be accessed randomly. The following section briefly outlines some recent research in digital video browsing.

4.3.1 Related Research

As mentioned earlier, the micon may be used for browsing purposes. An interactive micon can be equipped with a cursor bar which moves along the side face (backwards and forwards in time). In an approach used in [32], frames or camera shots, depending on the type of micon, may be selected and played by simply moving the micon cursor bar along the edge to the appropriate spot and clicking. Similarly, browsing through video streams is accomplished in [34] by simply selecting a frame from the stack. However, none of these approaches would be considered iterative - that is, there is no repetition which allows users to delve into lower levels of granularity.

Mills et al solved this problem with their video magnifier tool [27]. The video magnifier allows the user to recursively magnify the granularity of the current view of a video while preserving a global context. By global context we mean the answer to the question, "What am I looking at with respect to the video as a whole?". Earlier work by Furnas has addressed this problem with *fish-eye* views of data [24], as has recent work by Mackinlay et al who present data graphically on a long folded "wall" using perspective transformations [26], and in a companion paper, present hierarchic information graphically in 3D trees of cones called *cone-trees*. The video magnifier produces a hierarchic view of the video with smaller and smaller levels of granularity. Also, the upper levels of the hierarchy remain displayed while browsing at a lower level. This allows the user to visualize his/her current position in the video with respect to the video as a whole, thus preserving global context.

In [35], the video stream is represented by a linear row of Rframes. Rframes are representative frames which composite data directly from the stream as well as motion tracking information which reveals camera and temporal information. Limited browsing can be achieved by scrolling the row or by selecting an Rframe and seeing a result-row of

Rframes. In the result row, frames which are very similar to the target are shown as normal, those which are somewhat similar are squished on the x-axis by 67%, and those which are non-similar are shown squished by 95%.

The difficulty of creating good browsing tools means that each of the above mentioned tools have their inadequacies. For instance, the Rframe approach uses an interesting representation but then forces users to browse based on similarity measures, a method that may not be appropriate in many cases, particularly if the user has yet to see the video. The stack of frames approach is limiting because the value of revealing the edge pixels along the stack is debatable and limited screen real-estate means that only a small stack of frames can be displayed at one time which defeats the purpose. The video magnifier is powerful but yet does use a basic browsing unit based on camera information, rather it samples frames evenly from the stream. In this way camera information goes unnoticed and as we have discussed, this may mean real content information also goes unnoticed.

4.3.2 Determining Camera Operations

Discovering camera operations after the fact is crucial to video tools which offer a basic browsing unit that depends upon camera-shots. Hence any system which can offer scene browsing or shot browsing needs to be able to first detect these scenes and shots. Much research has been done in this field [6][31][32] and we offer a brief summary in this section.

First let us define a camera-shot, or cut, as that group of frames that were captured continuously during the filming process - the camera was not turned off at any point. That is, a camera-shot has a camera-break at each end. The basic premise of segmenting video streams into cuts is that frames on opposite sides of a camera-break have radically different content (based on some appropriate metric). Thus, detecting the cuts is a matter of

comparing adjacent frames for significant content changes. One way of doing this is by making pair-wise comparisons of pixel intensity values where two pixels' intensity is said to be different if their difference exceeds some threshold. Then, adjacent frames are said to be different if the number of pixels that are deemed different also exceeds some threshold. A second way of detecting cuts is to compare intensity or color histograms. Making cut detection more difficult is the possible presence of special effects serving as transitions. These include fade-ins and outs, wipes, and dissolves.

Determining camera operations such as pans and zooms can be accomplished in similar ways; one finds motion vectors which reveal these operations. For instance, by examining blocks of pixels in groups of frames it is possible to compute a vector indicating how the positions of the blocks changed through the group. Then, using suitable thresholds, it can be concluded that in general camera pans are revealed as horizontal vectors, camera zooms are revealed as radial vectors, and so on.

4.4 A Novel Tool - the Video-tile

By examining the attempts of other researchers we were able to devise a unique and highly effective tool to both represent and interactively browse video streams. We wanted the tool to be compact and intuitive and to reveal the maximum amount of information as possible - accepting the fact that video streams are content heavy. In summary, the tool had to:

- be content-based
- be compact (that is, not take up the entire computer screen)
- be iterative (hence interactive)
- use the camera cut as a basic browsing unit
- allow viewers to instantaneously build a rough cognitive model of the video

When large amounts of data must be displayed on computer screens, often the limited real-estate of the physical screen forces data to be shown in such a small scale that it

is illegible. Consider a detailed map of Europe - fitting it onto the screen would make all but the capital city names too small to read. Other alternatives exist however. Zooming into a region is effective for displaying the local detail but the overall structure is lost. Using two windows to integrate context and detail can work but uses extra screen space and creates more cognitive load for users [36].

In [24] and [36] researchers have developed what are called fisheye views. This is a method of displaying a large amount information on the screen, including detail *and* global context. Fisheye views contain a focus which is the point on the view that is displayed in the most detail. The remainder of the view is displayed in less and less detail, depending on its proximity to the focus, its importance, and possibly other user-parameters.

Figure 4.2. A graph and a possible fisheye-type view of the same graph, focusing on node D

Figure 4.2 shows a graph and a possible corresponding fisheye representation. Note that since the focus is on node D, nodes which are further in the Euclidean sense from D are displayed correspondingly smaller. The fisheye view can be an effective for viewing 2-dimensional graphs and maps. For instance, often in complex hypertext systems, users complain of "getting lost" in the information web. As a result, node-link maps may be used as aids but the maps themselves are often so big that users can even become lost in them! Hypertext is one application where these views are effective. What about video

streams which are linear in the temporal dimension? Figure 4.3 shows the only way to apply the fisheye view to a linear stream. The focus frame is displayed in detail neighbor frames are in less and less detail depending on their temporal distance from the focus.

video stream (10 frames)

Figure 4.3. A video stream and a fisheye-type view of the same stream, focusing on frame 6

Figure 4.4. An insecteye

In the conception of video-tiles, we shifted slightly away from the fisheye view towards what we call an insecteye view. Insects like flies have what are called *compound eyes*. This means that unlike human eyes, their eyes consist of many small eyes, organized together into a large (often honeycomb) bundle. In this way, the insect sees a much greater panorama than do humans - figure 4.4 illustrates what is meant. In the next section we explain how we utilize a view such as this.

• **Description:** The basic premise of video-tiles is to allow hierarchic browsing through various levels of granularity in a way similar to the video magnifier. The shape of a video-tile is rectangular and is composed of processed frames of video of varying size. In the center of the tile is a video keyframe which represents the current focus. Initially this could be the median frame, the first frame, or even a salient still of the video - we will call this level 1. Along the perimeter outside of this center frame (in contact with its edges) there is a *ring* of level 2 frames each of whose length and width is half that of the center frame (one quarter of the area). This level has 12 frames which represent a finer grain view of the video. Where the video was represented by a single frame at level 1, it is represented by 12

frames at level 2. Similarly, level 3 can be a ring of 36 frames surrounding level 2, of half the size of those at level 2 (see figure 4.5).

Figure 4.5. The basic shape of the video-tile and temporal ordering of sub-tiles

We have seen that even up to level 3 or 4 that the sub-tiles show enough data to convey *some* meaningful content information. We find, however, that economically level 2 is a good compromise. As a part of the interface the video-tile displays two scales (see figure 4.5). One indicates the current focus (e.g. the scale position in the bar is analogous to the current focus in the video stream) and the other the domain of frames represented in this iteration (e.g. the length of the bar is proportional to the number of frames in the domain); hence, global context is supplied. Temporal information is also revealed. The sub-tiles in level 2 are ordered temporally - we think of time as starting from the NW corner and flowing clockwise. Thus in the initial iteration, the level 2 icons tell the entire story of the video stream in a condensed way. At any level, the sub-tiles are chosen so that they are as equally distributed throughout the current domain as possible.

- **Basic Browsing Unit:** The basic browsing unit is the camera cut. This means that sub-tiles represent the first frames after every camera cut in the video. We have found that in the news context, this is the most informative unit.

- **Interaction:** The video-tile is interactive. That is, by selecting any one of the represented cuts in the tile, a new tile is produced that now features the selected cut as the level 1 sub-tile. If we assume that the first video-tile drawn has a domain of frames equal to the number of frames in the video then the second tile drawn (after the first click) will have a

domain equal to some subset of previous domain, say one half of it. The new domain is centred around this just selected cut and the outer rings of this tile span only the new domain of frames. This process of selection and zooming may iterate and in this way the user continually zooms into more and more detail. The number of levels in the tile may be changed at any time, thus the user can *grow or shrink* the tile to see a greater or fewer number of representative cuts per iteration. *Zooming out* is also possible with the tool. By selecting the appropriate button, the current domain of frames doubles (as opposed to halving), the current focus remains fixed, and a new tile is drawn. In this way the user effectively *takes a step back* to get a wider perspective on the current focus point in the video stream.

Figure 4.6 illustrates the semantics of a single browse event, namely a selection of sub-tile 6. Tile 1 represents a 3 minute video consisting of 5400 frames. The numbers represent the numbers of the actual frames sampled to create the sub-tiles. The user has made a selection on the sub-tile which was sampled from frame 2109. The information for the next tile is now sent over the network and assembled into tile 2. Tile 2 shows frame 2109 as the level 1 frame and represents a region of 1350 frames *around* frame 2109 as sub-tiles on level 2. Thus the domain of frames is halved to 2700, as is the duration of the stream being represented to 1 minute 30 seconds.

Figure 4.6. Browsing with video-tiles

The video-tile has advantages over all other analogous video browsers - that is, it is compact and reveals content and temporal information graphically. We feel that it is an

effective tool for video, particularly in the news context. Figures 4.7-4.9 illustrate a browsing session wherein a user zooms into a particular spot of an animated cartoon. For this test case, the basic browsing unit is *not* the camera cut but rather groups of frames which need not have camera cuts at both ends (although every camera cut belongs to a group). Note that in the first tile the cartoon is represented by twelve sub-tiles at level 2 and the focus is at the first frame which happens to be the title of the cartoon. These level 2 sub-tiles tell the complete story in a condensed visual way. Note also that the level 2 sub-tiles become more and more similar as we zoom in - this is because of the increased granularity.

Figure 4.7 Initial tile. Focus is on first frame but the entire story is represented as sub-tiles at level 2

Figure 4.8. Focusing on the part of the story where the cat is blown out of the hatch - i.e. sub-tile 6 from Figure 4.7

Figure 4.9. Zooming in further into this part of the story

In the context of multimedia news, we expect that television news stories in particular will be integrated into the system so that video reports and clips from TV news will be media objects within multimedia news articles. By video reports we mean the kind of news story filed by, say, a foreign correspondent or a CNN reporter. What is the structure of these video reports and will they map well into our video browser? We believe that they will based on their innate temporal structure. By watching a number of these reports it is easy to find a generic temporal structure into which most fall. Figure 4.10 illustrates the structure.

Figure 4.10. Temporal structure of a news *video story*

The first shot is usually a context shot with the reporters voice-over. It sets the scene for the story and may consist of several camera shots. Next we see the reporter as she tells us the story. Then, again with voice-over, a series of action shots. Finally we see the reporter again as she summarizes and signs off (e.g. "Susan Smith, CNN News, Budapest."). It is not hard to see that by simply taking the first frames of all the camera cuts, the news video story is represented compactly and in a very informative way. Alternatively, the entire news broadcast (e.g. CNN Headline News) has a broad temporal structure along the lines of: anchor-person shot, video story, anchor-person shot, etc., commercial, weather shot, sports shot, etc. We could map the entire broadcast into the video-tile very effectively using a basic browsing unit of the camera cut.

In some video tools, such as the hierarchic video browser, representative frames are selected from the stream because of their position in the stream, for example they are chosen because they are the 100th, 200th, 300th etc. frames. In other systems, such as

ours, frames are chosen because they represent camera transitions or breaks. We chose to use camera breaks since in the news context they segment the streams into more meaningful sub-streams, or groups. Figure 4.11 illustrates the frame selection procedure.

Choose the next frame as a representative icon

Figure 4.11. Choosing frames for the video browser. Frames following camera cuts are selected to represent that series of frames between it and the next cut

We will now show a second video being browsed using the video-tiles. Again, this video's representative icons were selected evenly spaced out from the video, and also because this video *did not have any camera cuts at all*. This video is a one minute clip of a part of a Michael Jackson music video in which a series of people are shown, each *morphing* smoothly into the next. The clip relies heavily on this special effect and is very linear since each person is morphed from the previous person. Figures 4.12 through 4.14 show a user choosing a particular part of the video and zooming into its detail.

Figure 4.12. Initial tile. Focus is on first frame but the entire video is represented as sub-tiles at level 2

Figure 4.13. Focusing in on a particular part of the video

Figure 4.14. Zooming in further into this part of the video

4.5 Video-tile as a Tool for News Scenario Visualization and a Generic Tool for Temporal Visual Data

We would like to argue that our content-based browser for news videos is in fact an effective browser for multimedia news scenarios and for temporal data from *other* domains. Because of the unique view of data that it takes and displays, its interactivity and its simplicity, it can be effective for these other types of data.

Recall that a multimedia news article is authored on what is ultimately a time-line. Suppose we have a time-line as in figure 4.15.

Figure 4.15. A simple scenario for a multimedia news article

To map the scenario information to the video-tile we do the following: first, at authoring time the author must associate a key-word with every text object and designate a key-word or key-image for the entire scenario. Next, after the scenario is assembled, all the images used in the scenario are sampled and iconified, and key-frames are extracted automatically from video objects used in the scenario. The tile that results from the scenario in figure 4.15 is shown in figure 4.16. Compare this to the timeline in figure 4.15 to see the difference in information revelation.

Figure 4.16. A video-tile-type layout for a multimedia news scenario

The three text objects used are represented on sub-tiles by the key-words, "Clinton", "Taxes", and "Stocks", thus giving us a hint as to what the text objects present. The presence of the audio object is signified with the speaker icon, and the video and image objects are signified by representative frames. Again, time flows clockwise from the NW corner so that by scanning in that direction the user comprehends the scenario. In a more complicated scenario, one could click on a sub-tile and zoom into that part of the scenario in more detail just as with video browsing - further, clicking on the "play" button will play the scenario commencing at the current focus. Representing scenarios in such a fashion allows both users and authors to peruse scenarios non-sequentially, determine their relevance, and to start the scenario from any point.

During the creation of this tool, we also sought feedback from a group of filmmakers from a small independent filmmaking company. They described digital video editing software to us and in particular, emphasized that even the best editors represent cuts using a simple icon captured from the first frame of the cut, and how this has caused them frustration in the past. It is even more glaring when one realizes that most of the time *the film editor and filmmaker are not the same person!* Should we expect the first frame of a cut to reveal much information to an editor who has absolutely no other information about the cut? They felt that the video-tile was a good way to represent a cut and we showed them a modified video-tile which could not only represent a series of cuts but also reveal camera information such as; close-up, wide-shot, zoom-shot, pan etc. This is achieved by overlaying special symbols onto sub-tiles which reveal camera information. Figure 4.17 shows the camera symbols overlayed onto blank sub-tiles which reveal, at a glance, that the groups of frames represented by these sub-tiles have (in order); a camera pan to the right, a wide-angle shot, a zoom-in shot.

Figure 4.17. Sub-tile overlays which reveal camera information; a pan to the right, a wide angle shot and a zoom-in shot

Overlaying camera information on all the sub-tiles allows filmmakers and editors to review the shots and instantaneously grasp camera information (and hence the semantics of the shot) more easily. Camera data is often a crucial part of creating emotional reactions in films and should therefore be used as a visual index into it, as we have illustrated here.

By overlaying other information onto sub-tiles the video-tile becomes a very generic and useful tool. For instance, overlaying several types of information is a technique that is often used in geographic information systems (GIS). Applying overlays in similar ways can make video-tiles effective in a GIS setting also.

4.6 Remarks

Representing and browsing video streams is indeed a challenge. We have created a novel tool called the video-tile which addresses many of the concerns traditionally voiced by researchers - it has two scroll bars which indicate current focus and current domain, thus it provides users with global context so that they do not become *lost* in the video. The video-tile is compact - it does not take up a lot of screen real-estate. It is also graphical and reveals content directly from the original stream and does not attempt an inappropriate mapping such as creating textual indices to the video. The indices are strictly graphical and the interface is simple and it is easy to use - users need only click on the sub-tile where they wish to zoom. Video-tiles can be a generic tool for data from filmmaking to multimedia news to geographic information systems. In the next chapter we discuss the implementation and results.

2 of/de 2

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

PRECISIONSM RESOLUTION TARGETS

Chapter 5

Implementation and Results

In order to transmit or receive information of any kind to or from a user of a computer system, some kind of human-machine interface is necessary. The interface could be a simple text window or something more extravagant. A system such as MULTINODES requires an interface which can accomodate the manipulation and transmission of large amounts of user-data.

In this chapter we provide screen-shots of some sample sessions on MULTINODES. Next we summarize the concepts of building graphical user interfaces with an object-oriented application builder such as Motif, making specific reference to our interfaces and then we discuss some user reactions to our system which we have demonstrated at numerous conferences and exhibitions. We then give the results of a bitrate test that we performed, testing the performance of running TCP/IP on the OCRInet. Finally we discuss some implementation aspects of the video-tiles video browser.

5.1 Sample Multimedia News Sessions

In this section we attempt to make more clear the functionality and tools that we have implemented by presenting a series of snapshots of the system in use. For each we describe what is being depicted and explain the functionality if necessary.

Figure 5.1 shows the user interface front-end to the news database. This is the window that users are presented with initially. The buttons along the bottom of the interface allow the user to: (in order) search the database, initiate a video browsing session, manipulate the personal scrapbook, view the key-word tree for this database, follow-up the current article, get help, or quit. Of course the main use of this interface is for making a document selection and viewing the news. This is accomplished by using the *Document* button or text field and then the *Daily News* button. Figure 5.2 shows a user creating a new user profile. It is being called *classical.pro* and consists of four keywords. This profile can serve as a filter between the user and the news database.

Figure 5.1. The news application's front-end

Enter Keywords

Bach
Handel
orchestra
symphony

Profile Name
classical.pro

OK Cancel

Figure 5.2. Creating a custom profile

Figure 5.3a. Example of a news article from the hypernews database

Figure 5.3b. Example of a news article from the hypernews database

Figures 5.3a and 5.3b shows how hypernews articles appear to the user. Note that users may read, scroll, or click on links to browse the hypernews.

Figure 5.4 shows a user initiating a search. The user is apparently searching for articles regarding baseball or hockey or one that specifically relates information about a minor league team; the Hull Olympiques. When *OK* is clicked the search is performed. The resulting hits are returned in a list box. Figure 5.5 illustrates this list. Documents have been found that match one or more of the keywords. Selecting this document loads and displays it. This is shown in figure 5.5.

Figure 5.4. Entering a search criterion

Figure 5.5. A list of documents matching the search criterion

Figure 5.6. Viewing the document which matched the search criterion

Once the document is retrieved and read, related articles may be retrieved by clicking on the follow-up button. Figure 5.7. shows the result of selecting a document from the follow-up list from the article shown in figure 5.6.

Figure 5.7. Following-up the current document

Viewing the key-word tree can be done at any time. Figure 5.8 shows what the keyword tree looks like mapped into a scrolling text-box.

Figure 5.8. The key-word tree for the database

Figure 5.9. Viewing the contents of a scrapbook

If the user has previously added some articles to his/her scrapbook, he/she may want to review them. This is accomplished by clicking on the scrapbook icon and then selecting any of the items therein. Figure 5.9 shows a user viewing a scrapbook listing.

5.2 Graphical User-Interface Construction

In chapters 3 and 4 we have introduced both the front-end interface to the multimedia news data and the video-tiles interface for browsing video streams. We discussed in detail their functionality within the news system. In this section we discuss the general concepts of object-oriented graphical user-interfaces (GUI's) and how they are built.

The term object-orientation refers in part to a process of reuse and specialization of data. Applied to the concept of programming, this means that much time may be saved by borrowing and adapting code that has already been created and debugged. In GUI programming, all user-interface components (called *widgets*) such as pushbuttons, scroll bars, etc. belong to a certain class of widgets. The widget itself has a set of attributes, functions and resources that are specific to it and the class it belongs to also has attributes and resources which every widget in that class inherits. There is, in fact, a hierarchy of widget classes the root of which is called *Primitive* and is the ancestor of all other classes [37].

A second type of hierarchy is created when the programmer begins to actually build his/her GUI. When a widget is needed, such as a push button, an instantiation of that widget class is created. In this way, an *instance hierarchy* is created by the application builder. Each instance has a parent widget which controls the layout of its child widgets. For instance, a parent widget of the class *Form* allows children to be layed out in any arbitrary way whereas a parent of class *RowColumn* allows children to be layed out only in either rows or columns.

Our news application's front-end has a complex heirarchy of instances. Figure 5.10 shows the interface and figure 5.11 shows the simplified instance-hierarchy. Note that in order to create, for example, the Daily News push button in the location that we wanted it that is was necessary to first instantiate three RowColumn widgets. This is because the higher the instance-tree the more layout capabilities are inherited and thus the more control the programmer has to place widgets where they are needed.

Figure 5.10. The news application's front-end interface

Figure 5.11. The instance-hierarchy for the front-end (simplified)

5.3 User Reactions

We have had the opportunity to exhibit our system at a number of conferences and exhibitions in the Ottawa region and beyond and user response to our system has been mostly positive. At the Central Canada Exhibition in Ottawa we had a demonstration booth where people could come to interact with the system. Feedback was positive, the most common question being, "When will the system be available to Internet users?". A concern of the public was the type of billing. Would the billing be by time or by the amount of content that one asks for? Generally, people were happy with the concept of the user profile and impressed with the notion of updates that occurred more than once per day. Several users asked if television news could be retrieved and watched through the interface. Finally, although some users took a moment to grasp the concept of textual hyperlinks, most had no problem browsing and enjoying the hypernews.

It should be noted that systems such as this one which are relatively new to users are difficult to evaluate based on user reactions alone. This is because many people, though

uneducated about computer technology, are overexposed to information from the media about what they should expect from such systems. Television shows and reports can confuse viewers about what the current state of technology is and what can and cannot be accomplished with it.

We have also approached filmmakers for their input and opinions. We have worked with a small, independent filmmaking company in Toronto and have given them demonstrations of the news system and in particular the video-tile video browsing tool. They have a combined knowledge of all areas of filmmaking; writing, directing, producing and editing. They unanimously agreed that the tool is interesting and effective and could be very useful in the area of film editing. This is because when a film is made the first step is usually to complete the shooting - i.e. the collection of all data onto film. This is done under the supervision of the director. After this stage, a large amount of raw footage is available and it is the editor's job to cut and assemble the footage to create the finished film. Of issue here is that often the editor is not the same person as the director so that where the director has intimate knowledge of each camera shot the editor does not. This can be a real stumbling block since in almost all commercial video editing systems camera shots are represented by the first frame of the shot which may not convey much information to the editor. The video-tile can provide a more revealing representation of camera shots so that even an editor who was not involved with the shooting of the film can quickly grasp camera shots and complete his/her job.

5.4 Some Test Results

During a demonstration at the Central Canadian Exhibition (CCE), we conducted an experiment to test the bitrate that we were achieving using Mosaic as a hypermedia browser. We had a 15 Mbps virtual ATM channel between the CCE site and the database server in the laboratory at the University of Ottawa. We suspected that by using Mosaic

and the HTTP server to request documents over the OCRInet that we would waste much of the bandwidth allocated to us. We transferred several media objects by requesting, via the HTTP server, the object, and then awaiting their complete arrival. The results are shown in table 2 (column 4). These results emphasize the importance of eventually building a browser which runs natively on the ATM protocol and thus uses the bandwidth to the maximum. These results are due to the combination of Mosaic's slow requests and the relatively slow response of the server. The tests were performed with no other clients accessing the database - thus we feel that the database server delay is small but still possibly significant. Straight file transfers (not using HTTP) achieved much higher bitrates (column 5).

Media Type	Size (bytes)	Transfer Time (s)	Mbps using HTTP and Tcp/IP	Mbps using File Transfer Protocol
Video	1,567,450	9.21	1.36	5.70
•	1,243,568	7.36	1.35	7.11
•	4,231,256	26.06	1.30	6.15
•	11,534,187	70.3	1.31	11.39
•	5,981,579	36.7	1.30	8.55
Image	242,089	1.80	1.08	4.84
•	212,391	1.65	1.03	12.14
•	237,221	1.88	1.01	5.42
•	214,582	1.41	1.22	10.10
•	211,343	1.31	1.29	11.27
Audio	1,093,409	7.18	1.22	3.75
•	408,876	3.11	1.05	12.25

•	84,925	1.19	0.57	3.24
Text	95,382	0.82	0.93	9.30

Table 2. Bitrates while transferring media over ATM using TCP/IP

5.5 Notes on the Video Browser

The video-tiles video browser is built upon routines and processes which are very important to its functionality. Since the video browser is content-based and uses the camera-shot as its basic browsing unit, the following steps must be taken to pre-process the video:

- the video must be parsed to determine the location of all camera breaks. This is accomplished using camera-operation detection algorithms (not part of this work).
- video-icons must be extracted from the video. This is done by using the location of each camera cut, proceeding to the next frame and sampling it.
- for each video, the collection of icons are saved in a portion of the database -each associated with an actual frame number from the original video.

When a user wishes to browse the video, he/she effectively browses the icons rather than the video itself. The small icons can be transferred much more quickly over the network than can the original size frames. During a browsing session, the user begins by seeing icons representing the entire video stream. By selecting icons which represent sub-segments of the video, the user may focus on a particular part of the video. At each selection, the region of the original video that the user is examining is halved. The following is the pseudo-code for the video browse (assuming a two-level video-tile - i.e. a focus frame and 12 sub-tiles):

INITIAL ITERATION:

```
/* The terminology tile[i] means the the i'th video sub-tile */  
/* focus_frame means the current focus frame on the video-tile*/  
/* Let the number of icons extracted from this video be n */  
/* */
```

```
Put icon[1] on the focus_frame;  
step = n / 12; count = step;  
for x = 1 to 12 {  
    put icon[count] onto tile[x];  
    count = count + step;  
}
```

SUBSEQUENT ITERATIONS: (each time tile[n] is pressed)

```
/* On subsequent iterations, the user wishes to zoom into a spot in the video*/  
/* */  
current_domain_of_frames = current_domain_of_frames / 2;  
Put the icon corresponding to tile[n] onto the focus_frame;  
Choose 12 icons from the new domain and put them on the sub-tiles (as above);
```

In the next chapter we summarize, draw conclusions and suggest areas for future research.

Chapter 6

Conclusions

This thesis has examined in detail many of the concepts and details required to design and implement a multimedia news service of the future. Current technology and terminology were outlined, the architecture of MULTINODES, our multimedia news-on-demand system, was introduced and the feature set of the news application discussed. We then discussed in detail the news video browser and finally gave some background to building graphical user interfaces and outlined our implementation. The following main achievements were made: i) MULTINODES is a new type of news service that incorporates qualities of television and newspapers as well as having qualities that do not exist in other media such as being personal, and intelligent, and ii) MULTINODES has a unique and functional content-based news video browser which allows non-sequential browsing of video streams. The rapid development of computer technologies, such as broadband networks and switches, that will make possible the running of high-end applications such as home shopping and video-on-demand in the home means that there is a need for useful informative applications for home customers. We feel that multimedia news will be one of these and we have discussed the architecture of our system which consists of a news

production center, a hypernews database, a news content database and user sites distributed over a network. The architecture uses some existing protocols so that large scale implementation is relatively easy and offers uniquely functional features to authors of hypernews or fully synchronized news. We described the minimal feature set for this application. We feel that features such as personalized news using multimedia profiles, news follow-ups, bi-directional media assembly, intelligent news agents and searching/querying are necessary for such an application. We have implemented these features and have had success and positive feedback from our users. The system we have designed and implemented has advantages and features not found in other comparable implemented systems and we believe it will be the prototype for future systems.

Finally, we discussed in detail the problems and challenges associated with video streams. The first main challenge is how to represent a (large) video stream in a compact content-revealing way and the second is how to support non-sequential content-based browsing through a video stream while providing local detail and global context. We introduced our tool to answer these challenges, the video-tile, and discussed its usefulness in the context of multimedia news videos and also as a generic browser for temporal data including multimedia news scenarios.

6.1 Suggestions for Future Research

There are many sub-issues related to this research that could be followed-up with further work. There are currently no widely adapted standards for document architecture and certainly none which have been geared specifically to synchronized news documents. Architectures like SGML do not support multimedia because there is no temporal aspect to the model - hence it will not handle continuous media such as video and audio. There has been little research into what users expect to see in multimedia news documents; do they want to watch videos or read text-based stories? Also, is there a medium between glossy,

flashy news and informative, less-attractive news? In other words, will too many features spoil the medium? News agents are another issue that could support future research. Recall that an agent is a process running semi-independently on the users' behalf. How then should agents communicate with each other and with their users? How much information about one user should an agent transmit to another user's agent? What are the social and ethical issues regarding agents? This implementation made use of a hypermedia browser that was already well-established. There is a need for the development of a new browser with which to browser news and other information which will make use of a faster protocol and will take advantage of fiber networks. Further, how do people really interact with news information? Is there a "universal" access metaphor that will allow document creators to make more readable/browseable documents?

There is a need for further research into video browsing. It should be possible to use a combination of text and frames during the process of browsing. Perhaps by supplying a video frame and text a browsing session can be automatically "directed" into appropriate segments of the video.

References

- [1] Hoffert E., Gretsich G., "The Digital News System at EDUCOM: A Convergence of Interactive Computing, Newspapers, Television and High-Speed Networks", *Comm. of the ACM*, Vol.35, No.4, April 1991
- [2] Bender W., Lie H., Orwant J., Teodosio L., Abranson N., "Newspace: Mass Media and Personal Computing", *Proc. of USENIX'91*, June, 1991
- [3] Koons W., O'Dell A., Frishberg N., Laff M., "The Computer Sciences Electronic Magazine: Translating from Paper to Multimedia", *Proc. of CHI'92*, May3-7, 1992
- [4] Lippman A., Bender W., "News and Movies in the 50 Megabit Living Room", *IEEE, Globecom*, Tokyo, November, 1987
- [5] Miller G., Baber G., Gilliland M., "News On-Demand for Multimedia Networks", *Proc. ACM Multimedia 93*, June 6, 1993
- [6] Ohkubo M., Kobayashi N., Nakagawa T., "Design of an Information Skimming Space", *Proc. ACM Multimedia 93*, June 6, 1993
- [7] Jackson D., "Extra! Readers Talk Back!", *Time Magazine*, Time Canada Ltd., Spring, 1995
- [8] Jacobs P., Rau L., "SCISOR: Extracting Information from On-line News", *Comm. of the ACM*, Vol. 33, No.4., November 1990
- [9] Kelly K., "What People Really Want on the Net", *Wired Magazine*, Louis Rossetto, February 1995
- [10] Fox E., "Advances in Interactive Digital Multimedia Systems", *IEEE Computer*, Vol.24, No.10, pp.9-22, October, 1991
- [11] Koegel J., "The Advent of (Standardized) Multimedia Environments", *World Conf. on Educational Multimedia and Hypermedia*, AACE, June, 1993

- [12] Minoli D., Keinath R., "Distributed Multimedia Through Broadband Communication Services", Artech House, Boston, 1994
- [13] Buford J., "Multimedia Systems", ACM Press, New York, 1994
- [14] Gibbs S., Tschritzis C., "Multimedia Programming", ACM Press, New York, 1994
- [15] Miller M., "Introduction to Digital and Data Communications", West Publishing Co., New York, 1992
- [16] Vetter R., Spell C., Ward C., "Mosaic and the World-Wide Web", *IEEE Computer*, pp49-57, October, 1994
- [17] Karmouch A., Emery J., "A Multimedia Document Architecture and Rendering Synchronization Scheme", *Proc. Second Intl. Conf. on Broadband Islands*, Athens, 1993
- [18] Rody J., Karmouch A., "A Remote Presentation Agent for Multimedia Databases", *Proc. of IEEE Int'l Conf. on Multimedia Computing & Systems*, May 1995
- [19] Berners-Lee T., Cailliau R., Luotonen A., Nielsen H., Secret A., "The World-Wide Web", *Communications of the ACM*, Vol. 37, No. 8, August, 1994
- [20] Berners-Lee T., "Hypertext Transfer Protocol - A Stateless Search, Retrieve and Manipulation Protocol", Internet RFC (Draft), November, 1993
- [21] Emery J., "Design and Implementation of a Time-Based Multimedia Document Architecture", M.A.Sc. Thesis, University of Ottawa, Dept. of Electrical Engineering, 1994
- [22] Maes P., "Agents that Reduce Work and Information Overload", *Comm. of the ACM*, 37(7), July, 1994
- [23] Castro J., "Just Click to Buy", Time Magazine, Time Canada Ltd., Spring 1995, p.74
- [24] Furnas G., Generalized Fisheye Views, *Proc. of CHI'86 Human Factors in Computing Systems*, ACM, 1986
- [25] Mackay E., Davenport G., Virtual Video Editing in Interactive Multimedia Applications, *Comm. of the ACM*, 32(7), July, 1989

- [26] Mackinlay J., Robertson G., Card S., Perspective Wall: Detail and Context Smoothly Integrated, *Proc. of CHI'91 Human Factors in Computing Systems*, ACM, 1991
- [27] Mills M., Cohen J., Wong Y., A Magnifier Tool for Video Data, *Proc. of CHI'92*, May3-7, 1992
- [28] Tesodosio L., Bender W., Salient Video Stills: Content and Context Preserved, *Proc. ACM Multimedia 93*, June 6, 1993
- [29] Tonomura Y., Akutsu A., Otsuji K., Sadakata T., VideoMAP and VideoSpaceIcon: Tools for Anatomizing Video Content, *Proc. of INTERCHI'93*, April 1993
- [30] Ueda U., Miyatake T., Sumino S., Nagasaka A., Automatic Structure Visualization for Video Editing, *Proc. of INTERCHI'93*, April 1993
- [31] Zhang H., Gong Y., Smoliar S., Tan S., Automatic Parsing of News Video, *Int. Conf. on Multimedia Computing and Systems*, IEEE, 1994
- [32] Zhang H., Smoliar S., Developing Power Tools for Indexing and Retrieval, *S.P.I.E.*, Vol 2185, 1994
- [33] Zhang H., Gong Y., Smoliar S., Tan S.Y., Automatic Parsing of News Video, *Int. Conf. on Multimedia*, IEEE, 1994
- [34] Cruz G., Hill R., Capturing and Playing Multimedia Event with STREAMS, *Proc. of ACM Multimedia '94*, ACM, 1994
- [35] Arman F., Depommier R., Hsu A., Chiu M-Y., Content-based Browsing of Video Sequences, *Proc. of ACM Multimedia '94*, ACM, 1994
- [36] Sarkar M., Brown M., Graphical Fisheye Views of Graphs, *Proc. of CHI'92*, ACM, 1992
- [37] Johnson E., Reichard K., "Power Programming...MOTIF", Management Information Source Press Inc., New York, 1993
- [38] Karmouch A., A Multimedia Information and Communications System: MEDIABASE, *Proc. of Multimedia Comm. '93*, Banff, 1993, pp.234-243

- [39] Sun Microsystems, The HotJAVA Browser: A White Paper, Sun Microsystems, 1995
- [40] Niblack W., Barber R., Equitz W., et al, The QBIC Project: Querying Images by Content Using Color, Texture, and Shape, SPIE, Vol.1908, 1993
- [41] Arman F., Hsu A., Chiu M., Image Processing on Compressed Data for Large Video Databases, *Proc. of ACM Multimedia'93*, ACM Press, June, 1993, pp267-272
- [42] Rowe L., Boreczky J., Eads C., Indexes for User Access to Large Video Databases, SPIE, Vol.2185, 1993, pp150-161
- [43] Li Li, A. Karmouch, N.D. Georganas, Real-time Synchronization Control in Multimedia Distributed Systems, ACM Computer Communication Review, vol. 22, no. 3, pp. 79-86. 1993.
- [44] L. Li, A. Karmouch, N.D. Georganas, Multimedia Teleorchestra with Independent Sources: Part 1: Temporal Modeling of Collaborative Multimedia Scenarios (to appear) ACM Multimedia Systems, 1994.
- [45] L. Li, A. Karmouch, N.D. Georganas, Multimedia Teleorchestra with Independent Sources: Part 2: Synchronization Algorithms, (to appear) ACM Multimedia Systems, 1993.
- [46] Gomez L., Pea R., Distributed Multimedia Learning Environments: Why and How?, Technical Report, Institute for the Learning Sciences and Bellcore, May, 1992.
- [47] Cole B., The Technology Framework, Special Report on Multimedia, IEEE Spectrum, March 1993, pp32-39.

END

0 | 5 - 0 6 - 9 | 6

FIN