The Abstract Media Model

by

West Suhanic

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Systems Design Engineering

Waterloo, Ontario, Canada, 1997

0-612-30651-8

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

The Abstract Media Model

The goal of this thesis was to create a comprehensive model which could be used as a framework for thinking about, working with, and understanding the many elements of media. This model is called the Abstract Media Model(AMM). The AMM has been built using a systems design methodology so that not only were the overall technical aspects of problems carefully considered, but there was equal concern for the economic, social, human, and political parameters. The AMM in addition to providing a media framework gave rise to:

- a new media representation called the Abstract Media Code(AMC). The AMC is important as it can represent many media types. The AMC, because of its time code foundation, also serves as a bridge. It plays this role by linking the existing media production community with its huge investment in analogue-based tools, techniques and methodologies to the emerging digital world. The AMC is also an example of convergence in media representations and as such can support the technology convergence we are currently witnessing.

- a deccntextualised, service-oriented media model which is implemented, in prototype form, using distributed object computing(DOC).

- a new type of database system built using neural net technology which utilises the Abstract Media Code.

Each of these accomplishments was made possible by the partition-based design of the AMM. This type of design has resulted in the AMM providing a fragmented and decontextualised view of media. The fragmentation and decontextualisation of media is important as it is the basis for recombination which in turn presents opportunities for technological, social and cultural innovation.

Acknowledgements

I owe thanks to many people. These include:

- My supervisors, Professor Barry Wills and Professor John Robinson.
- Professor Paul Beam, English Department, University of Waterloo.
- Professor Michael Murphy, Director of the Rogers Communications Centre, Ryerson Polytechnic University.
- Professor Ted Kesik, School of Architecture, Ryerson Polytechnic University.
- Professor(Emeritus) Isaac Morgulis, Film and Photography, Ryerson Polytechnic University.
- Professor Dennis Mock, V.P. Academic, Ryerson Polytechnic University.
- Professor Darryl Williams, Film and Photography, Ryerson Polytechnic University.
- George and Michelle Suhanic.
- Guilherme and Maria Martins.
- and my wife, Eduarda Martins.

Lastly I wish to thank the Canadian tax-payer for providing me with an opportunity to educate myself.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

The goal of this thesis is to create a media-driven model which can be used as a framework for comprehending and working with the many elements of media. This model is called the Abstract Media Model(AMM).

Figure 1.1: Some Elements of Media

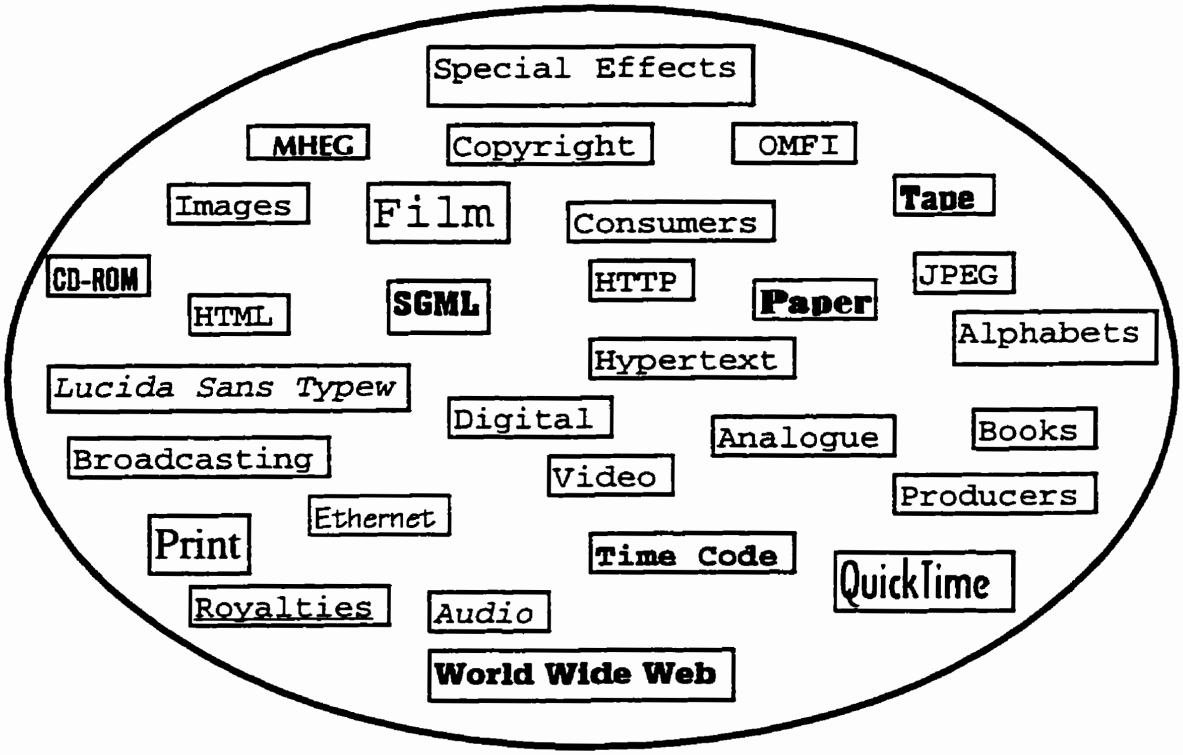


Figure 1.1 contains some of the standards, technologies and terms which content providers, media users, application developers, technology developers and service providers must understand and utilise. The purpose of Figure 1.1 is to show and convey the variety of issues that have to be addressed when working with media.

The framework has been built using a systems design methodology so that "Not only must the overall technical aspects of these problems be carefully considered, but there must be equal concern for the economic, social, human and political parameters." (Systems Design Engineering, Graduate Studies and Research, 1991).

The thesis is structured so that Chapter One contains the definition of medium that will be used for the research, and the type of model that will be built. Chapter One also provides a needs analysis, followed by sections which deal with the focus of the research work, related work, a problem statement, the constraints that the Abstract Media Model must satisfy, the design criteria and the thesis contributions. Chapter Two presents and develops the AMM. Chapter Three is devoted to developing the media representation of the AMM and explaining why media representation is important. Chapter Four sees the AMM being used as an analysis and design tool. Chapter Five contains the prototype implementation of the AMM using distributed object technology.

1.1) Media Definitions

One definition of medium, found in the second definition of the Oxford English Dictionary, is:

an intermediate agency, means, instrument or channel

Note that the Oxford Dictionary goes on to state that media, the plural of medium, such as newspapers, radio, television, etc., can be collectively defined as:

vehicles of mass communication.

The Webster's Third New International Dictionary has several definitions of medium including the following:

The material or technical means for artistic expression

The inclusion of the term " artistic" in the above definition is unreasonably narrow. Science, design and engineering are all disciplines which need to be expressed. Therefore I will define medium to be:

The means for expression or communication.

In order to facilitate reference to the various media types, the following classification scheme is provided.
" Still imagery" denotes:

- still photographic images
- graphs
- maps
- CAD-generated images
- geometric models
- single frame pictures.

" Continuous imagery" means any sequence of still imagery and " Text-based media" denotes:

- traditional printed text
- digital text
- PostScript documents
- mark-up based documents where the mark-up languages include the Standard Generalized Markup Language(SGML) and the Hyper Text Markup Language(HTML).

Finally the term " audio" will be used to represent digital, analogue and MIDI-created audio. Continuous media will then refer to audio and continuous imagery.

The AMM is designed to function with all media types. Its role is to help users resolve their media problem regardless of their medium choice. This is the motivation for the model's name: the Abstract Media Model. The term abstract media was coined to move the user's focus away from a particular medium towards solving the problems and challenges posed by working with media in general.

1.2) Model Definition and Type

1.2.1) Definition of a Model

One definition of a model is: "A description or analogy used to help visualise something(as an atom) that cannot be directly observable"(Webster's New Collegiate Dicti onary). Roe, Soulis and Handa(1969) suggest that:

> *A model is defined as a meaningful representation of pertinent particulars of reality.* (p. 230)

For the purpose of this work, the particular reality that is being dealt with is media. The AMM will then provide the user with a visualisation of media's "pertinent particulars of reality". Therefore a model will be defined, for this thesis, as:

> *A visualisation and simplified representation of media's pertinent particulars of reality.*

1.2.2) Types of Models

There are several types of models. One type is a verbal/logical model which uses verbal analogies and sometimes give rise to paradigms(Intriligator, 1978). For these models, systems are given a characteristic of purposefulness. For example, in physics the "principle of least action" has a particle in motion behaving "...as if it were minimising the energy required for its motion."(Intriligator, 1978, p. 16).

Another type of model is a physical model which can be a scaled-up or scaled-down version of a real world system. For example a model train set can be used to emulate, for study, new switching systems. A physical model of a DNA or protein molecule can be built and the results predicted and observed in organisms and their behaviours. In each case such physical models can be easier to work with, because of restrictions to scale and function, than the entities they represent.

Models which represent relationships geometrically have many applications ranging from biology to economics. The same is true of algebraic models where systems are represented by sets of equations. These types of models are also used for simulation purposes where a system becomes so complex that a solution is not possible. Rather, it is only possible to simulate the model's behaviour under a variety of assumptions on a computer. Econometric models are algebraic models with one or more random variables. An econometric model "...represents a system by a set of stochastic relations among the variables of a system."(Intriligator, 1978, p. 22).

Descriptive models include "...static three dimensional models, written descriptions and reports, and detailed drawings."(Roe, Soulis, Handa, 1969, p. 232) and fuzzy linguistic models that "...are based on collections of IF-THEN rules with vague predicates and use fuzzy reasoning."(Yager, Filev, 1994, p. 156). Descriptive models, like geometric and mathematical models, can be used for evaluation and prediction. Most models are descriptive in one way or another. Descriptive models, in addition to describing situations also can be used "...for purposes of definition, specification and instruction."(Roe, Soulis, Handa, 1969, p. 233). Essentially models can be categorised as descriptive and manipulative. Each type has a purpose and a place. Notwithstanding the popular and pervasive role of manipulative mathematical models, descriptive models play a very important role, especially in the design process, as they "...can be used to obtain necessary resources and also for product promotion, installation, operation and maintenance."(Roe, Soulis, Handa, 1969, p. 234).

The Abstract Media Model is a descriptive model. It is descriptive because it identifies and presents media attributes in a non-mathematical framework. This descriptive approach is not without precedent. The Reference Model of Open Systems Interconnection(Zimmerman, 1980) is a descriptive model developed and used to comprehend, organise and control a very heterogeneous networking environment. Network vendors work with the framework provided by the Open Systems Interconnection model. Another example is the Dexter Hypertext Reference Model(Halasz, Schwartz, 1990). Hypertext system developers refer to the Dexter model's description of an ideal hypertext system when building a hypertext system. Therefore the AMM was created given this successful tradition of building descriptive models.

The AMM provides a framework for thinking about, and working with, media. It will describe what I believe to be media's fundamental attributes. This is important because once this model is in place it can then be used to understand, contextualise and utilise the many media technologies, protocols and terms. Figure 1.1 contains some of the technologies, protocols and terms. I put that diagram at the opening of the thesis for two reasons:

1. to make obvious how many different elements can be brought to bear on a media project.

2. to present examples of media's fundamental attributes; i.e., content, storage, commerce, delivery, representation, expression and usage. These are the attributes that come together to form the AMM which is the subject of Chapter 2.

As was noted above, Figure 1.1 contains some but not all of the technologies, protocols and terms of the media world. As time advances, Figure 1.1 will grow as new elements are introduced. A good example of a potent new arrival is the World Wide Web(WWW). Therefore a framework for working with the many existing, new and unknown elements is a valuable tool.

1.3) Media Needs

" The multimedia community agrees almost unanimously that both the largest potential and the most serious problems lie in the development of distributed multimedia(DMM) applications." (Muhlhauser, Gecsei, 1996, p. 48). To support the development of distributed multimedia applications, an encompassing framework must be developed which provides:

- *organised access to all (DMM and other)services relevant to application development, and*

- *a consistent, simple, and homogeneous view onto these services, following a certain mental model or paradigm*(Muhlhauser, Gecsei, 1996, p. 48).

The AMM, by identifying fundamental media services and a paradigm providing a consistent view onto these services, fulfils the need for an encompassing framework. Furthermore frameworks are " ...becoming increasingly common and important. They are the way object-oriented systems achieve the most reuse." (Gamma *et al.*, 1995, p. 28).

The hypermedia community has also identified support for time-based media as an area needing further research(Gronbaek, Trigg, 1994). For film makers and television producers the basic needs are similar to those that existed when the film/entertainment industry started in 1895; i.e., how to work with continuous media. This means, for example, providing the infrastructure, information and content needed to pull together two hundred thousand still images to make a movie. In the field of hypertext/hypermedia, if someone is building a hypermedia package on space-craft design, for example, the model must describe what information would be needed by the linkage mechanism so that, say, the ninth frame of a documentary can be linked to an equation of orbital mechanics. If an educator with a very limited budget is putting together a course on linguistics, the model must provide an information structure for the authoring software and the continuous media server so that text which is found on a machine in, say, Sweden can be linked with the audio component located in

Singapore and the video segment found at Laurentian University in Northern Ontario. On the provider side of the problem, the Canadian Broadcasting Corporation and the National Film Board will want to digitise their video libraries. What information should be included with the content so that users ranging from video editors, to educators, to design engineers can access and work with this proprietary content? Indeed, what are the broad questions that have to be asked when setting up a digital library? These are the type of needs that are taken into account in the design of the AMM.

1.4) Focus of Work

The focus of this research is to create and develop a comprehensive framework, called the Abstract Media Model(AMM), for understanding and working with a wide range of media. With this focus a number of research areas can be used in conjunction with the AMM to design media systems. These areas include:

- Database design for continuous media or the physical synchronisation of continuous streams distributed across ATM networks. This work is also being addressed by other researchers such as Little and Ghafoor(1993), Delodder, Verbiest and Verhille(1994) and others.

- User interfaces for working with continuous media. These have been extensively examined by such researchers as Laurel(1992), Schneiderman (1992), Norman(1991) and Norman and Draper(1986).

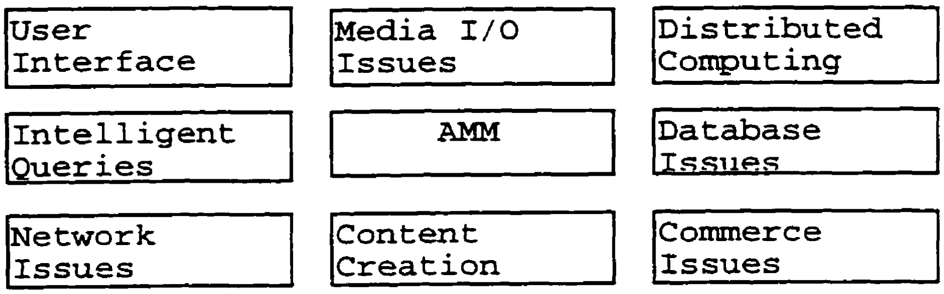
- I/O issues relating to multimedia systems. See, for example, the work of researchers like Reddy and

Willie(1994).

- Building environments for working with multimedia and hypermedia systems. Examples include the Andrew Toolkit(Borenstein, 1989)(Sherman, Hansen, McInerny, Neuendorffer, 1990), Bolt, Bernak and Newman's Slate(Lison, Crowley, 1989), the AthenaMuse project(Hodges, Sassnet, 1993), the Amsterdam Hypermedia Model(Hardman, Bulterman, van Rossum, 1994), the work of Gibbs and Tsichritzis(Gibbs, Tsichritzis, 1994) and the World Wide Web and its various browsers(Berners-Lee et al., 1994).

Figure 1.4.1 provides a diagram of the AMM's relationship to other research areas. The AMM's position in the middle is not to suggest that the AMM is an all-encompassing and central solution. Rather the intention behind Figure 1.4.1 is to show that the framework provided by the AMM is one way to begin to harness and use the results of these other research areas.

Figure 1.4.1: AMM and Other Media Research Work



The Abstract Media Model has been designed to be of use to investigators in each of these areas and others. For example, application builders could use it to develop the next generation of non-linear editing systems which support distributed virtual studios. Builders of digital collections and libraries could use the model to determine what data to include with the digital content so that network-accessible, proprietary collections can be built. The AMM could be used by data architects as they build multimedia data models. Multimedia file systems designers could employ the AMM to help them determine what data representation(s) should be provided to the user. The designers of compression schemes could use this model to build or enhance existing compression schemes to support abstract media. Builders of interactive hypermedia could use the framework to construct interaction schemes or event scheduling schemes. Just as the Dexter Hypertext Reference Model(Halasz, Schwartz, 1991) and the Hypertext Abstract Machine(Campbell, Goodman, 1988), were developed to help understand and build hypertext systems, the Abstract Media Model can, perhaps, provide a method of instruction and construction for a wide variety of media.

1.5) Problem Statement

The desired state or goal is to build a descriptive model which represents media's dimensions and provides a framework for thinking about and working with media. It is intended to allow the wide variety of media users, creators and technology providers, each having a particular point of view or area of expertise, to consider the implication(s) of their design decisions for their fellow media brethren. For example, builders of device drivers could use the model to determine how their software could potentially be used by creators of distributed hypermedia. A developer of a multimedia tool-kit could refer to the model to determine what information structure must be included in the tool-kit given their intended audience and applications. Similarly developers of compression schemes or network services could refer to the AMM to determine what functionality they should supply to support the many dimensions of media or determine what dimensions of media they wish to support.

1.6) Constraints

The model must satisfy several constraints. These include:

- The model should be media-driven. This means that it will be derived from the fundamental characteristics of media: content, storage, commerce, delivery, representation, expression and usage.

- The model should be application-neutral so that it can be applied to a variety of applications ranging from hypermedia authoring systems, book publishing, non-linear digital editing systems and real-time performance support systems.

- The model should support multiple points-of-view. It must be of use to content providers, application builders, network service providers, compression chip designers to name a few. A key part of the model's value lies in the fact that it will be of use to the whole spectrum of the media industry and not tailored to any particular sector, industry or user.

- The model should be equally applicable to both the analogue and digital domains. It should function as a bridge between the two. The bridging capability is especially important given the history of, and continuing huge investments in, analogue technology.

Given these constraints, here are the criteria used to judge the solutions.

1.7) Design Criteria

The design criteria for the AMM are: 1)how well the model fits the data where the data are abstract media; 2)how well solutions are provided to media problems and 3)how well new insights are provided into the structure of media.

1.8) Thesis Contributions

The primary contribution of the Abstract Media Model will be a general framework for working with, thinking about and understanding a wide variety of proprietary and non-proprietary media in a networked or non-networked world. The AMM can be used to:

- analyse existing media, tools and technology
- design new media, tools and technologies.

Another potential contribution of the AMM is that of facilitator. The AMM, by providing a common lexicon and structure, could facilitate the discussions of media theoreticians, practitioners, performers and technologists. This last point is especially important as these groups tend to use different terms and measures. The digital technologists speak about " bits" and " bytes" and " free, ubiquitous Internet" access whereas traditional media practitioners speak about time code and " pay-per-view" scenarios. In terms of solving issues identified by the media research community, the Abstract Media Model provides a framework and paradigm to support distributed multimedia applications(Muhlhauser, Gecsei, 1996). The framework, by capturing the design decisions common to the media domain, supports design reuse over simple code reuse. This will allow the application designer to focus on application development(Gamma et al, 1995). The Abstract Media Code presented in chapter 3 could be of use to the hypertext community as they seek ways of working with time-based media(Gronbaek, Trigg, 1994). The AMC could also make a contribution to the MPEG-7 specification. MPEG-7, the latest member of the MPEG family, is called the "Multimedia Content Description Interface". Its goal is to specify a standardised description of various types of multimedia information. This description shall be associated with the content to allow searching for material that is of interest to users(http://drogo.cselt.stet.it/mpeg/mpeg_7.htm).

To address the needs of media users in general, the AMM is designed to play the role that the International Standards Organisation's(ISO) Reference Model of Open Systems Interconnection(OSI) played for the network community; that is, just as the ISO/CSI model provided a baseline framework for the computer/communications industry(Zimmermann, 1980), the AMM provides a baseline framework for the media industry.

Chapter 2: The Abstract Media Model

This chapter presents the Abstract Media Model given the research goals, constraints and design principles outlined in Chapter 1.

2.1) Fundamental Attributes of a Medium

The purpose of this section is to identify the fundamental attributes that I believe comprise a medium. The six fundamental attributes are: content, expression, commerce, usage, delivery and storage. A depiction of this relationship is found in Figure 2.1.1 below. The role of Figure 2.1.1 is to provide the reader with a visualisation of the components that comprise a medium. The justification for: presenting media as comprised of these six attributes and commerce as an all encompassing attribute, is provided by a fundamental analysis of media which is the subject of the rest of this chapter. Figure 2.1.1 is re-expressed using a Table construct. This alternate representation is found in Table 2.1.1(p. 17). The Tabular representation is introduced as I will also be using it to represent a medium and its components throughout the thesis because tables are easier to work with. However when reading the tabular representation, I ask the reader only to read down columns, not across rows. The tabular representation is solely a re-expression of Figure 2.1.1. The fundamental media analysis follows Table 2.1.1. Note that chapter 4 contains many examples of the AMM being used as an analysis and design tool.

Figure 2.1.1: Fundamental Media Attributes

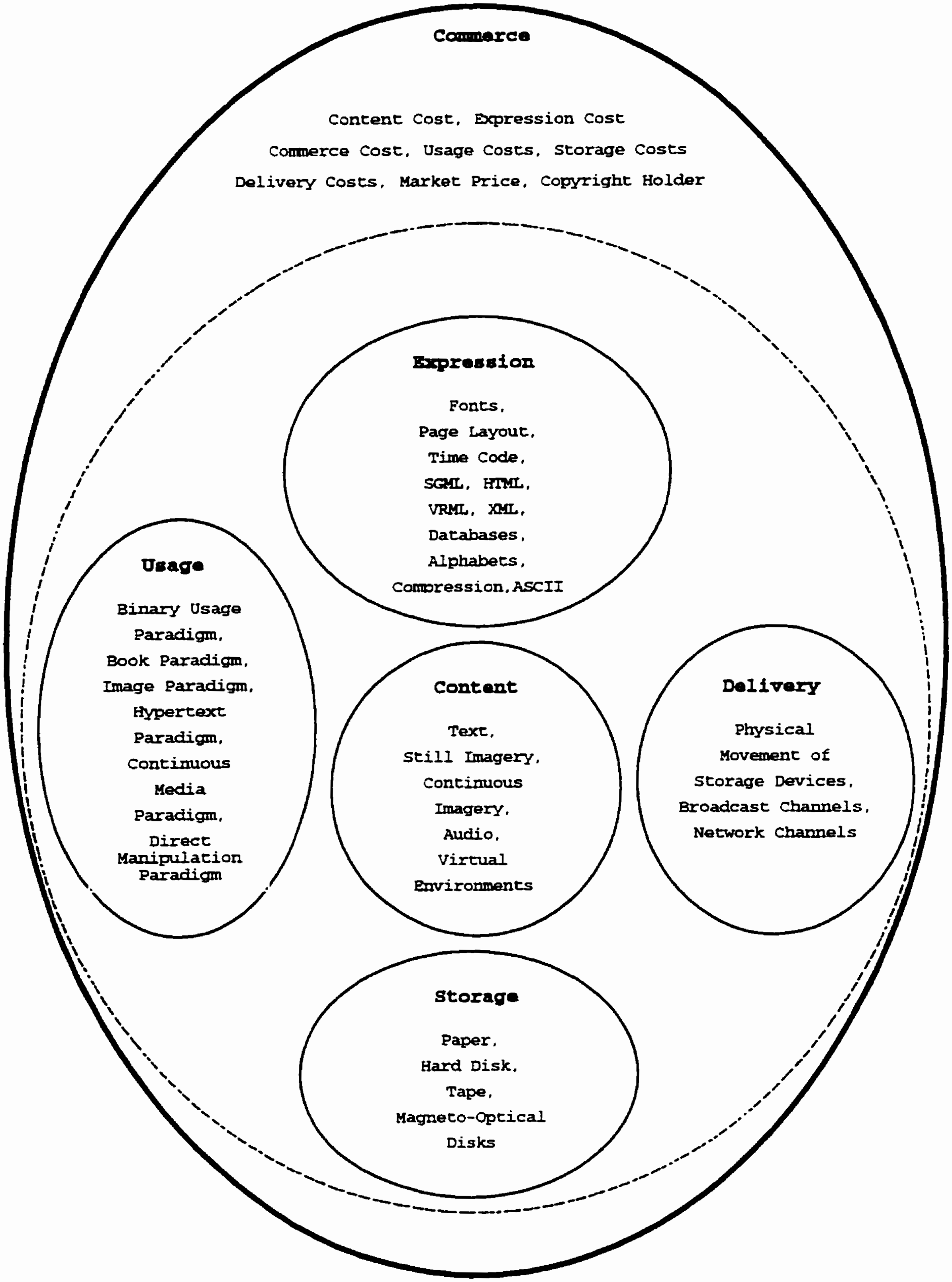


Table 2.1.1: Fundamental Media Attributes

CONTENT	EXPRESSION	COMMERCE	USAGE	DELIVERY	STORAGE
Text	Fonts	Content Cost	Binary Usage Paradigm	Physical Movement of Storage Devices	Paper
Still Imagery	Page Layout	Expression Cost	Book Paradigm	Broadcast Channels	Hard Disk
Continuous Imagery	Time Code	Commerce Cost	Image Paradigm	Network Channels	Tape
Audio	SGML, HTML, VRML, XML	Usage Costs	Hypertext Paradigm		Magneto-Optical Disks
Virtual Environments	Databases	Storage Costs	Continuous Media Paradigm		
	Alphabets	Delivery Costs	Direct Manipulation Paradigm		
	Compression	Market Price			
	ASCII	Copyright Holder			

2.1.1) Content

A medium, as I have defined it, provides the means for communication. Content is the essential meaning or as the Oxford English Dictionary says: " The substance or matter(of cognition, or art, etc.) as opposed to form." . Content is the matter that is to be communicated. Content can take many forms. An example is the oral utterances which have " ...constituted almost all of human existence." (Ong, 1995, p. 10). The oral content base of humanity was augmented by the written word with the earliest examples having occurred 5,000 to 6,000 years ago(Ong, 1995). Hieroglyphics used by the Egyptians, Aztecs and Mayans(Bridgwater, Kurtz, 1956) and the Palaeolithic paintings of Spain and France are early examples of still imagery which gave the pictorial representation of human experiences. Peter Mark Roget's 1824 " The Persistence of Vision with Regard to Moving Objects" gave the world continuous imagery, the base for film and video. This content base is being augmented by new content such as virtual reality and multi-dimensional audio(Begault, 1994).

2.1.2) Storage

The very act of creating content requires that it be stored. For example, shooting a TV show means recording it on video tape, film stock, hard or optical disks. A standard feature-length movie can take up to six terabits of storage. This exceeds the Library of Congress in digital form(Stein, 1994). Problems of preservation and storage have existed ever since humans have attempted to record ideas. Human memory has been an important storage device. Given that " ...we have no evidence of writing before 5,000 to 6,000 years ago - a mere nothing in 150,000 to 500,000 years of existence." (Ong, 1995, p. 10) memory has served as the earliest intra-generational and inter-generational knowledge transfer mechanism. Memory's role has changed over time " ...as different media circumstances exempted it from strictly rhetorical obligations and enlisted it into the service of ethical, metaphysical and historical remembrance." (Kelber, 1995, p. 443). One different media circumstance was the recording of ideas on stone as manifested in the Palaeolithic paintings and carvings found in the caves of Spain and France. Another example is the recording of commercial transactions and stories(Ullman, 1969)(De Kerckhove, 1995) on papyrus rolls. The central problem, then and now, is how to store content. In addition to all of the earliest storage devices like stone, clay, papyrus rolls, paper and memory, we can now use magneto-optical technologies, tape, solid-state memory and holographs.

2.1.3) Delivery

Assuming an audience of at least one, content must also be delivered. In pre-alphabet days content was delivered via speech, smell, vision and touch. The early Greeks used drama, songs and poetic forms to deliver knowledge(Laurel, 1991). Oral presentation will always be an important delivery channel. From a historical perspective the oral stage has dominated almost all of human existence(Ong, 1995). With the introduction of writing 5,000 to 6,000 years ago, the concept of delivering content by handing the storage device that had the writing on it from person to person was introduced. The best modern example of this is the book selling/publishing tradition. In addition to the traditional delivery methods, content can be delivered via the broadcast technologies of radio and television. As we move into the digital era, delivery of content can be via networks(Cerf, 1991).

2.1.4) Usage

Content must be in a form that allows it to be accessed. Accordingly content must be accompanied by what I call a usage paradigm. The word paradigm is defined in the Webster's New Collegiate Dictionary to be " ...an outstandingly clear or typical example or archetype." . Within the context of this thesis a usage paradigm will be an example of a way or mode of accessing or using media. For example, the book usage paradigm has the user reading content which has been set to a page granularity. In the Western world each page is read from upper left to lower right before turning to read the next. The usage paradigm for oral content is binary; i.e., talk or don't talk. This binary paradigm also applies to the perception of images, either you look at them or you do not. The binary and book paradigms have been joined by the hypertext, continuous media and direct manipulation usage paradigms. These paradigms exist to varying degrees in both the analogue and digital worlds.

2.1.5) Expression

A medium expression is another fundamental attribute that I propose. In the context of this thesis I define a medium expression as an entity or process that can alter, re-represent or augment the content stream. The motivation for employing media expressions includes:

- saving space on the storage device
- enhancing content by utilising different fonts, alphabets and perhaps languages
- adding information to facilitate content manipulation, representation or understanding.

For example, the Greek alphabet became the standard writing system and common information system for many Indo-European languages(De Kerckhove, 1995, p. 81). Greeks did not always separate words(Tversky, 1995, p. 37). It has been suggested that printer craftsmanship and pride led, in the tenth century, to the custom of putting a space between each word(Lucky, 1991, p. 13). Formatting of text such as adding space between words, creating paragraphs and effective page layout can be viewed as an expression as it facilitates understanding of a content stream. Other examples of expressions include: fonts and the ASCII, EBCDIC and UNICODE systems. The former can enhance presentation, understanding and minimise storage while the latter can be used to express a digital information stream(Lucky, 1991).

More recent examples of media expressions are the Standard Generalised Mark-up Language(SGML), its subsets including, the Hypertext Mark-up Language(HTML) and the Virtual Reality Mark-up Language(VRML). In all these cases the content stream, which can include text, video, audio and images, is integrated and enhanced by these mark-up mechanisms. The links found on a web page, created via HTML, can be viewed as a knowledge map. This knowledge map is itself an expression as it is adding information to facilitate content representation and understanding albeit most likely at a higher level.
Each of these mark-up schemes impacts the content stream in a way not observed by the user. For example the structure imposed by an SGML Document Type Definition(DTD), though not visible, impacts the use of that stream by establishing beforehand the hierarchy of structural devices by which content is ordered and expressed. Still imagery, video, film and audio special effects provide more examples of expressions. The special effect could be a blurring or transition effect. In this case the video stream has an effect applied to it so that the original stream is augmented and transformed. Compression technology can be viewed as an (invisible) medium expression as it transforms and re-represents a content stream with the goal of reducing bandwidth requirements for both transmission and storage. For " lossy" compression schemes the content stream is permanently altered in order to lower transmission and storage costs while " lossless" schemes maintain the original content stream but at greater storage and transmission costs.

2.1.6) Commerce

Cuneiform, the world's first syllabary, was created by the Akkadians. They developed it by utilising the Sumerian invention of stylised pictograms which were created to record commercial transactions. Cuneiform was an important development as it played a pivotal role in the development of the Phoenician, Greek and Roman alphabets(De Kerckhove, 1995, p. 25),(Tversky, 1995),(Bahr, Johnston, 1992). The point is that commerce was instrumental in developing alphabets which in turn permitted the concept of authorship. Authorship has since evolved from the complete anonymity of the first Sumerian bookkeepers(De Kerckhove, 1995, p. 22) and the medieval scribes(Kelber, 1995, p. 432) to the modern day celebrity authors. This later development has its roots in the Gutenberg press which " ...did away with anonymity, fostering ideas of literary fame and the habit of considering intellectual effort as private property." (McLuhan, Fiore, p. 122). As soon as " ownership" , and hence the notion of " property" enters the picture, commerce becomes a possibility as all media takes on a " market value" . The notion of media as a commodity was first legally instituted in 1710 by the Statute of Anne, the world's first copyright statute passed by the English parliament(Samuelson, 1995). The statute, by encouraging educated men to "...compose and write useful books"(Samuelson, 1995, p. 16), attempted to achieve a specific economic purpose: to make book-writing a profession with a "product ". That ownership and commercial exploitation of media are realities, has never been more evident than in the 1990s, with massive media corporations like Viacom and Turner Broadcast Systems whose market value is based, almost entirely, on the content they control. In addition to controlling vast amounts of content, the media conglomerates are paying huge sums to obtain and control more.

Two examples are the 1990 sale of the United Artist film library for $625 million(Fleming, p. 146) and the 1996 sale of Carolco's film library for $58 million(Globe and Mail, ROB). Movie and TV producers and musicians purchase stock footage for inclusion into their work. Indeed "...digital compositing has, in a sense, made practically every moving image a potential stock shot."(Kim, 1995, p. 82) which can be bought and sold. The use of commercial media libraries is growing, especially for interactive multimedia. For example, The Sony Wonder Technology Lab, an interactive science and technology lab in New York City, uses a variety of archival footage from the Image Bank, a company specialising in film footage, photography and illustrations(Kim, 1995, p. 82). Other collections, such as the Video Tape Library Inc. have been used to provide footage for museum exhibitions, games and theme parks(Videography, Fall, 1995). One company, Global Village Stock Footage, allows their stock footage library to be accessed via the Web at http://www.video.videosource.com/footage(Kim, 1995). My point is that media, which is made up of content, expression(s), delivery mode(s), storage mode(s) and usage paradigms, is a product which can be traded in a market. It is this complete entity that is the commodity. The holder of the media copyright can lay claim to all or a portion of any income stream derived from the media. Indeed Sumner Redstone, the chairman of Viacom Corporation which owns MTV and Paramount Studios, stated that his job was: "...to exploit our copyright in every format and over a vast myriad of distribution systems whether it's broadcast, telephone, cable..."(O'Shaughnessy, 1994, p. 237).

Therefore a medium is a product and it has to be treated as such. This explains why the AMM must have a "commerce " attribute. Therefore, in addition to the attributes of content, storage, delivery, usage and expression, a medium also now has a commerce attribute which is, on the surface, the price of any given medium. The price of the medium is however comprised of the price of the other fundamental attributes. For example a book's price reflects:

- **the cost of the content,**
- **the cost of the storage mechanism; i.e., the cost of paper and printing.**
- **the cost of the delivery mechanism; namely, moving the books to market.**
- **the cost of the usage mechanism; i.e., the cost of creating and laying out the content in a book form.**
- **the cost of the expressions such as font, layout and look.**
- **the cost of doing business or the cost of commerce.**

Therefore the commerce attribute has many cost elements as is evident from Table 2.1.1. In order to facilitate the discussion all these costs will be represented by the term media costs so that:

Media Costs = Content Costs + Expression Costs + Commerce Costs + Usage Costs + Delivery Costs + Storage Costs

Therefore the fundamental attributes that comprise a medium are:

Content, Expression, Commerce, Usage, Storage and Delivery

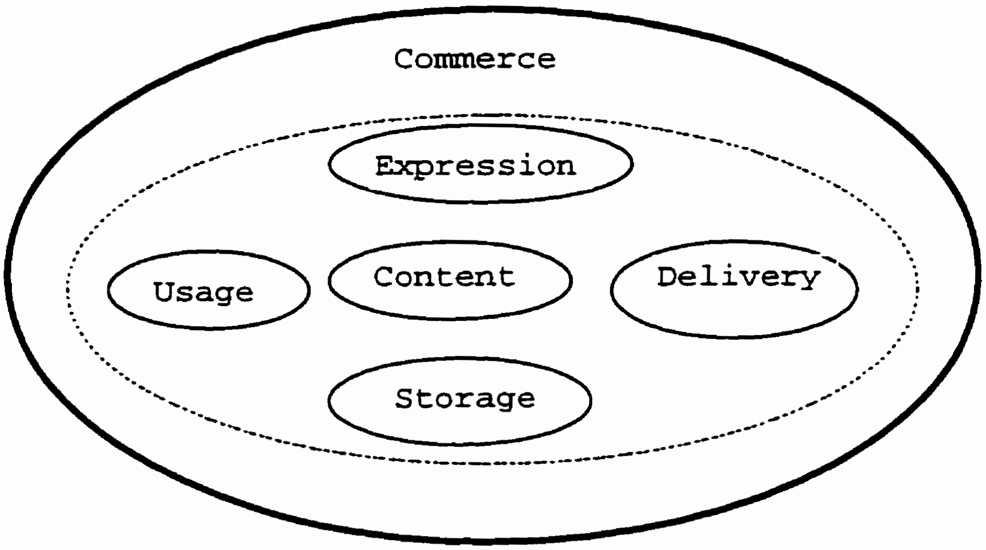
Another way to view this relationship is to conceive of a medium as a function of the six attributes:

Medium = F(Content, Expression, Commerce, Usage, Delivery, Storage)

Examples of each attribute are found in Table 2.1.1(p. 17).

Therefore a model comprised of content, expression, commerce, usage, delivery and storage is the foundation of the abstract media model. I refer to this model as the **Natural Media Model(NMM)**. The term natural indicates that it is the foundation model which will have another attribute added to it in order to create the Abstract Media Model(AMM). Figure 2.1.2 is a graphical representation of the NMM.

Figure 2.1.2: Natural Media Model



In Figure 2.1.2 I have placed " content" at the centre of the model just for the purposes of introducing and explaining the model. It does not imply that content is more important than any of the other elements. No attribute dominates and the structure has no logical centre or a fixed point of view as per the constraints of Section 1.6.

The commerce attribute is represented as enveloping the others because users of a medium purchase the whole, not only parts. For example you cannot go to a book store and only purchase a book's words, you have pay for everything that goes with the words; i.e., paper, binding, printing, shipping and layout. Therefore the commerce attribute must be applied to the entire medium package. At this point I will present some examples. I will use the NMM to represent a book and a web page.

Figure 2.1.3: NMM representation of a Book

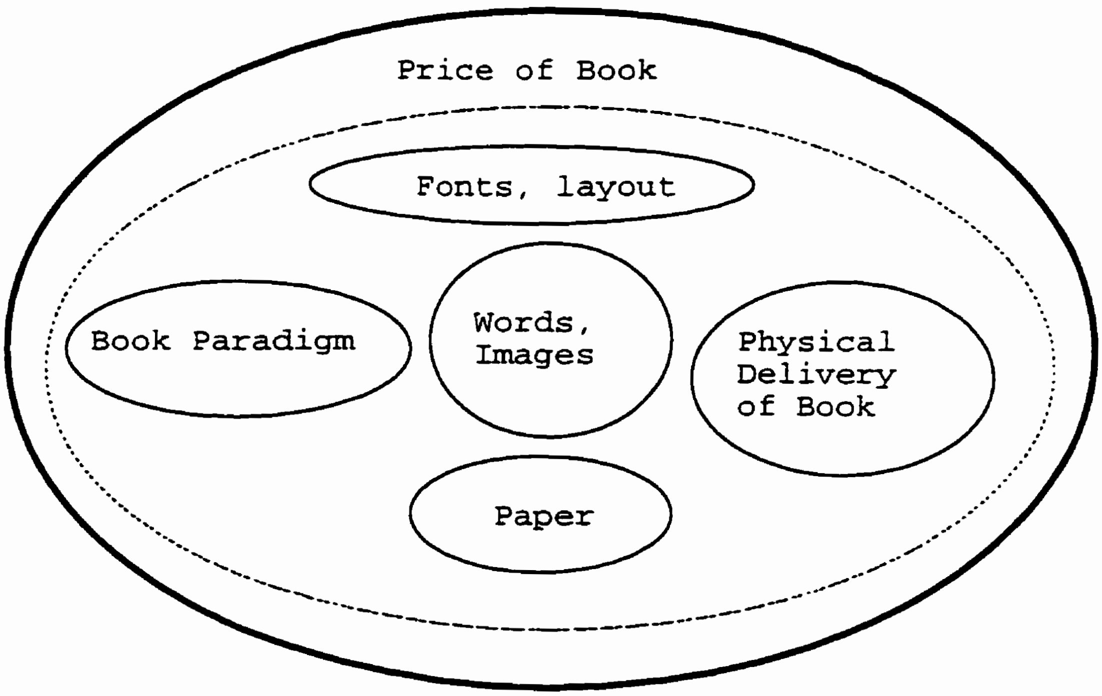


Figure 2.1.3 contains the NMM representation of a book. Note that Figure 2.1.3 is Figure 2.1.2 with each of the six attributes set to real values. In Figure 2.1.3 commerce is set

attributes set to real values. In Figure 2.1.3 commerce is set to the price of the book, expression is set to fonts and layout, delivery is the physical movement of the book to retailers and customers, content is the words and images that make up a book, the usage paradigm is, obviously, the book paradigm and the storage attribute is now set to paper.

Figure 2.1.4: NMM representation of a Web Page

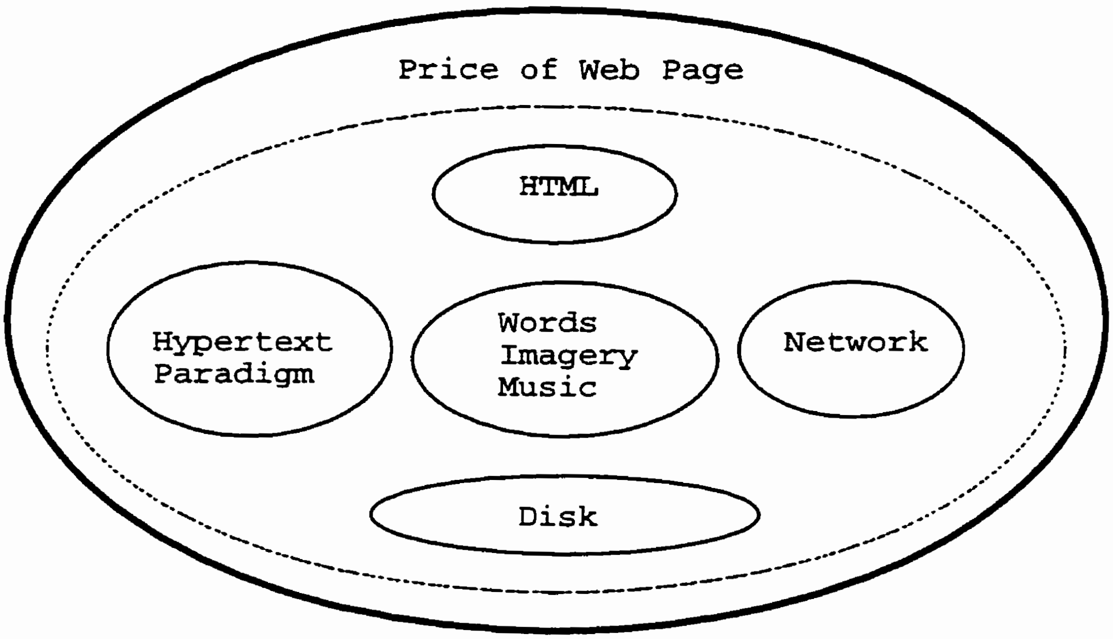


Figure 2.1.4 contains the NMM representation of a web page. As with Figure 2.1.3, Figure 2.1.4 is Figure 2.1.2 with each of the six attributes set to example values. In this case commerce is the price of the web page, expression is set to HTML, delivery is via the World Wide Web, content is the words, still and continuous imagery and audio that can make up a web page, the usage paradigm is the Hypertext paradigm and the storage attribute is now set to hard disk although it could be magnetic tape or a CD-ROM.

The role of the three above figures is to demonstrate how the attributes exist together to create the entity that I call the NMM. Starting the transition from the NMM to the AMM is the subject of the next section.

2.2) Explaining the Term " Abstract" in Abstract Media Model

In order to move from the natural media model to the abstract media model the term " abstract" must be explained.
" Abstract" is used to convey the notion that we are working with one general media type that I call abstract media. It provides a single term that includes all existing media types; i.e., books, films, photographs, Web pages etc.

There are three reasons for creating the abstract media concept. The first is derived from the observation that any medium can be considered as a stream. The only distinguishing factor is that either the stream or the perceiving entity moves. For example, text differs from film only in that film is made to move before our eyes whereas humans move their eyes to read the text. Therefore the difference between media types is not intrinsic but rather one of perception.

The second reason for creating the abstract media concept is to move media professionals away from only thinking about their particular medium(s) towards considering the broader media problem(s) and challenge(s) they face. With this broader, richer context they may see different ways to solve problems and face challenges.

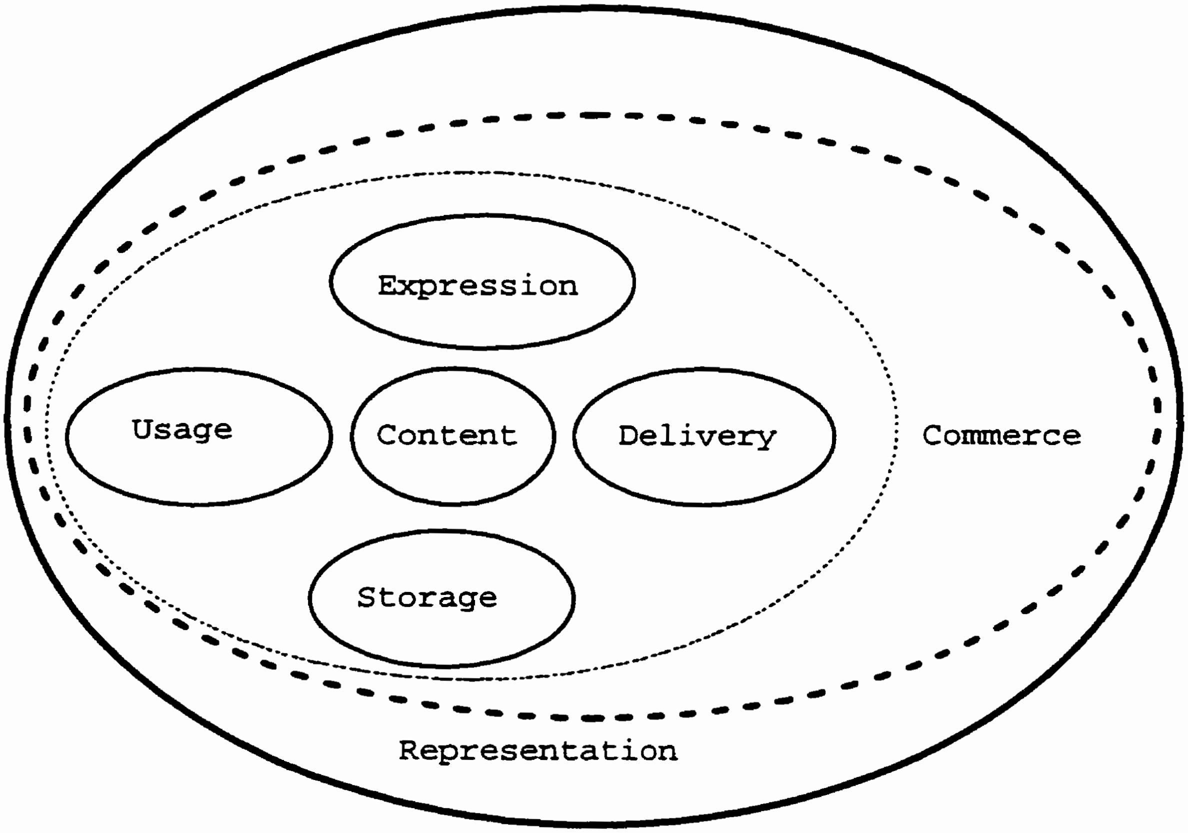
The third reason is that a concept such as abstract media allows the development of a single media representation. This means that specialised representations do not have to be developed for each of text, continuous imagery, audio or still imagery. By having one general media representation we no longer have to worry about integrating multiple media representations when creating multimedia. Rather multimedia can be developed from this one representation. Accordingly, I have developed a representation for abstract media known as the Abstract Media Code(AMC). Just as time code provided a representation for working with video, I hope the AMC will do the same for abstract media. The next chapter will explain the importance of a media representation and develop the AMC.

With the above definition of, and motivation for using, the term abstract media, a definitive description of The Abstract Media Model can now be provided. The Abstract Media Model is a model for working with and understanding abstract media where an abstract medium is comprised of the seven attributes: commerce, content, delivery, expression, representation, storage and usage.

2.3) Moving from the NMM to the AMM

The previous sections presented the NMM. In this section we move from the Natural Media Model to the Abstract Media Model(AMM). This will be accomplished by introducing one more media attribute: media representation. This attribute is not a fundamental element of a medium. A general media representation like the AMC is, however, becoming a required innovation given the ever increasing use of media and the creation of new media types.

Figure 2.3.1: Abstract Media Model



Given the content, expression, commerce, usage, delivery, and storage attributes, how do we access and work with these in a coherent fashion? A media representation is needed where the term representation is defined to be "... a set of conventions about how to describe a class of things."(Winston, p. 21). The 'class of things' in this case is media. A number of media representations have been devised. These include:

1) bit streams
2) words
3) objects
4) sequence of frames.

Which one of the above four alternatives is best is not the question as each representation will be appropriate under certain conditions. Rather what is important is recognising that the media representation attribute exists and has an important role to play just as the Artificial Intelligence community has demonstrated the importance of Knowledge Representation for general problem solving(Winston, 1984)(Newell, 1986). For example, a media representation has to: allow film makers to access content from an National Film Board(NFB) archive via a network, pay the royalty to the content owner and edit the acquired content into a new film. The representation entity's role and responsibility is to provide the representation necessary to access abstract media. It is the interface that content providers, application designers, technology providers and content users will utilise.

Multimedia has brought to the foreground the need for a generic representation structure that can be used to access and integrate media. The convergence of technology must be accompanied by a convergence in media representation. A uniform media representation will greatly aid users to work with the world's media bases.

A media representation must accommodate, address and access each of the AMM's constituent attributes. This means the representation is not independent of the other attributes. If you are an application designer, this attribute provides the media structure that will be used by your application to work with media. If you are a content provider the media you provide will be accessed using this attribute. If you are building a new type of compression/decompression engine, the engine should be designed to provide the services indicated by a media representation. For example, if the media representation attribute dictates that many resolutions can be chosen or that pixel level access is to be provided, the compression engine would have to be designed to provide these services. If media can be accessed at the pixel or byte level, then the media usage schemes, media storage mechanisms, media commerce entity and media delivery mechanisms must provide pixel level access. This example indicates how media representation has implications for each of the other attributes.

I will present a media representation in the next chapter. However my representation is only one example. There could be many representations. In this way another contribution of the thesis is a start to media representation. Just as Artificial Intelligence has devoted much energy to the development and investigation of knowledge representation, this thesis is providing an exemplar of the field of media representation.

2.3.1) Media and Knowledge Representation

Knowledge representation is a well defined field in Artificial Intelligence(AI). Luminaries like Newell, in his paper "The Knowledge Level", identified knowledge representation as a priority for the AI community. However all knowledge must be represented using some sort of media. For printed publications such as journal articles and books, text and graphics are the media used to express ideas. Films and videos are examples of a continuous medium being used as the expression vehicle. The point is that just as knowledge representation is an important research area, I propose that the vehicle used to present any knowledge is equally important. A media representation which can be used to access and integrate text, audio, still and continuous imagery will allow people to represent knowledge using a wider variety of media or at least make the task easier. This will give people who are attempting to present new ideas a richer set of media with which to express themselves. This richer set of media will also permit the conveyance and communication of more complex ideas. Until now most ideas have had to be expressed via words and diagrams. With multimedia technology, words, diagrams, music and 3D motion scenes can be used to present an idea. The World Wide Web currently provides some of the functionality needed to express ideas using the available traditional and new media.

This leads to the possibility that media representation is an independent entity. A good media representation is an essential first step for a good knowledge representation. The former must be in place for the latter to realise its full potential. Some ideas require more than text to be adequately presented and understood. However, if there is no mechanism for accessing media in a variety of forms, sources and locations, then the highly expressive medium is unlikely to be used. If this is the case then people will use traditional tools to express their ideas. The AMM's media representation attribute's goal is to provide a representation that will allow access to abstract media. Knowledge workers need a rich set of media to express their ideas. I will present a media representation called the Abstract Media Code in Chapter 3.

2.4) Attribute Interrelationships

The goal of this section is to present the interrelationships among the attributes of the AMM where each attribute can influence the other. The implication of each attribute impacting the other in a media-driven model will be clarified via examples.

If a person selects a usage paradigm, this choice necessarily implies certain storage, content, expression, delivery and representation attributes. If a standard book paradigm is selected then content is limited to text and still imagery. Storage of the content will be on paper. Expressions employed with this type of content stream have traditionally taken the form of different fonts and page layouts. The delivery channel will be the physical delivery of the books to the readers. The representation attribute will be set, in this case, to a page-based scheme. If the same book content is re-structured for a hypertext system, it may be dramatically different from the paragraph-chapter structure. For example under a hypertext paradigm one would not have to provide the bridging sentences and paragraphs as one goes from section to section, paragraph to paragraph and chapter to chapter. The reason is that the linking mechanism subsumes most if not all of the bridging work. Also under a Hypertext usage paradigm, a text-based piece of work can be augmented by still and motion imagery as well as audio and 3D-models. In this case the structure, amount and type of content will be altered by the usage paradigm.

The same situation applies if the World Wide Web(WWW) is used to deliver a course. In this case the content could be presented via words, still and moving images, and audio. Usage will be a Hypertext paradigm and the storage could be on a hard disk, CD-ROM or tape. Expression, both visible and non-visible, of the content is done via HTML. The delivery could be via a network or a standalone computer. Because of the use of multimedia, the Web causes multiple representations to be employed. For example, if video is used then a frame-based representation could be employed while the textual content could employ a traditional page representation scheme. Also, HTTP itself views everything as a byte stream hence a byte stream representation is also a possibility. Therefore usage will dictate content, storage, expression, delivery and representation attributes.

The Web uses the Internet as the delivery channel. However, because the Internet is currently bandwidth-limited, all Web content must be designed to display within this limitation. This is evident because many Web pages now offer, not only multimedia Web pages, but also text-only Web pages. In this example the actual content is changed to accommodate the delivery channel.

To further demonstrate the delivery channel impacting the content, the WWW's limited bandwidth can force changes to the structure of content. Instead of creating one large, twelve megabyte text file that may take fifty minutes to download, using a 28.8 kilobits per second modem, the author may break it into many small chunks which can be downloaded in, say, twenty seconds each. In this case the delivery system is forcing an alteration in the content structure.

The Dolby theatre surround sound system has four channels where three speakers at the front of the theatre will be used for dialogue, effects and music, while nine speakers will be placed around the theatre to provide surround sound effects. The audio engineer must mix the audio to exploit these delivery channels(Begault, 1994). This is an example of a delivery channel modifying the expressions applied to content.

The storage mechanism also impacts content. The development of affordable mass digital storage technology, such as hard disks and CD-ROMs, helped to make additional content available to content creators. CD-ROMs filled with clip art are commonly used by many writers. Not only has the storage system directly impacted content, it has also expanded the broadly accessible content base. However a storage mechanism can also constrain the quantity and type of content. For example since a CD-ROM holds only around 640 Megabytes of data, content creators may have to limit their use of, for example, digital video and use less space consuming content like text and digital audio.

Media expressions can also directly alter the content base. An example of this is special effects where a stream of video or audio is altered by adding the special effect to the content. The effect can be a video dissolve, a texture mapping or an audio filter. For example a digital audio filter known as a phase vocoder can vary the length of an audio segment(Dolson, 1986). An immediate application of this is for taped interviews that may fall short by fifteen seconds. In this case the audio filter can be applied to the original audio stream and a new audio stream results. The end result is that the original content stream is changed thereby demonstrating how media expressions can alter content. This example can also be represented via the NMM. The AMM could be used once media representation, the subject of the next chapter, is explained.

Table 2.4.1: NMM Representation of a Radio Interview

CONTENT	EXPRESSION	COMMERCE	USAGE	DELIVERY	STORAGE
Radio Interview	Analogue-to-Digital Converter Phase Vocoder	Media Costs Copyright Holder Market Price	Continuous Media Paradigm	Broadcast Internet	Hard Disk

Table 2.4.1 contains the NMM representation of a radio interview. In this case the content is the Radio Interview or the oral utterances that make up the interview. As I am assuming that the interview is being recorded digitally, an analogue-to-digital converter is a necessary expression. Also since the interview may need to be lengthened a phase vocoder could be utilised. The commerce attribute is set to the media costs, copyright holder and market price. The usage attribute is set to the continuous media paradigm. The Delivery mechanism can be via a broadcast medium or via the Internet. Finally the content will be stored on a hard disk.

The relationship of the commerce attribute to the others, I believe, must be addressed on a different level of granularity. The commerce attribute must be applied to the subsets of attributes found in the dashed ellipse of Figure 2.1.2. The reason for this relationship structure is the nature of media. Consumers buy books and videos. The complete integrated entity is purchased. The attributes of content, storage, usage, expression and delivery are not bought separately by the media consumer, at least not yet. For this reason, I show the commerce attribute as being related to the others taken as a whole. This all encompassing relationship is depicted in Figure 2.1.2. For example, just as content can have commercial value, the fact that monetary remuneration is possible may cause content to be produced. A video may be made instead of a film as the former is cheaper than the latter to produce. In this example, commerce determines the type of content that is made. A publication may be electronically stored and delivered because the cost of newsprint may make a traditional book or magazine uneconomical. In a bandwidth limited world where a user pays on a per-byte basis, the representation scheme utilised may be that representation which minimises transmission costs; i.e., transmits the fewest bytes. In this case commerce can impact the choice of representation schemes. Therefore commerce's influence is not limited to content. The delivery channel, storage method, expression, usage and representation attributes can all be influenced.

A matrix representation of the notion that each attribute can influence the other, is found in Figure 2.4.1. As is evident from the preceding discussion each attribute can have a wide and diverse influence. However representing the complete impact that each of these attributes will have on each other is difficult. How does one measure the effect of using paper as the delivery mechanism on the writer's creative process and therefore content? Has the content creator already taken into account the implications of using paper? While these are difficult questions to answer, the AMM will provide a framework that can be used to address, investigate and resolve a variety of relationship structures among the attributes.

Figure 2.4.1: Interrelationships

	CONTENT	EXPRESS	COMMERCE	USAGE	DELIVERY	STORAGE	REPRE
CONTENT	C	ICE	ICCM	ICU	ICD	ICS	ICR
EXPRESS	IEC	E	IECM	IEU	IED	IES	IER
COMMERCE	ICMC	ICME	CM	ICMU	ICMD	ICMS	ICMR
USAGE	IUC	IUE	IUCM	U	IUD	IUS	IUR
DELIVERY	IDC	IDE	IDCM	IDU	D	IDS	IDR
STORAGE	ISC	ISE	ISCM	ISU	ISD	S	ISR
REPRE	IRC	IRE	IRCM	IRU	IRD	IRS	R

Figure 2.4.1 represents the interrelationships between the attributes that make up the AMM. The notation IDC is used to represent the interrelationship between delivery and content. The interrelationship between commerce and usage is represented by ICMU. The main diagonal only contains each attribute. The reason is that the interrelationship between each attribute and itself is the attribute.

This matrix is used to concisely represent the preceding interrelationship discussion. For example I discussed above how the interrelationship between content and usage is different from the interrelationship between usage and content. As discussed above once the usage paradigm is chosen this imposes constraints on the content. For example the standard book usage paradigm implies text and still imagery. However if a hypertext usage paradigm is selected the idea could be communicated via text, still and continuous imagery, audio and virtual worlds. Therefore the book usage paradigm limits the possible content choices. This shows the impact of the usage paradigm. If there was no such impact of the usage paradigm or delivery mechanism or representation scheme on content then this matrix would be symmetrical; i.e., ICU equals IUC, IDC equals ICD and IRC equals ICR. However as the preceding discussion has indicated this matrix is not symmetrical.

2.5) AMM: An Operational Representation

This section has the goal of providing a quickly accessible version of the AMM. This will be accomplished by providing an operational representation of the AMM and an operational specification of the AMM. The two taken together will provide the quickest method of accessing and utilising the AMM.

Figure 2.5.1: AMM Operational Representation

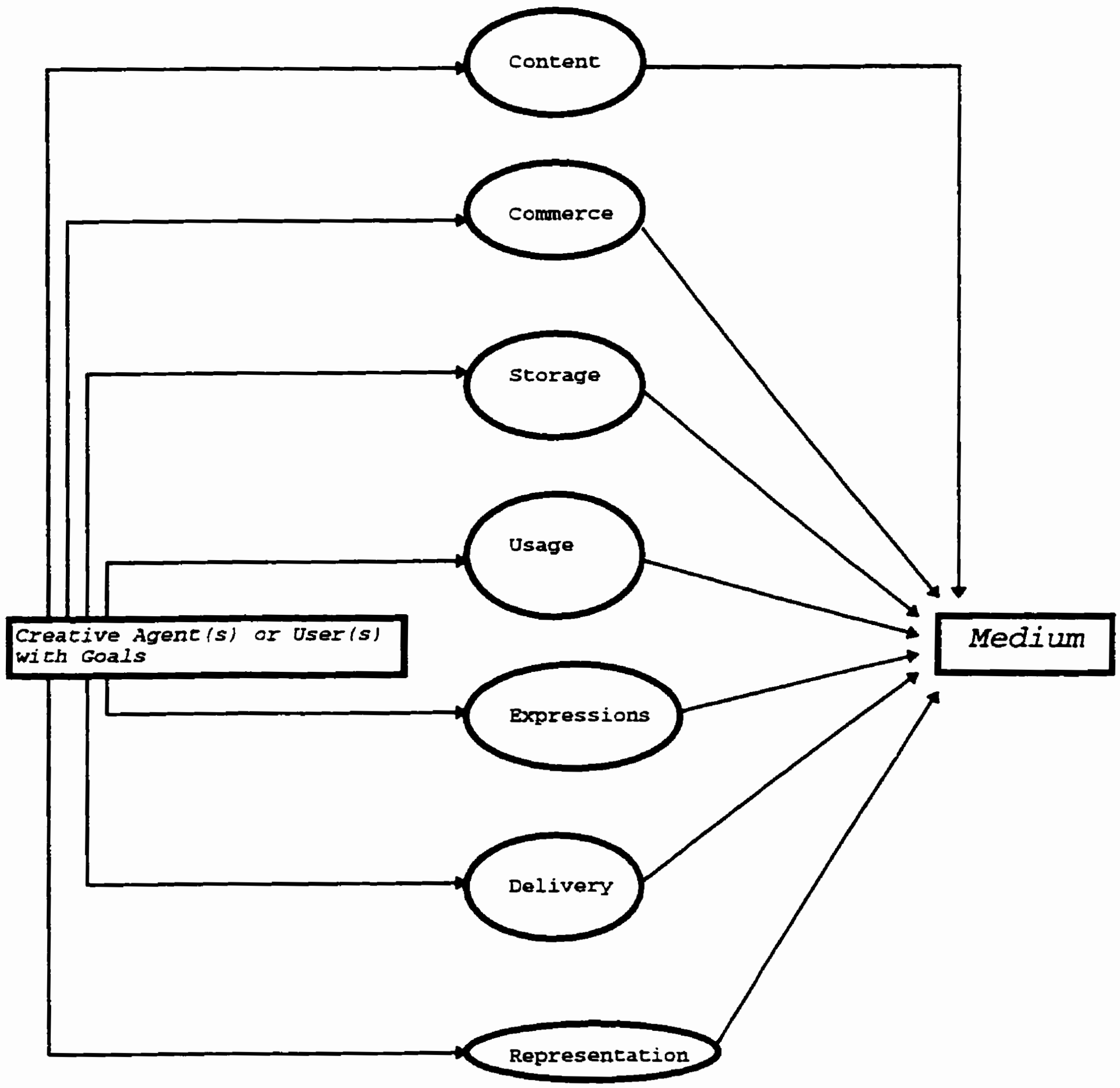


Figure 2.5.1 is another representation of the AMM. It shows the role that each attribute plays in the process of creating a medium including the creative agent(s) or user(s). As indicated in Figure 2.5.1 the creative agents or users have goals. These goals can range from performing a

strategic analysis of an organisation's media holdings, to creating a book, to the initial design of a device driver for a new type of digital audio board. Figure 2.5.1 provides a visual framework that the creative agents or users can use when attempting to realise their goals via the AMM.

The role of Table 2.5.1, below, is to help specify the operational representation of Figure 2.5.1.

Table 2.5.1: AMM Operational Specification

CONTENT = {text, still imagery, continuous imagery, audio, virtual environments}

EXPRESSION = { Fonts, Page Layout, Time Code, SGML, HTML, VRML, XML, Databases, Alphabets, Compression, ASCII}

COMMERCE = {content cost, expression cost, commerce cost, usage cost, storage cost, delivery cost, market price, copyright holder}

USAGE = {book paradigm, continuous media paradigm, still image paradigm, hypertext paradigm, direct manipulation paradigm, binary usage paradigm}

DELIVERY = {network, broadcast, physical delivery}

STORAGE = {paper, hard disk, tape, magneto-optical}

REPRESENTATION = {words, frames, objects, bit streams, AMC}

Table 2.5.1, a re-specification of Table 2.1.1, contains the possible values for each of the seven attributes. These values should be used to populate the operational representation of Figure 2.5.1. The possible values presented for each attribute are not meant to be exhaustive or static. As new technologies, paradigms and content possibilities are developed, these would have to be included. Furthermore the list of possible expressions will change subject to the application area. This version provides a starting point.

2.6) Conclusions

The goal of this chapter has been to present the AMM. This media model has always implicitly existed. It has been embedded in media and, therefore, has not been evident. This chapter's role was to make the model explicit. The AMM, by representing media as composed of seven attributes, provides a further fragmentation and decontextualisation of media where the Greek alphabet was an early example. The process of fragmenting and decontextualising is important as it is the basis for recombination which presents opportunities for technological, social and cultural innovation(De Kerckhove, 1995, p. 201). For example philosophy, science, religion and law are disciplines which present us with knowledge and information. Each of these disciplines was constructed out of words by the human mind(Frye, 1993, p. 66). They were built by minds decontextualising and fragmenting the words of the existing knowledge base so that they could be reformulated into new knowledge/information constructs. The AMM, by fragmenting and decontextualising the attributes that make up media, hopes to perform a similar function whereby the different attributes could be recombined to create new media which could in turn support the expression of new and different ideas, structures and disciplines.

Chapter 3: Media Representation, the Abstract Media Code.

This chapter's goal is to describe the development of a specific media representation. This representation, called the Abstract Media Code(AMC), is important as it is the mechanism for users to work with the media.

As mentioned in Section 2.2 a media representation must accommodate, address and access each of the AMM's constituent attributes. In order to accomplish this requirement I propose to extend an existing representation: time code. The media industry has been using this representation for many years. Context information on time code is the subject of the following section.

3.1) Time Code

Time code is an international standard supported across North America and Europe by the Society of Motion Picture and Television Engineers(SMPTE) and the European Broadcasting Union(EBU)(Fortner, 1992). It is basically an electronic signal that provides a specific and unique address for each electronic frame. This address allows users to locate and access any frame on the video tape. Each video frame, where "frame " is a term borrowed from the film industry, has a unique identifying number known as the time code address or number. This number has the form 00:00:00:00 where the successive pairs of digits represent hours, minutes, seconds and frames. The number of frames that comprise, for example, a two-hour video depends on which type of television broadcast transmission system is being used. Currently there are three major formats in common use.

Table 3.0 shows the frame rate of the major transmission systems. Note that film travels at 24 frames per second.

Table 3.0: Frames Per Second(fps)

National Television System Committee(NTSC)	30 fps
Phase Alteration by Line(PAL)	25 fps
Systeme Electronique Couleur Avec Memoire(SECAM)	25 fps

The NTSC system is used in Canada, Japan and the United States. The PAL standard is used by England, Holland and West Germany. SECAM is used by the former Soviet Union and France. Television sets capable of functioning with all three standards now exist(Fortner, 1992).

Video tape recording was introduced in the mid 1950s. To edit video tape people initially tried to use the "cut and splice" techniques of the film industry. However video tape images, unlike film, are not visible as they are electrical impulses on the tape. This led to the 1960s work on developing the electronic equivalent to film editing(Lehtinen, 1994), (Zettl, 1992), (Fortner, 1992). The new edit system identified each video frame by the use of an electronic code or time code. It allowed the electronic searching of video tape for frame locations so that editing could take place. In 1967 the first practical system appeared. It adopted a time code system similar to the one developed by NASA during the GEMINI and APOLLO space programs. NASA used time code to tag satellite telemetry tapes(Fortner,1992). Time code was originally called EECO code for the company that developed it. The Society of Motion Picture And Television Engineers eventually standardised the time code system and renamed it SMPTE time code(Lehtinen, 1994). Time code is also used by the European Broadcasting Union so sometimes it is referred to as SMPTE/EBU time code.

It is recorded in two different ways. The first is Longitudinal Time Code(LTC). This type of time code is recorded on the linear audio track that travels the length of the video tape. The second type is Vertical Interval Time Code(VITC). VITC is recorded in one of the unused lines of the vertical interval, along with the video signal (Lehtinen,1994). In addition to different recording methods, there are two different formats of SMPTE time code: drop frame and non-drop frame. These formats exist because a way had to be found to make "...the colour TV broadcasts compatible with all the existing black-and-white TV sets..." (Rona, 1990, p. 71). In order to introduce the colour signal, the frame rate had to be slowed to 29.97 frames per second. The implication of this is that a per second frame rate of 29.97 causes time code to be slightly slower than real time, approximately 3.6 seconds longer than real time per hour. This may seem insignificant but in the TV broadcast world, since advertisers pay for real "air " time, programming must reflect real time. The solution to this problem was the introduction of "drop frame time code". As its name implies, "drop frame time code " drops frames. It skips frame number 0 and 1 at the beginning of every minute that is not a multiple of ten(Rona, 1990, p. 72). This causes the time code to accurately mimic real time. Time code is used extensively in the film/TV/audio industries as an editing and synchronisation tool. It is a fundamental enabling technology for the media industries. The next section will present my version of time code and my media representation known as Abstract Media Code(AMC).

3.2) Abstract Media Code(AMC)

Time code is currently represented as HH:MM:SS:FF where HH is hours, MM is minutes, SS is seconds and FF is frames. The AMC expands basic time code to get the extra representation power. It is expanded to accommodate each attribute of the model so that time code, which is a media expression, becomes part of a more generalised representation found in (1) below.

(1):USAGE:DELIVERY:STORAGE:CONTENT:EXPRESSION:COMMERCE

(1) is the totally general representation that contains each attribute of the AMM, including the representation attribute. (1) itself is the representation attribute. Further refining (1) to make explicit the inclusion of Extended Time Code(ETC) generates the AMC:

(1a) :USAGE:DELIVERY:STORAGE:CONTENT:[Exp 1, ..., Exp n,ETC]:COMMERCE

The AMC of (1a) differs from (1) in that the expression attribute has been replaced by [Exp 1,..., Exp n, ETC]. This syntax indicates that Extended Time Code(ETC), along with any comma-separated list of expression(s)(Exp), will make up the expression term. The motivation for the time code extensions is provided by the journalistic practice of answering the five following questions: Who, What, Where, When, and Why(W5). Please remember the constraints of Section 1.6 which specified that the model must be media driven and function in the analogue and digital worlds. This is why the technologies of time code and the five Ws are borrowed, as both of these play important roles in the analogue and digital media industry.

3.2.1) AMC: The Operational Specification

This section presents the operational AMC. It is called this because it will explicitly define what should be placed in each of the AMC's elements.

The ordering of the attributes in (1a) follows the ordering of the URL used to access content on the WWW; i.e., usage:delivery:storage:content:expression or http(usage)://address(network delivery)/filename(content and storage).html(expression). Therefore I will follow this ordering convention for the AMC. The instantiation of every element of (1a) is not mandatory. For example in a traditional analogue tape editing environment the AMC could be comprised of only one expression, time code. The other attributes are optional.

Table 3.2.1: AMC, Operational Specifications

USAGE = {book paradigm, continuous media paradigm, still image paradigm, hypertext paradigm, direct manipulation paradigm, binary usage paradigm}

DELIVERY = {network(agent), broadcast(agent), physical delivery(agent)}

STORAGE = {hard disk, CD-ROM, tape, optical disk, paper }

CONTENT = {Content Title}

EXPRESSION = { number of expressions, comma-separated list of expressions, ETC}

ETC = {*who* -content creator.
what -content type; i.e., audio, continuous imagery, text, virtual environment.
where-geographic location given as degrees longitude, minutes and seconds and degrees latitude minutes and seconds.
when -time code given as year, month, day, hour, minute, second, frame.
why -description field.}

COMMERCE = {copyright holder, market price}

Table 3.2.1 contains the set of possible tokens for constructing the AMC. The specification provided in Table 3.2.1 is not static. It can be expanded, for example, as new storage devices become available and new delivery channels are developed. The AMC is not a static entity but has been

developed as a representation capable of accommodating new developments, different situations and applications.
In Table 3.2.1 each element of the delivery attribute was followed by the word agent in parentheses. This is used to convey the fact that there are many possible delivery networks, broadcast channels and physical delivery methods. For example two possible network delivery examples are Compuserve and the Internet. If you consider the WWW as different from the Internet then there are three network delivery possibilities. In Canada alone there are several broadcast channels available. The possibilities include but are not limited to CBC, CTV, Global, TSN and Rogers to name a few. Finally the physical delivery of media can be performed in a variety of ways via a number of agents including: stores, couriers and shippers of all sorts.

The expression element was not specified in Table 3.2.1. The reason for this is that I do not know all the possible expressions. For a list of possible expressions please refer to Table 2.1.1(p. 50). Finally the "when " element of the ETC will be defined as the creation date of content or media. For example the "when " element would indicate when a piece of video was shot, a picture taken or a painting completed. Given this operational version of the AMC, it will now be applied in the following series of examples.

3.2.2) AMC Examples

The AMC is readily applicable to the analogue world. For example the purchasing of a text book at the University of Waterloo(UW) bookstore can be represented via the AMC:

(i) ::UW: paper: "Network Computing ": Lucida Sans Typew:$C80.00

In example (i) the delivery of the text book where the text is stored on paper is via the UW bookstore which is acting as the delivery agent. The text book, "Network Computing ", was printed using the Lucida Sans Typew font. Its cost is $80.00. Only the usage attribute remains unspecified because everyone knows that a book uses the book usage paradigm. It could however be included, in which case (i) would become:

(ia) : bp: UW: paper: "Network Computing ": Lucida Sans Typew:$C80.00

where bp denotes the "book usage paradigm ".

Similarly the purchase of the CD-ROM "Return To Forever " by the Jazz Musician, Chick Corea, at the local HMV can be represented as:

(ii) : cmp: HMV: CD-ROM: "Return To Forever "::$C30.00

where cmp represents the continuous medium usage paradigm. The representations (i) and (ii) are examples of the AMC used for simple content purchases. They could be used in sales ledgers or an advertising supplement or point-of-sale system.

In the next example a video tape editor receives a tape with the following AMC information on the label:

(iii) :cmp: Courier Name: Video Tape: "Ontario Nature Footage": 00,00,00,00-00,12,30,15(ND) ::

In example (iii) the video tape with the Ontario Nature Footage was delivered by a courier. In this example the expression is a traditional time code range indicating that Ontario Nature Footage is twelve minutes, thirty seconds and fifteen frames in length and that non-drop frame time code was used.

The AMC's role is to provide representation for all media types. To realise the goal, traditional time code can be applied to text. Instead of using page numbers, a time code number could be given to each page where a page could represent a frame. This means that text could be synchronised with the other traditional continuous media via the established mechanism of time code and one media representation. Text differs from film and video in that the latter two are moved before the viewer's eyes, whereas the former requires that readers move their eyes while the text remains still. Therefore example (i) could be re-expressed as:

(iv) :bp: UW: paper: "Network Computing ": 00,00,00,00-
00,00,10,00(ND) :$C80.00

Example (iv) now has the number of pages in a text book. In this case the text book is, by using time code, ten seconds in length or, since non-drop frame time code is being used, it is 300 pages long as there are 30 pages per second and ten seconds in total. Please remember that the reason for setting a page equal to a frame is the goal of using one media representation; i.e., the AMC.

Time code could be applied to a painting or a photograph or a map. In the case of a sketch the AMC would be:

(v) ::Art Gallery of Ontario: paper: "Untitled Work " : 1928,12,22,00,00,00,00: $C2Million

Example (v) has the Art Gallery of Ontario showing an Untitled Work worth $C2Million dollars. It has a time code of 1928,12,22,00,00,00,00. The traditional time code of 0 hours, 0 minutes, 0 seconds and 0 frames has in this case been expanded to include the date, Dec 12 1928. This is the "when " element of the ETC and provides the cre ation date of the "Untitled Work ".

In the next example, all the elements of the ETC are populated. Suppose Videographer Bob White shoots-(with a video camera)-children running at City X's City Hall because City X's City Hall has a very good play ground. In this case the AMC would be:

(vi) :cmp:: video tape: "City Hall Play Ground ": Bob White, "video ", 70,00,00-40,00,12, 1996,07, 01,13,15,00,00-1996,07,01,13,15,00,09,09, "Example of Good play ground": CBC

In Example (vi) the delivery attribute is unspecified and the commerce attribute is partially specified as it only notes the content owner, which in this case is the CBC, and not the content price. The content is stored on video tape and the content title is "City Hall Play Ground ". The content creator is Bob White(Who). "video " is the 'What'. 70,00,00 - 40,00,12 is the geographic location of City X in terms of degrees latitude, minutes and seconds and degrees longitude, minutes, seconds(Where). The content was shot starting at 13 hours, 15 minutes on July 1, 1996 and lasts 9 seconds and 15 frames or 9.5 seconds(When). "Example of Good Play Ground" is why the content was shot. The capturing of the geographic location could be accomplished via camera mounted Global Positioning Systems(GPS) which, in their military versions, can track positions with centimetre accuracy.

Suppose after some time the CBC has digitised the content and made it available via a link on its Web page. In this case the AMC would be:

(vii) :hypertext: www.cbc.ca: hard disk: /home/content/City_Hall_Play_Ground: mpg, Bob White, "Children running at City Hall ", 70,00,00-40,00,12, 1996,13,15,00,00,00,00- 1996,13,21,12,00,00,09,09, "Example of Good play ground": CBC,$C0

In example (vii) the usage paradigm is hypertext, the delivery of the content is via the WWW-(network delivery)- as indicated by www.cbc.ca. The storage of the content is on a hard disk and the content is the file /home/content/City_Hall_Play_Ground. The expressions for this digital version include MPEG encoding and the ETC as described above. The content owner is the CBC and it is free.

This example is also interesting in that it demonstrates how the AMC could enable the Web to become a distributed edit suite, as well as a content repository. Please note that in HTTP v1.1, the specification of content byte ranges is supported. Perhaps in HTTP v1.2 Extended Time Code or a mechanism for translating the byte ranges into ETC will be available. This is important as it would allow the integration and co-operation of the computing and media creation industries via the use of a common representation scheme.

To summarise, the specification for the AMC will be:

(viii) :USAGE:DELIVERY:STORAGE:CONTENT:[Exp 1, ..., Exp n,ETC]:COMMERCE

where ETC will be expressed via "who ", "what ", "where ", "when ", "why " and the COMMERCE attribute will be composed of content owner and price. The ETC could accommodate virtual environments via the addition of extra dimensions to the "where " attribute. This could allow accessing content found in virtual reality environments down to measures as refined as individual pixels. Also the ETC could be employed by users to search for content. For example instead of searching for a filename like dog.mpg, users could base their searches on some or all of the five W's. This type of searching mechanism based on content description is now possible with the Portable Network Graphics(PNG) standard. For a complete description of PNG and its capabilities please refer to http://www.w3.org/pub/WWW/Graphics/PNG . Note that in Chapter 5 I will present a Neural Database System I developed where its training and query patterns are based on the AMC.

The application of time code to video, text and still imagery is another example of why I want to use the term "abstract media ". It is a neutral term moving the focus away from the traditional representations associated with particular kinds of media and allowing a new representation that can be used within the context of a comprehensive media framework capable of supporting multimedia. Finally note that the representation attribute is another element of media cost. It takes time and resources to choose and implement a media representation which ultimately adds to the cost of media.

3.3) AI Representations and the AMC

As previously noted, the AMC is a representation that encompasses the other attributes of the model. It also meets the other requirements of a good representation as defined in the AI literature. For example Winston(1984, p. 24) identified a good representation as having the following characteristics:

- Good representations make the important things explicit
- They expose natural constraints, facilitating some class of computations.
- They are complete. We can say all that needs to be said.
- They are concise. We can say things efficiently.
- They are transparent to us. We can understand what has been said.
- They facilitate computation. We can store and retrieve information rapidly.
- They suppress detail. We can keep rarely used information out of sight, but we can still get to it when necessary.
- They are computable by an existing procedure.

The AMC makes important things explicit. It facilitates working with abstract media. It can make the natural constraints of the media explicit. For example film requires a playback speed of 24 frames per second. This can be incorporated into the AMC. I am not sure nor do I claim that the AMC is complete. It is however easily expandable. The AMC is concise, precise and compatible as it uses media terminology and measurement that are standardised across large agencies, processes and industries. It is also an accessible media representation given that a design constraint was the bridging of the analogue and digital realities. The AMC provides ample storage indices which can be used to store and retrieve content. The AMC in its current form can be used in a very simple fashion so that a user does not have to access or provide very much information. It can however provide a lot of detail. The AMC is capable of being manipulated and accessed by the current computer technology. For example it can work in conjunction with the OMFI, MHEG, HyTime and QuickTime. It could also represent possible future changes and enhancements to SMPTE time code.

3.4) Conclusions

This chapter has presented the AMC and how it is derived from and based in the media industry. Its form-(see example (viii) above)- has been borrowed from the analogue and digital worlds. Its possible applications beyond providing the entry mechanism into the AMM include: future enhancements to SMPTE time code, the generation of interesting storage indices for multimedia databases, providing a frame work for future content usage and supplying the interface for a multimedia filesystem. It could also be encoded directly into an MPEG video stream. This chapter marks the end of the theoretical discussions. Chapter 4 demonstrates how the AMM can be used as an analysis and design tool while Chapter 5 contains an application of the AMM.

Chapter 4: Analysis and Design using The AMM

This chapter will demonstrate how the AMM can be used as an analysis and design tool. It will be used to analyse a number of existing analogue and digital media and technologies. It will then be employed as a design tool. Note that Table 4.1.1 has seven attributes compared to the six of Table 2.1.1(p. 17). The reason for this is that the representation attribute, first introduced in Chapter 2 and fully developed in Chapter 3, has been sufficiently motivated and explained so that it can now be used. Note that the representation attribute is the difference between the NMM and the AMM.

4.1) The AMM and Books

Table 4.1.1: AMM and Books

CONTENT	EXPRESS.	COMMERCE	USAGE	DELIVERY	STORAGE	REPR.
Text	Fonts	Media Costs	Book Paradigm	Physical Movement of Book	Paper	words
Still Imagery	Page Layout	Market Price				
	Mark-up	Copyright Holder				

Table 4.1.1 contains the representation of a book in terms of the AMM. Please note that EXPRESS. is the abbreviation for EXPRESSION that I will be using and REPR. is the abbreviation for REPRESENTATION. The content of a book can include text and still imagery. A book employs several expressions including Fonts, Page Layout and Mark-ups. The commerce attribute of a book is multidimensional. It includes:

- media costs associated with the book
- the copyright holder
- the market price of the book.

The usage attribute of a book is a book paradigm where, for example, in Western cultures, a book is composed of pages read from top left to lower right and each page follows the other. The delivery mechanism for a book is via the physical movement of the book to the reader. The content is stored on paper and the representation is pages.

4.2) The AMM and Videos

Table 4.2.1: AMM and Videos

CONTENT	EXPRESS.	COMMERCE	USAGE	DELIVERY	STORAGE	REPR.
Analogue Continuous	Time Code	Media Costs	Continuous media	Physical Movement of	Tape	Frames
Imagery	Special Effects	Market Price	Paradigm	Tape		
Analogue Audio		Copyright Holder				

Video cassettes can be purchased or rented in corner stores and dedicated video stores. In terms of the AMM a video tape's content can include continuous imagery and audio. Its expressions include time code and special effects. Time code is added to enable editing and therefore content creation. Just as for books the commerce attribute is populated with copyright holder, media costs and market pricing where the latter can be either outright purchase or rental. The usage paradigm is that of continuous media; i.e., play, stop, fast forward and rewind. Delivery of the video is again the physical movement of the tape. The content is stored on tape and the representation is frames.

These two sections on books and video cassettes are examples of how the AMM can be used as an analysis tool. In both these cases each medium is expressed in terms of the model.

4.3) AMM and Digital Libraries: A Preliminary Design

The goal of this section is to perform a preliminary design of a digital library using the AMM. I will just assume that all the cost-benefit analysis has been performed and that a community has decided to build a digital library.

The first issue to be decided is the representation to be used for accessing the digital library. This representation will have to perform multiple purposes. These purposes include:

- allow the media to be accessed and searched in a networked and non-networked environment

- allow the media to be used for a variety of purposes beyond just the traditional uses associated with a library

- provide a bridge to the content of traditional/non-digital libraries.

The representation scheme I propose is the AMC with a major classification scheme such as Dewey Decimal, Library of Congress, or INSPEC thesaurus as an expression. In this way all the attractive elements of the AMC could be utilised and a classification system could be introduced. I mentioned that a link would be needed to the traditional analogue libraries. This is especially important given the experience of both England and France who are building new national libraries. In both cases the cost of the buildings is less than the costs associated with digitising the collections(Lesk, 1994, p. 655). The American Library of Congress has received $13 million to digitise its American historical collections(Becker, 1994). Therefore, given the huge digitisation costs, a link to traditional libraries is necessary.

The next issue to be addressed will be that of commerce. How will authors, publishers and other inhabitants of the media food tree, including the library itself, collect their fees? This issue has to be solved up front and designed right into the representation scheme which gives access to a collection. In the case of media which is owned by the public this is not an issue. However the digital library could make available content from private collections that would not be accessible to the general population in a traditional library. For example assume that a collector has a rare Beatles song. The collector could receive a fee each time the song is accessed. This would give the collector an incentive to make the song available. Therefore the issues introduced by the commerce attribute of the AMM must be addressed immediately.

The library will have to support a variety of usage paradigms. The reason for this is that the library contains the complete spectrum of media. Therefore a complete set of usage attributes will be present.

The next issue to be decided is that of delivery. A digital library will have to decide how it will deliver its collection. Will access to material be local only or will network access be possible? For the purposes of this section I will make a design decision that the library will provide local and network access with emphasis on network access. The reason for this emphasis is an economic decision to minimise the amount of resources placed into bricks and mortar. Once this decision is made then the network infrastructure has to be designed as this will be the major way into the media collection. At this point work from actual digital libraries could be utilised. For example the Stanford Digital Library Project is developing the "Information Bus "(The Stanford Digital Libraries Group, 1995). This work could be utilised to provide the digital library's media delivery.

The digital library will have to support a number of media expressions. Given the electronic and network based aspect of the library, SGML and its sibling HTML would be recommended mark-up schemes. In addition to these, other mark-up schemes which support intelligent querying and, say, printing would be desirable. Candidates for these last two roles are, respectively, the Agent Communication Language(Genesereth, Ketchpel, 1994) and PostScript which is a page-description language. The former could be used to implement intelligent querying while the latter would support the printing of attractive documents. All these mark-up schemes would be stored separately from the content. This type of architecture would allow multiple mark-ups to be applied to the same content. For example a piece of text could at one instant have HTML applied to it so that it is accessible to the browsers currently inhabiting the Internet. The same piece of text could then have PostScript applied to it in order to get an attractive print version.

Indeed this digital library could be a test site for the distributed mark-up concept I will introduce in Section 4.9.1.

Storage and all attendant issues will be one of the most important issues facing any digital library. While the digitising costs will be substantial, storage of all the digitised material constitutes another major expense. The first task is to decide which storage device should be used. Just as the ancient Egyptians had to chose either parchment, papyrus, wax or stone(Ullman, 1969) to store their content, the same problem exists today as existed in the fourth century B.C.. However now the choices include tape, CD-ROMS, disk drives, solid state ram drives and potentially holograms. Therefore given the multiplicity of choices, how would a digital library choose a storage device? This decision is subject to many constraints including: cost, security, access time and durability. Table 4.3.1 is provided as an aid to help make this decision.

Table 4.3.1: Storage Device Comparison

TECHNOLOGY	**ACCESS TIME**	**DURABILITY**	**SECURITY**	**COST**
TAPE	VERY SLOW	DURABLE	SECURE	VERY CHEAP
DISK	FAST	DURABLE	SECURE	EXPENSIVE
MAGNETO OPTICAL	SLOW	VERY DURABLE	VERY SECURE	CHEAP
SOLID STATE	VERY FAST	?	?	VERY EXPENSIVE

This table's role is to help decision makers choose a storage device. Each technology has a number of strengths and weaknesses. For example, tape is a relatively cheap storage device that has a relatively slow access time. Disk drives have much faster access times but are also far more expensive. Currently disk drives, on a per megabyte basis, are ten times more expensive than tape. The message from this table is that the appropriate storage device will be determined by the applications needs. For example if high security is an absolute requirement, then a CD-ROM drive would be a good choice as it cannot be erased and can last a relatively long time. However if cheap access to a vast quantity of data is needed then perhaps a tape based storage solution is the answer. In certain instances a combination of storage media is necessary. For example if you run a large production facility with terabytes of content then a mixture of tape and disk drive or solid state RAM drives may be the best solution. Tape would be used to store content and the disk drives would be used to edit the content when necessary. However this would require the computing power to swap content between the slower medium, tape in this case, to the disk drives. Regardless of which storage device is chosen the issue of technological obsolescence must be flagged. The reason for this warning note is that the chosen storage device may become incompatible with the other components of the storage system. This means a user may not be able to access stored media. Therefore an anti-obsolescence cost has to be worked into the equation(Adelstein, 1996).

In addition to the storage issue, reducing the amount of space used to store media must be addressed. This is where the concept of compression arises. Again delving into history, compression technology was employed as early as the fourth century A.D. when a slanting script was used to save time and space on expensive parchment(Ullman, 1969). Similarly, we must deal with the same issue today; i.e., put more content on less space. Note the most effective compression technology from a quality of service point of view is one which gives the user an infinite number of resolutions to choose from. The infinite number of resolutions is exploited by the decoder technology which is capable of giving the user the demanded resolution. This is more effective from a storage point of view since the one stream is capable of generating many resolutions instead of having to store all the desired resolutions. Compression raises a number of other issues. What functionality should the decoder have? Is the compression scheme appropriate in a bandwidth constrained network setting? Is the compression scheme capable of delivering to people only what they want? For example can a frame differencing scheme like MPEG give users access to the desired frame without them having to play through all frames preceding the one they desire? This becomes especially important in a pay-per-byte world.

While Table 4.3.1 highlights the issues associated with choosing a physical storage device, Table 4.3.2 presents a cost comparison of digital and analogue storage. This is especially relevant to digital libraries which may not be able to initially afford a completely digital collection.

Table 4.3.2: Storage Costs

	Real Estate	**Special Facilities**	**Personnel**	**Preservation**
Analogue	high	high	high	extremely high
Digital	lower	lower	high	extremely high

Table 4.3.2 presents some of the storage costs. These include real estate costs such as physical space requirements, insurance etc. The storage of analogue content requires more space than does the digital equivalent and therefore the incurred real estate costs will be higher. The special facilities costs include climate control, special lighting, special handling etc. These costs are higher for analogues such as ancient books, paintings and films than they are for digital versions. Storage of media in either digital or analogue form requires much specialised infrastructure. In the case of analogue storage like libraries, the indexing scheme must be put in place, the content must be tagged and this tag information must be put into a collections database. The content must then be stored in the library by people. A similar process must be executed for the storage of digital media. The difference is that digital media could be placed in a database or stored on a device which could in turn be archived in a traditional library or placed into a mechanised library robot. The point is that this aspect of the storage process is not a trivial exercise and requires skill. The retrieval process can only be as good as the storage

infrastructure allows.

The personnel costs involved can be equally high for both types of library as each requires people with specialised skills. The conversion/preservation costs are not insignificant for either type of content. The preservation of, for example, films is quite costly. The first major film preservation effort took place in 1965 when Metro-Goldwyn-Mayer spent $35 million US to preserve their film libraries. These costs are ongoing as each year new films are made and they must be transferred to more stable storage devices. The conversions costs of transferring from analogue to digital are very significant. For example it currently costs, in Toronto, $75.00 per minute to digitise video or $9000.00 for a two hour movie.

Table 4.3.3: Access

	Local Only	**Remote**	**Unlimited Copies**	**24 hour access**
Analogue	yes	maybe	no	maybe
Digital	no	yes	yes	yes

Table 4.3.3 outlines some of the issues associated with providing access to media. Analogues such as books can only be accessed locally whereas digital versions can be accessed remotely via a variety of networks. Analogues could be accessed remotely if the item is delivered to the remote location, whereas digital versions can be accessed via networks. There exists only a limited number of analogue

copies but many users can access the same digital work if so architected. The digital could be accessed via, for example, the World Wide Web. If facilities are open 24 hours a day, 365 days a year, then many users could access an analogue collection.

The retrieval process can only be as good as the storage infrastructure allows. The retrieval and querying process could borrow from the work of the UC Berkeley's Digital Library Project on ZQL which is a cross between the Structured Query Language Version 3(SQL3) and Z39.50(Wilensky, 1995). SQL is the standard data definition and manipulation language for relational databases. It is being developed by the ANSI/X3H2 committee which proposes standards for computer systems languages(Loomis, 1987). Z39.50 is a network protocol. Unlike HTTP and GOPHER, Z39.50 is session-oriented and stateful(http://www.nlm.nih.gov/publications/staff-publications/rodgers/Z39.50/Z39.50#what). ZQL could then work with the chosen representation to provide the needed information retrieval services.

This section demonstrates how the AMM can be used as a design tool. In this case the AMM was applied to performing a preliminary design of a digital library.

4.4) The AMM and Web Pages

Table 4.4.1: AMM and Web Pages

CONTENT	EXPRESS.	COMMERCE	USAGE	DELIVERY	STORAGE	REPR.
Digital Continuous Imagery	HTML VRML	Media Costs Market Price	Continuous Media Paradigm	Physical Movement of Storage Device	Paper Tape	File
Digital Audio	Java	Copyright Holder	Book Paradigm	Network Delivery	CD-ROM Hard Disk	
Digitised Text			HyperText Paradigm			
Digital Still Images			Direct Manipulation			
Virtual Reality			Still Image Paradigm			

The analysis found in Table 4.4.1 is the AMM view of Web pages viewed via Web Browsers. The content of the Web currently includes text, audio, continuous imagery, still imagery and virtual reality. The expressions used by the Web include the Hypertext Markup Language(HTML) and the Virtual Reality Markup Language (VRML). Currently the content and the expressions are mixed together. The commercial attributes of the Web include the normal commercial costs of media, the copyright holder and the market price. However, much content on the Web is freely available. The Web supports a number of usage paradigms including the continuous media, Hypertext and direct manipulation paradigms where the latter has been introduced by VRML. The delivery of the Web's content is primarily via Internet Protocol(IP) based networks. Since the content can also be stored on CD-ROMS, tape, floppy disks, magneto-optical disks as well as traditional hard disks, it could be delivered on these storage devices. The only constraints are that the target machine must be network enabled with an HTTP server or a Web browser.

Finally the content of the web currently uses a file-based representation; i.e., all content must be in individual files. Access to the content in the files will change in the near future as HTTP v1.1 introduces the concept of byte ranges. This will permit parts of a file to be transmitted as opposed to the whole file.

4.5) The AMM and HyTime

This section will present the AMM analysis of HyTime's Hypermedia documents. In the context of this analysis some background information on HyTime will be provided.

HyTime stands for Hypermedia/Time-Based Structuring Language. It was principally developed by ANSI committee X3V1.8.(Newcomb, Kipp, Newcomb, 1991). It is an international standard adopted and supported by the ISO. HyTime is "a standard neutral mark-up language for representing hypertext, multimedia, hypermedia and time based documents in terms of their logical structure." (Newcomb, 1995). Its foundation is SGML and these two together provide uniform:

- information representation permitting the coexistence of multimedia information in a variety of databases and applications.

- meta-information representations of the data objects including information on how to interpret the objects.

- multimedia information structure that represents the playback and processing requirements of multimedia objects and the execution order of the objects. (Newcomb, 1995b)

HyTime is an enabling technology which provides a consistent way of describing hyperlinks between any media.

Table 4.5.1: AMM and HyTime

CONTENT	EXPRESS.	COMMERCE	USAGE	DELIVERY	STORAGE	REPR.
Digital Continuous Imagery	SGML	Media Costs	Continuous Media Paradigm	Physical Movement of Storage Device	Paper	Document
	Finite Co-ordinate Spaces (FCS)	Market Price			Tape	
Digital Audio		Copyright Holder	Book Paradigm	Network Delivery	CD-ROM	
					Hard Disk	
Digitised Text			HyperText Paradigm			
Digital Still Images			Still Image Paradigm			

The content that can be linked via HyTime includes digital: continuous imagery, still imagery, audio and text. The expression used by HyTime is The Standard Generalised Markup Language(SGML). SGML is used for representing hypermedia documents where the structural elements of the documents are identified by a Document Type Definition(DTD). In addition to this expression Finite Co-ordinate Spaces(FCS) are also used. The FCS allows the scheduling of objects as in HyTime " ...an object is ultimately any sort of information at all." (Newcomb, Kipp, Newcomb, 1991, p. 77). The commercial attributes of HyTime include the normal commercial costs of media; i.e., content, expression(s), commerce, usage, media; i.e., content, expression(s), commerce, usage, delivery, storage and representation, as well market price and copyright holder. Part of the expression costs also include the costs of developing DTDs. HyTime does not " ...provide a model for interaction and, in keeping with the SGML philosophy, excludes presentation format modelling." (Koegel Buford, 1995, p. 278). However as it does bring together text, continuous media and still imagery then the usage paradigms used should reflect the media. Accordingly the potential usage paradigms are: continuous media, book and hypertext. Just as with HTML-based Web content, HyTime content could be via Internet Protocol(IP) based networks. Since HyTime content can be stored on CD-ROMS, tape, floppy disks, magneto-optical disks, hard disks and paper it could also be delivered on these storage devices. Finally since HyTime is a hypermedia document representation the representation used is a document scheme.

4.6) The AMM and OMFI

The goal of the Open Media Framework Interchange(OMFI) is to allow the interchange of digital media between heterogeneous platforms. The development of the OMFI, spearheaded by Avid Technology Inc., is a co-operative effort of many industry and standards partners(Avid Technology, 1993). The OMFI is supported by the Interactive Media Association(IMA) and is now included in the IMA's recommended practice for multimedia data exchange. The format is designed to include all the information needed to transport digital audio, video, images and graphics. It also has rules for combining and presenting the media. The OMFI supports "... a series of classes of applications with increasingly complex requirements for the manipulation of media."(Avid

Technology, 1993, p. 9). These include:

- Digital media creation applications such as graphics drawing packages, animation programs, character generators, media viewers and digitisation applications.

- Media composition applications that use media produced by other applications. The media sources can include complicated sequencing information such as multi-track audio and can include Digital Video Effects(DVEs), " cross fades" and " wipes" . These applications fully utilise the OMFI's composition capabilities; however they may not be able to produce the raw media footage necessary to display an effect.

- Full service media applications which provide the functionality of the previous two application classes while also providing the media production capabilities needed to create and play the finished composition.

Table 4.6.1: AMM and OMFI

CONTENT	EXPRESS.	COMMERCE	USAGE	DELIVERY	STORAGE	REPR.
Digital Continuous Imagery	Digital Video Effects	Media Costs	Continuous Media Paradigm	Physical Movement of Storage Media	Tape	Objects
		Copyright Holder			CD-ROM	
					Hard Disk	
Digital Audio	Audio Effects	Market Price	Book Paradigm	Network Delivery		
Digitised Text			HyperText Paradigm			
Digital Still Images			Still Image Paradigm			

Table 4.6.1 contains the AMM representation of an OMFI composition. The OMFI can support digital continuous imagery, digital audio, digitised text and digital still imagery. If the original source of the content was analogue, this can be tracked within the OMFI. The expressions supportable within the OMFI include DVEs and audio effects. The costs involved are the media costs. The OMFI is a standard for the interchange of data. It has nothing to say about usage. Therefore it can be incorporated with any number of the usage paradigms listed above. Given the focus of the OMFI standard, delivery is very important. Media which can be stored on tape, CD-ROM and hard disks can be delivered via network or by the physical transport of the storage devices. Finally the fundamental representation employed by the OMFI is an object. Objects are brought together to by a composition. A composition ultimately is a combination of media from a number of sources. The composition does not contain the digital media but rather has a pointer to the source media; i.e., content and composition information are stored separately.

4.7) The AMM and MHEG

MHEG is the acronym for the Multimedia and Hypermedia Information Coding Experts Group. MHEG, which is under development by the ISO's Multimedia Hypermedia Experts Group, defines standardised formats for media contents and the structure of complex interactive multimedia presentations. The goal of MHEG is to permit the exchange and playback of multimedia documents and presentations on heterogeneous systems(Meyer-Boudnik, Effelsberg, 1995). MHEG will allow an author to produce her CD-ROMs or interactive books in one universally acceptable format. It will also allow hardware and software suppliers to concentrate on producing the one standard MHEG engine rather than an engine for every presentation standard that is available. MHEG is an ISO standard adhering to the Basic Reference Model for Open System Interconnection(OSI). The "...encoding of MHEG is based on the architecture of the OSI presentation layer."(Meyer-Boudnik, Effelberg, 1995, p. 27). The presentation layer is layer six of the seven layer ISO OSI model. This layer "...performs functions that are requested sufficiently often to warrant finding a general solution for them, rather than letting each user solve the problems."(Tanenbaum, 1981, p. 20).

Table 4.7.1: AMM and MHEG

CONTENT	EXPRESS.	COMMERCE	USAGE	DELIVERY	STORAGE	REPR.
Digital Continuous Imagery	MHEG Types	Media Costs	Continuous Media Paradigm	Physical Movement of Storage Media	Tape	Objects
Digital Audio		Copyright Holder	Book Paradigm	Network Delivery	CD-ROM	
Digitised Text		Market Price	HyperText Paradigm		Hard Disk	
Digital Still Images			Custom Paradigm			
			Still Image Paradigm			

Table 4.7.1 is the AMM applied to MHEG. The content types that MHEG can work with include digital continuous and still images, audio and text. The possible expressions of MHEG are special MHEG types which permit for example the overlaying of titles onto video scenes. The commercial costs are the usual suspects. The usage paradigms of an MHEG production include: continuous imagery, still imagery and books. Since MHEG provides the ability to create user interfaces, it is possible that a custom usage paradigm could be developed. The content could be delivered via network or on CD-ROMs, tapes or hard disks. The representation employed by MHEG is object-orientation. The MHEG standard provides a number of classes including: content, action, link and composition classes. The MHEG class hierarchy while providing abstraction with inheritance rules "...does not give methods defining actions on the objects."(Meyer-Boudnik, Effelsberg, 1995, p. 29). In a fashion similar to the OMFI, MHEG separates content from mark-up information. Content is handled via the content class which allows included data and referenced

data. The former "...runs through the encoder/decoder for transfer, similar to all other pieces of data exchanged between two MHEG entities." (Meyer-Boudnik, Effelsberg, 1995, p. 30). The latter type is accessed via a unique reference encoded in the content object. The referenced data is then retrieved at runtime by the MHEG engine. This referenced data method is another example of the separation of content and mark-up.

4.8) AMM and the Dexter Hypertext Reference Model

The Dexter Hypertext Reference Model(DHRM) is "an attempt to capture, both formally and informally, the important abstractions found in a wide range of existing and future hypertext systems."(Halasz, Schwartz, 1994). The goal of the model is to provide a basis for comparing systems, developing interchange standards and interoperability standards(Halasz, Schwartz, 1994). This model has three layers where: 1)the storage layer describes the nodes and links that form a hypertext; 2) the run-time layer which describes the mechanisms supporting the user's interaction with the hypertext; 3) the within-in component layer deals with the content and structures within hypertext nodes. The Amsterdam Hypermedia Model(AHM) is an extension of the DHRM. It extends the DHRM to support hypermedia by including the concepts of time, high-level presentation attributes and link context. Some of the AHM comes from the CMIF multimedia document model.(Hardman, Bulterman, van Rossum, 1994), (Bulterman, 1993), (Bulterman, van Rossum, van Liere, 1991).

The DHRM is used for understanding and benchmarking Hypertext Implementations. It serves "...as a standard against which to compare and contrast the characteristics and functionality of various hypertext(and nonhypertext) systems."(Halasz, Schwartz, 1994, p. 30). It will also provide a basis for the development of interoperability and interchange among hypertext systems. The DHRM is an ideal. It is held out as a goal to be accomplished. It only tells users where they should be targeting and what they should provide.

Table 4.8.1: AMM and the DHRM

CONTENT	EXPRESS.	COMMERCE	USAGE	DELIVERY	STORAGE	REPR.
Digital Continuous Imagery	System Specific Mark-up	Media Costs	Continuous Media Paradigm	Physical Movement of Storage Device	Tape	Components
Digital Audio		Copyright Holder	HyperText Paradigm	Network Delivery	CD-ROM	
Digitised Text		Market Price	Still Image Paradigm		Hard Disk	
Digital Still Images						

Table 4.8.1 is the AMM view of the DHRM. The content types that can be in a DHRM type system include: text, graphics, audio, and continuous images. The expression in a Dexter system is the system specific mark-up necessary to build and support the link and non-link components. This mark-up can vary from system to system. The commercial costs are the usual which include the costs of creating the link and non-link components. The set of usage paradigms includes: continuous, still and hypertext. The first two are included since a link may be resolved to continuous media from a piece of text. The media of a Dexter type system may be delivered via the physical transport of storage device or via a network. The

possible storage devices include tape, CD-ROM and hard disks. The representation employed in the DHRM is a component/link combination where data containing components are connected by relation links(Halasz, Schwartz, 1994).

4.9) Content-Free Expressions

The WWW and HyTime use mark-up languages. The former uses HTML and the latter uses SGML. PostScript is another popular mark-up language. What these mark-up languages have in common is: the mark-up is embedded in the content; and each is a medium expression. As evidenced by the popularity of HTML and the adoption of SGML by corporations as a document handling system, embedded mark-up and therefore expressions is not a problem. How can the current generation of mark-up languages be modified to reflect the inherent commodity nature of media? The answer is to separate content from mark-up languages. The separation of content and mark-up is found in the OMFI's composition practice of providing only a pointer to source media(Avid, 1993) and in MHEG(Meyer-Boudnik, Effelsberg, 1995).

The application of this OMFI and MHEG practice of separating mark-up from content implements a media commodity scheme as content can be utilised with a number of mark-up languages and expressions. If the content and mark-up are separate then the author can sell her content to people who use different mark-up schemes. For example an author could form a partnership with an HTML/Web Page designer to create a WWW version of the novel the author just wrote. In parallel, a PostScript version could be also be created. In both cases this type of arrangement becomes even more effective in a distributed setting like the WWW. An author can, for example, turn his or her computer into not only the device upon which they write but also the appliance which serves up their content to the world to be used with a variety of mark-up schemes. This type of scheme applies not only to text but to images and sound. The authors can apply several sorts of mark-up to their work given the needs of each situation. They do not have to rely on one kind or variety. For example HTML, SGML, Time Code, MHEG, OMFI or a Dexter compliant hypertext system could be used. Furthermore mark-up experts can sell their mark-up schemes independently of the content. For example if an HTML expert designs a handsome web page then that design, because it is separate from the content, can be sold and reused many times as it is a commodity. Authors may choose a particular mark-up template to initially release their work. Selling mark-up is already happening to some degree in the SGML world as Document Type Definitions(DTD) are designed and sold. The American Department of Defence requires vendors to submit bids electronically using a specially designed DTD. As was previously discussed MHEG and OMFI allow the separation of their object classes and composition schemes from the content. In addition to addressing the content commerce issue this separation also saves storage resources as a single content copy can have multiple users.

From a computation point of view separating textual content from the mark-up would greatly speed up any text search as the text search engine would not have to search through mark-up. Also, separating the mark-up from the textual information can yield interesting results. For example if an HTML document had its HTML links pulled out from the content and the images separated from the links then interesting analyses and searches could be performed on the links themselves. Cluster analyses could be performed on the linkage patterns followed by certain researchers. A researcher could use this information as a knowledge map(Wright, 1993)(Kesik, 1996) which could be used to show paths followed by other researchers. If these paths become well trodden then maybe this is a path worth following. Extending the practice of separating mark-up and content leads to some interesting and original results. This is the subject of the next section.

4.9.1)Content-Free Expressions: The Next Generation

In addition to " mark-up-free content" , another mark-up architecture is distributed mark-up. Under this scenario the content could be on one web server in Europe while the mark-up information could be on another web server in Japan. In this case not only is the content mark-up free but both are distributed. Language could be considered a mark-up of sorts. By distributing the mark-up then each nation could apply its own mark-up language; i.e., its national language to an original work. Also distributing content, mark-up, mark-up position, mark-up type and then applying encryption algorithms to each of these may provide a greater degree of security than simply encrypting a whole composed document. Furthermore if the encryption algorithm exploits the fact that the distributed mark-up is used then part of the decryption algorithm would be deciphering each element in a particular order. For example the mark-up has to be deciphered before the content but after the mark-up position. Each user creating new content or reusing some content or mark-up could create at each site some part of the encryption scheme.

The AMC is itself a mark-up. Currently in the world of video, time code is embedded right on the video tape on the time code track. In this case the time code is embedded with the content. This example serves to highlight another important issue; namely, keeping the mark-up separate from the content allows the usage of appropriate mark-up. For example under certain circumstances it may make more sense to use frame numbers instead of AMC. However if the content is created with the AMC embedded in it, then everyone is forced to use the AMC even though it may not be the appropriate solution in many cases. Furthermore any mark-up takes up not only storage space but also takes up transmission bandwidth. If the mark-up is separate and if a media player can be built to function without mark-up then transmission costs could be reduced. This will become more important as we move more and more into a pay-per-use world where users are charged for bandwidth. For example, suppose a student needs a video that was specially prepared to demonstrate how a particular piece of equipment is built. In this case all that is relevant is that the student receive the video segment. Mark-up in this case, like AMC, could be viewed as excessive. Rather what is needed is a light-weight mark-up capable of providing minimal control capabilities and bandwidth utilisation. As another example, suppose a field engineer is repairing a transmitter in a remote region and she needs to see the repair video for that particular transmitter. Because of the remote location all communications are wireless and therefore relatively expensive. In this case a field engineer support system would have to be designed to minimise bandwidth utilisation. To realise this goal the overhead incurred by any mark-up would have to be minimised.

The concept of mark-up is not limited to SGML, HTML and other text mark-up languages. It is a wide spread concept employed by the current generation of strategic information systems. The term strategic information system is being used to represent those systems that organisations use to manipulate, store, access and create information. Examples include: database management systems, CAD/CAM systems, spreadsheets, Geographic Information Systems(GIS). A feature that these strategic information systems share with traditional mark-up languages like SGML is that the mark-up is necessary for each to function. In the case of databases and spreadsheets the mark-up is used to provide searching, security, calculations and consistency to the contained data and media. However one problem with these strategic information systems as they are currently designed is that once an organisation's data is placed in, for example, a database management system it is a very difficult, long and costly procedure to move it to another system that may provide more functionality, more performance or greater distributed capabilities. This architectural feature of the current generation of database systems and spread sheet software puts user organisations at the mercy of database system vendors. The best example of this is that many 20 year old installations of DB/2 running on equally aged IBM mainframes are still being used because of the expense and complexity involved in moving away from these systems. Given that information is many organisations' most strategic and valuable resource, an alternate information system architecture is needed. I propose that the new architecture utilise the concept of content free mark-up. In this case an organisation's media is laid out on the storage medium of choice in a predefined very simple pattern. This pattern could be designed by an international standards body. The vendors of database systems then have to build their systems so that at least equivalent functionality is provided by having their systems use the agreed upon information pattern as the sole means of contact with the media. Under this scenario users can then choose strategic information systems to suit the task at hand rather than trying to make a database system behave like a GIS system. Users are forced to make a database system into a GIS because they can't get their data out of the database system but they now really need a GIS. This type of architecture for strategic information systems explicitly recognises media as an economic entity that should be under the control of the owner organisation rather than the system software vendor. Traditional strategic information system architectures, as was previously discussed, alter this relationship by placing themselves between the media and their owners. Functionality is provided but at a long term cost composed of: 1) reliance on a sole vendor and the risk inherent in that; 2) inability to easily exploit advances in software technology; 3) inability to exploit new hardware; and 4) people must be trained to operate and maintain archaic equipment.

Therefore the economic imperative built into the AMM is in this case demonstrating how a traditional way of working with media can be changed to accommodate new views on content ownership and mark-up. It also demonstrates how very wide spread technology like database management systems could be re-designed to permit mark-up free databases. This concept is a new area of research for database management systems, geographic information systems, spreadsheet technology and all strategic information systems. I will present in Section 5.4 an example of a mark-up free database developed using neural network technology. The examples presented above are also demonstrating that under an economic scenario where the user has to pay for content and delivery, the type and quantity of mark-up must be flexible. Flexibility cannot be achieved if the content and mark-up are entangled. Finally everything that was said about mark-up could be also applied to any expression as I have defined it.

4.10) The AMM, Objects, Services and Systems

As depicted in Figure 2.3.1(p. 30), the AMM represents a medium as a group of attributes or objects. These objects come together so that media are created. The seven attributes that make up the AMM can also be interpreted as the services provided by the AMM. In this case these services are provided by systems; i.e., content system(s), storage system(s), delivery system(s), expression system(s), commerce system(s), usage system(s) and representation system(s). The plural is indicated for system as there can be, for example, multiple content and storage systems which can be provided via a local node or distributed across many compute nodes. This possibility of a distributed system is the subject of Section 4.13.

4.11) Symmetrical Services

The nature of the services offered could be symmetrical where symmetric is defined to mean that both the consumption and production of the service is possible. For example a content service located in Toronto could offer many rendered 3D models as well as the software tools necessary to create the 3D models. This is then a symmetric content service as it offers content and the tools needed to create the content. Some services are not symmetrical in nature. For example a storage service is not capable of being symmetric; i.e., storage services can only be provided, not created on the fly.

4.12) The AMM, Services and Web Servers

As an example these services are provided to a certain degree by the WWW servers. A Web Server is basically an integrated implementation of some of the AMM services as it provides: content, expression, storage, delivery and representation services. Content and the expression are both mixed together and separate. An example of the former is HTML pages while an example of the latter is provided by the ability of a web server to serve any sort of media including text, still and continuous imagery and audio. The storage service of a Web Server is predominantly provided by hard disks. However this service could be provided by CD-ROM, magneto-optical devices, tape systems, basically anything that can be integrated into the Web Server. For example, I have developed a multi-terabyte capable web server where the content can be stored on tape in an automated tape library or on hard disk or a combination of the two. The delivery of the content by a Web Server requires a network or at least a network-enabled operating system. Finally a Web Server uses a byte stream as the representation of the content. As evidenced by the growth of the WWW this type of representation is obviously adequate. The only services currently not provided by Web Servers are commerce and usage services. The former will require the pricing of media where this pricing could conceivably take place on a real-time basis so that the frequency of usage is reflected in the price. The pricing services could be provided by linking the Web Servers into central media markets which resemble today's foreign exchange and equity markets. The issue of usage is not and should not be addressed by a Web Server. This type of service falls into the domain of Web Browsers and other client side applications. The Web Server generates the byte stream. The client application should be able to use the byte stream in the provided form. The current generation of

Web Browsers successfully utilise the HTML stream to implement a hypertext usage paradigm. Web Browsers also demonstrate the implementation of a distributed usage service; i.e., the usage service is implemented separately from the other services.

As was previously discussed, a Web Server is an example of an integrated implementation of some of the AMM's services. A Web Browser is an example of one of the AMM's services being provided in a distributed fashion. Suppose we wish to access all of the AMM's services in a distributed fashion. The Distributed Abstract Media Model will be the subject of the next section.

4.13) The Distributed Abstract Media Model(DAMM)

The services of the AMM can be distributed. This means that the AMM's services do not have to be provided by one site. Rather the services can exist and be accessed independently across a number of sites where each site would provide the services for which they have expertise. The implication of this is that content services could exist in Montreal, Vancouver, Tokyo and Berlin, the storage services could be in Halifax and New York, and all other services could reside in Moosejaw.

This type of distributed construct can be implemented in both the analogue and digital worlds. It currently exists to a certain degree in the analogue world. For example a book is realised via distributed services. An author creates content. The expression, storage and delivery services are provided by the book's publisher and printer. Commerce services are provided by the book retailer. In the case of a book the usage and representation paradigms are pre-ordained. The making of a motion picture or television program is currently done in a distributed fashion. The original content is created at one site while the expression, usage, storage, representation and commerce services are provided by other sites. For example a film may be shot in New Zealand but the special effects are added by a special effects firm in Toronto or Berlin. This distributed model can also be implemented in the digital world. Using digital computer technologies the AMM can provide distributed services over what I call the mediaBus. The mediaBus serves the role of linking the services. It would be implemented using existing and emerging networking technologies such as: Asynchronous Transfer Mode, wireless communications and CORBA Object Request Brokers. Diagrams of the DAMM are found in Figures 4.13.1 and 4.13.2. Every service provided by the DAMM would be accessible via the mediaBus. The Internet which makes the WWW possible is an example of the mediaBus. In this particular case the Internet is a global mediaBus. This mediaBus currently offers many services including: ftp services, gopher services, rlogin services and web services.

The DAMM supports traditional media such as books, video and film. For example a book could be completely developed on the mediaBus from raw content creation to the actual printing of the book. The printing of the book would be realised by the storage service which would be responsible for running the printing press. For that matter a CD-ROM version of the book content could be produced if a CD-ROM mastering device was available. What is necessary to realise this goal is a piece of integration software that is capable of working with and integrating each one of the services that comprise the DAMM. As was previously discussed the WWW is an example of the integration software. Using the Web model a browser could be built which supports the BTP or the Book Transfer Protocol.

The BTP would be used in conjunction with the other services to present a book to the user via a browser-like interface. By distributing the AMM any medium can be produced to service many market segments ranging from the traditional book and news print to the Web. In doing so the previous reality that once content is in book form it is always in book form is no longer true. Under this scenario content on the mediaBus can become a book, a newspaper, a CD-ROM or a hyper-text. Since content could take any number of forms it could also be delivered via a number of delivery channels. This type of environment permits Just-In-Time-Media(JITM) where a user could get content in the form of their choice. In this type of environment the global content base; i.e., libraries, museums, individual content servers, becomes reusable. Indeed all services on the mediaBus become accessible. Users could access storage services, delivery services, content services, expression services-(For example they could access the creation of digital filters)-if so desired. Under this distributed scenario users who previously could not afford to implement all services of the AMM can now purchase the services via the mediaBus. The mediaBus also allows the addition of many services ranging from encryption systems to search agents which are not part of the AMM.

Figure 4.13.1: DAMM, Version 1

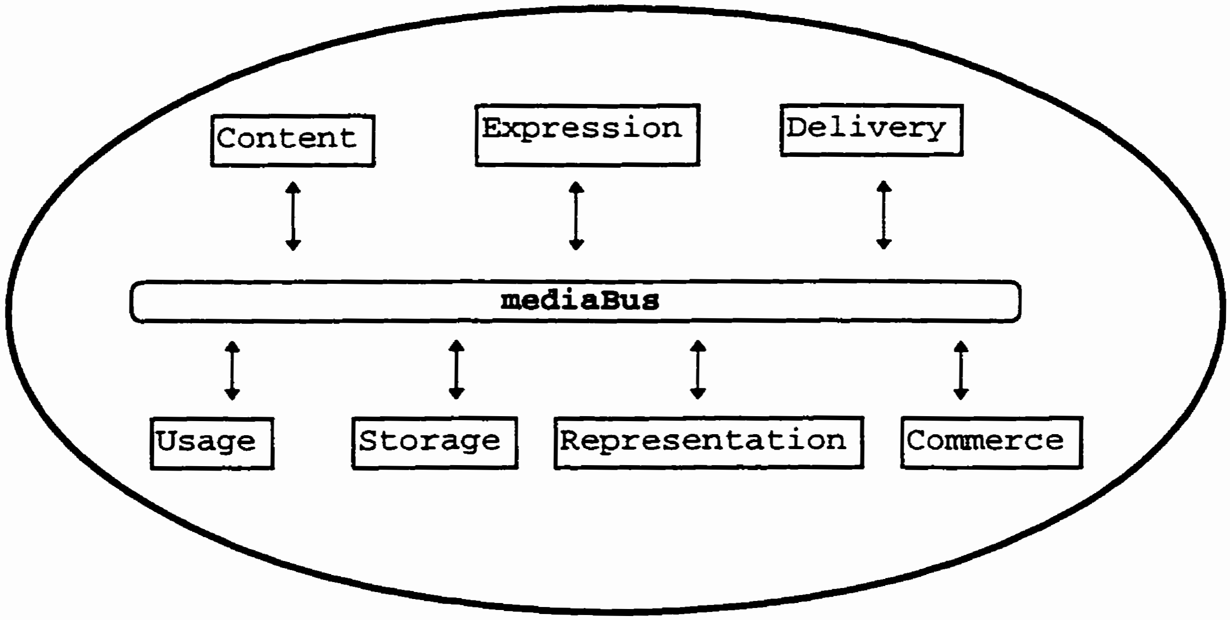
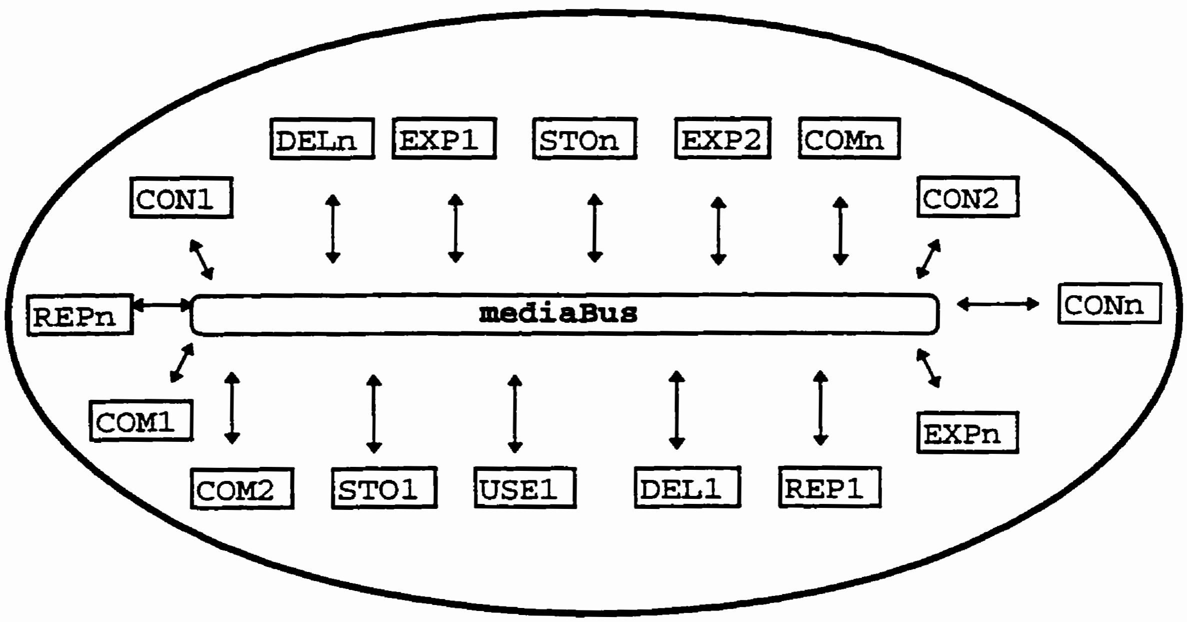


Figure 4.13.2: DAMM, Version 2



Legend: DEL = Delivery, EXP = Expression, USE = Usage, COM = Commerce, REP = Representation, STO = Storage, CON = Content

Figure 4.13.2 demonstrates the DAMM with multiple service nodes. The purpose of Figure 4.13.2 is to show that the same service can be provided by many different sites where these sites can be geographically dispersed but connected to the mediaBus via many different communications technologies ranging from ATM, wireless communications to fax machines. The AMM could be implemented on one machine. The best way to proceed would be to use a technology which supports both the distributed and non-distributed AMM. This will be the purpose of Chapter 5; i.e., using CORBA technology(Siegel, 1996), a skeletal version of the AMM will be implemented. CORBA technology allows applications developed with it to be either local or distributed.

4.14) Evaluation of the AMM

4.14.1) AMM Data Fit

The definition of a model used for this thesis is found in Section 1.2.1. The AMM was defined in Section 1.2.2 as a descriptive model. The goal of this section is to evaluate the Abstract Media model; i.e., how well the AMM fits the data.

The data being used to validate the model is qualitative in nature. The reason for this is that the AMM is not a parametric model which can be statistically estimated. Rather validation is provided by analysing traditional and non-traditional media using the AMM. This exercise validates the model by demonstrating that it can replicate the existing media. This technique is similar to that employed by statistical models; i.e., run the model over the historical data set and see how well it replicates. The AMM "fits " traditional and non-traditional media.

That the AMM's fit cannot be statistically measured is not unique. Many useful models which have been built to understand very complex systems face the same dilemma. For example, the OSI/ISO model was developed to meet the challenge posed by a communications world being populated by heterogeneous protocols and standards. The model was built in order to provide a framework for open systems and co-operation between communications systems. There was no official evaluation of the model. Rather it was the result of 100 experts coming together to devise a model to solve a problem(Zimmerman, 1980). The DHRM was created by a group of HyperText experts getting together in Dexter, New Hampshire. Each of these models was designed to fill a need. Each is a descriptive model whose data fit cannot be statistically or mathematically determined. Each is an artificial construct which provides insight into their particular area. This is what I hope the AMM can engender; i.e., it can be a starting point for a larger effort where the end result is a comprehensive framework that can be used to gain insight into media, create new media types and help manage and work with the vast variety of heterogeneous media standards, technologies and terms. My work is merely a starting point. I hope to identify the initial conditions of the model. I am proposing the AMM as a conceptual starting point.

4.14.2) Conceptual Fit

Gibbs and Tsichritzis(Gibbs, Tsichritzis, 1994) presented a set of requirements that a framework must satisfy. This framework was motivated by two primary concepts:

1) digital media
2) entities that produce, consume and transform digital media such as cameras, memories and displays.

The framework was also developed to support multimedia programming. This differs from my primary concept for the AMM; i.e., driven by the fundamental attributes of media. Furthermore the AMM can support multimedia programming in addition to providing a framework for media design, usage and understanding in both the analogue and digital worlds. However Gibbs places six requirements on his framework. These requirements are:

1. **Economy of concepts**: The framework should be based on a small number of concepts to avoid becoming a maze of media specific details. It is especially important to identify any general concepts that apply across media types.

2. **Open:** The framework should be extendible to incorporate new media types, data representations and hardware

3. **Queryable:** A framework should specify interfaces for querying environments concerning their capabilities.

4. **Distribution:** A multimedia framework should help partition applications that facilitate distribution. In particular, the objects within the framework should correspond to easy-to-distribute units or subsystems. Many future multimedia applications are likely to be distributed, so the utility of a framework is greatly diminished if it conflicts with distribution.

5. **Scaleable:** A multimedia framework should support

scaleable media representations. If media representations are scaleable, applications can increase quality as platform performance increases.

6. **High-level Interface:** A multimedia framework should provide the high-level interfaces for media synchronisation, media composition, device control, database integration and concurrent media processing activities.(Gibbs, Tsichritzis, 1994).

The AMM meets these framework requirements. The AMM is based on a small number of concepts, it is media driven. It is open in that any new media type can be introduced. Indeed I fragmented and decontextualised media into the seven attributes that make-up the AMM in order to facilitate the development of new media types. Furthermore the AMM places no constraints on the data structures or hardware. Whatever is needed can be utilised.

Each attribute of the AMM can be made queryable. Also since the DAMM supports the concept of multiple services, an application can query the capabilities of expression service A and if this does not offer the user what he desires then another expression service can be queried.

The AMM was designed to support distribution(see the DAMM above). The AMM places no limits on quality and capability. Therefore it is scaleable. Finally the AMM offers a high-level interface to support media synchronisation, media composition, device control, database integration and concurrent media processing activities. The synchronisation issue would be handled be the delivery object which would invoke the appropriate engines to guarantee synchronisation. Media composition is the responsibility of the content creation tool which will call the desired media expression. Device control will be provided by the usage object. Database integration will again be the responsibility of the expression class.

Therefore the AMM conceptually fits the framework proposed by Gibbs. Further evidence of the AMM's conceptual fit is provided by Muhlhauser and Gecsei who say:

> *We do not favor a monolithic system or a complicated underlying model but rather a simple yet powerful paradigm, modular yet harmonized components and sophisticated component-based programming.* (Muhlhauser, Gecsei, 1996, p. 60)

The AMM has simple underlying paradigm that supports component-based non-monolithic programming where the components are naturally harmonised as they are derived from and form the basis of media.

4.15) Potential Model Abuse

The AMM, like any model, is a compromise between reality and useability. Useability is achieved if the model provides insights into the complexities of the system being studied where these insights are not obtainable from the direct observation of the system under investigation. However useability is realised by making certain assumptions and idealisations which reduce the model's reality. Striking a balance between reality and useability "...is the essence of good modeling."(Intriligator, p. 14). Appreciating and understanding the inherent compromises of any model is important as it will prevent model misuse and abuse.

The AMM is a descriptive framework for working with and understanding media. The DAMM, on the other hand, is a prescriptive entity that can be implemented. As I stated earlier, the DAMM offers a design pattern for distributed media applications. While this is helpful, it is probably not the only media design pattern or it may be the wrong type of design pattern for certain users or conditions. Any user tasked with developing a distributed media system should spend some time considering the particular problem and should seek out other pertinent design patterns. Some clear thinking and some comparison shopping could help avoid a costly, time-consuming, inappropriate development effort.

The AMC offers a very flexible media representation. This flexibility, while useful, can be problematic. The problem is that the AMC places no constraints on the amount of information it can contain to represent a piece of work. This means that a fully specified AMC for a still image could, potentially, exceed the size of the still image in terms of byte size. If an AMC is done for each frame of a film, the space occupied by the AMC could exceed the storage requirements of the entire film. Striking a balance so that the size of the AMC is limited while providing sufficient descriptive power to facilitate access to material is not easy. However any system that implements the AMC must find this balance.

4.16) Conclusions

This chapter has presented the application, uses, future and fit of the AMM. Via the design examples and showing how the AMM can be adapted to providing distributed media services, I believe I have demonstrated that the AMM can provide a blue-print for building media systems and services. This blue-print, because it is media based and not technology based, will not become obsolete.

Chapter 5: Implementation

The goal of this chapter is to implement a specification of the AMM and an example of one of the AMM's services: the storage service. These will be implemented using the Object Management Group's(OMG): Common Object Request Broker(CORBA), Interface Definition Language(IDL), (Siegel,1996)(Schmidt, 1996) and the C++ computer language(Stroustrup, 1986),(Eckel, 1996). The OMG creates software specifications, not software. The software is developed by companies such as IBM, Hewlett-Packard, Digital Equipment Corporation, IONA Technologies, Chorus Systems. These corporations use the OMG specification to create CORBA -compliant systems. The next section will explain why I chose CORBA, and therefore IDL and C++ technology, as the implementation tools.

5.1) Why use CORBA, IDL and C++?

I chose CORBA technology because I wanted a tool that could be used to:

1. model systems in an object-oriented way,

2. implement both non-distributed and distributed systems.

As indicated in Section 4.10, the AMM can be interpreted as a set of objects that provide the media services necessary to create a medium. Accordingly a tool capable of supporting object-orientation is necessary. The Distributed Abstract Media Model explains the second requirement. The AMM can be treated as non-distributed and the DAMM is the distributed version. CORBA allows me to implement both the AMM and the DAMM because it supports Distributed Object Computing(DOC). The benefits of DOC include:

- Providing the benefits of object-orientation to Remote Procedure Tool-kits.

- Enabling, at higher levels of abstraction, inter-networking between applications.

- Providing a foundation for building higher-level mechanisms that facilitate the collaboration among services in distributed applications(Schmidt, Vinoski, 1995), (Siegel, 1996).

The benefits of using object-oriented technology; i.e., encapsulation, inheritance, dynamic binding, parameterised types and polymorphism, have been well documented and dealt with by(Stroustrup, 1986), (Eckel, 1996), (Gorlen, Orlikow, Plexico, 1990).

Enabling the inter-networking of applications at higher levels of abstraction is beneficial as it allows one to focus on the task, rather than having to worry about the semantics of very low level byte streams like TCP/IP sockets or Named Pipes under Windows NT. Indeed "A primary objective of DOC is to enable developers to program distributed applications using familiar techniques such as method calls on objects."(Schmidt, Vinoski, 1995, p. 2). This type of environment allows one to access the services of a remote object as easily as those of a local object via a simple method call. For example the services of the storage attribute could be accessed from a client application by invoking object methods such as backup() and restore(). This example will be completely developed later in this chapter.

Providing a foundation for building higher-level mechanisms that facilitate the collaboration among services is essential. Part of this foundation is the Object Request Broker(ORB), the central component in CORBA. It provides mechanisms for invoking methods on local/remote objects. These mechanisms also automate "object location, creation, activation and object management"(Schmidt, 1996, p. 16) as well as moving messages and data between objects. The message passing capability of an ORB is immediately useful as the ORB provides the functionality needed to implement the DAMM's mediaBus.

The object creation, activation and management capability can be used to implement and support the location transparency of services. This means that all the services of the AMM could be on one compute node, or several different, geographically dispersed, compute nodes. This capability supports the scenario depicted in Figure 4.13.1(p. 92), as well as allowing the AMM and DAMM to be built one service at a time. All of the AMM's services do not have to be implemented at once. The object-oriented nature of the AMM supports this type of modular construction; services can be added and changed as time and resources permit.

Implementing a system in this modular fashion also allows the management of complexity and technological and financial risk because a media infrastructure can be created object by object. The modularity and location transparency also permit the replacement of each service as better and more efficient services are created.

It could also set up a market for services where the mediaBus supports services of all types and users select the service they want or require. For example there could exist two-hundred and twenty storage services, seventy-four expression services, fifty-five commerce services, twelve delivery services, seven-hundred content services, nine representation services and four usage services, all accessible via the mediaBus.

Having motivated and justified my need for a DOC environment, why was another DOC environment such as OSF's Distributed Computing Environment or Microsoft's DCOM not chosen? I chose CORBA because it provides the functionality I need. It provides "...an infrastructure to integrate application components into a distributed system"(Schmidt, 1996, p. 7) as well as supporting standalone applications. It gives me a "two-for-one " option so that both the AMM and DAMM can be simultaneously implemented. CORBA is a mature technology that is also available on a wide variety of operating systems including most UNIX flavours, Windows NT, OS/2 and Mac OS. Finally, CORBA is based on the specification produced by the Object Management Group(OMG) which is an open, 600-member consortium with membership around the world. I found this solution far more attractive than a proprietary option such as DCOM which has become available only recently under Windows NT 4.0. Finally the OSF's DCE was not used as it is more procedural than object-oriented(Schmidt, 1996). As I indicated above, the AMM is comprised of objects and this therefore renders the OSF DCE inappropriate.

Once I selected the CORBA technology, this decision dictated that the OMG's Interface Definition Language(IDL) be used(Siegel, 1996)(Schmidt, 1996). The OMG IDL is an object-oriented interface definition language which supports both single and multiple inheritance, exception handling and strong variable typing. OMG IDL allows an object architect to specify "... the operations an object is prepared to perform, the input and output parameters they require, and any exceptions that may be generated along the way."(Siegel, 1996, p. 17). It is a formal language, defined in terms of a grammar, whose role is only to define interfaces. In order to programmatically implement and access these OMG IDL interfaces, a programming language must be used. OMG IDL can be mapped to multiple programming languages. Language mappings currently exist for: C++, C, Smalltalk, COBOL and Java. OMG's IDL also ensures operating system independence. Note that other Interface Definition Languages exist. These include the Open System Interconnection's(OSI's) Abstract Syntax Notation One(ASN.1). The ASN.1 is a formal language with a grammar used to describe abstract types and values but not methods(Rose, 1996), (Kaliski, 1991). Snacc is a free ASN.1 to C/C++ compiler available at ftp://ftp.fokus.gmd.de/pub/freeware/snacc. It may be possible to link ASN.1 and OMG IDL via C++. The OSF's DCE also has an IDL. However it is procedural as opposed to object-oriented.

Finally, I chose C++ as the programming language because:

- C++ is an object-oriented language,
- OMG IDL is a superset of a C++ subset.

There is a strong relationship between C++ and OMG IDL. I made a decision to exploit the synchronicity between C++ and IDL. As I have indicated, I needed an object-oriented modelling tool. C++ and OMG IDL provide the requisite object-orientation capabilities. The object-oriented heritage of C++ and IDL owes much to the Simula-67 language(Stroustrup, 1986). Simula-67 was the first Object Oriented Programming(OOP) language created in Scandinavia in 1967. It was created to function as a modelling tool where Simula-67 would allow the creation of a general description of an entity's characteristics and behaviours. In Simula-67 the description of an entity's characteristics and behaviours is called a class and an instantiation of a class is an object(Eckel, 1996). Therefore given C++'s modelling and object-oriented heritage, the choice to use it in order to implement the interfaces specified by the OMG IDL seems natural.

5.2) Applying the AMM

A prototype version of the AMM has been implemented. The AMMda or the AMM demonstration application is a prototype system that demonstrates the use of a prototype set of AMM services. However before discussing in greater detail the prototype system, the methodology used to build it must be presented.

5.2.1) Methodology

The methodology used to build the AMMda is the Object Modeling Technique(OMT) of Rumbaugh *et al*. There are many methods available with the most popular including Booch(Booch, 1991), Responsibility Driven Design(Wirfs-Brock, 1990) and the OMT. The OMT, which incorporates many of the principles found in Booch and RDD, focuses on thinking about and utilising abstractions that exist in the real world in order to develop software(Rumbaugh, 1991). The object-oriented development supported by OMT promotes a way of thinking about complex problems. In this regard the OMT approach is very appropriate for me to use as I have modelled the AMM as a media framework comprised of objects. In the next section, a brief description of the OMT will be found as it has been thoroughly explained elsewhere (Rumbaugh, 1991). I will then proceed to use the OMT to build the AMMda.

5.3) The Object Modelling Technique

The OMT is a way of thinking abstractly about problems by using real-world concepts. It places great emphasis on using object-oriented constructs and concepts to model real world phenomena. Object-orientation, under the OMT, is not solely a technique for programming(Jezequel, 1996). The OMT has the following stages: Analysis, System design, Object design and implementation. In the analysis stage, the analyst builds a model of the problem domain in order to reveal the important properties. The model objects should be " ...application domain concepts and not computer implementation concepts such as data structures."(Rumbaugh, 1991, p. 5). The analysis model is extended in three dimensions: the object model, the dynamic model and the functional model. These three models form a modelling tripod with the "... functional model specifying what happens, the dynamic model specifying when it happens and the object model specifies what it happens to."(Rumbaugh, 1991, p. 123).

The system design stage has decisions being made about the overall architecture, performance characteristics, resource allocations and a strategy for attacking the problem. With the system design in hand, an object design can be performed. The object designer will implement a design model with implementation details using the analysis model and the strategy worked-out during the system design. The focus of the object design is to develop the necessary data structures and algorithms. The implementation stage will see the translation of the object design into a particular programming language.

5.3.1) Implementing the AMMda

The OMT will now be applied to developing the AMMda.

5.3.1.1) Problem Statement

The problem to be addressed is to demonstrate the how the AMM can be used. In order to realise this goal a *prototype* set of AMM services was implemented. These services have limited functionality as this prototype's goal is to provide a demonstration of how the model can be used. A full production version of the AMM will be the subject of future work.

5.3.2) Analysis: Object Model

As dictated by the AMM, the object classes that form the object model are: Content, Expression, Commerce, Usage, Delivery, Storage and Representation. The object model is depicted in Figure 5.3.2.1. The role of this diagram is to show each class, their operations and how they are linked to one another.

5.3.2.1) Identifying Associations and Attributes

- Content, as extensively discussed in Chapter two, can be text, still and continuous imagery, audio and virtual environments.

- Expressions are applied to and part of content. For example HTML can be used to mark-up a piece of text. In this case the expression is applied to and part of the content. An image filter can be used alter a still image in some desired fashion such as smoothing edges. A compression method can be used to represent content differently so that it occupies less storage space.

- The Commerce class will determine the economic value, if any, of media. The commerce object could be applicable to other classes. For example an expression could have economic value. A delivery channel will have certain economic properties.

- The Usage class will determine the employed usage paradigm. This could be determined *a priori* or a dynamic attribute of a sophisticated application. As was noted in Chapter two the selection of the usage paradigm will impose constraints on other attributes and therefore in the context of this discussion, object classes.

- The Delivery class determines how media will be delivered. The delivery class can take several modalities, as described in Chapter two, and can impact the content.

- The Storage class will be able to store content, expressions and other attributes. The device actually used to perform the storage will be determined by economics and performance considerations.

- The Representation class will be used to implement the Abstract Media Code.

5.3.2.2) Identifying Operations

- The Content class will offer the ability to *create* new content and *manipulate* existing content.

- The expression class allows the *application* of its expression(s) to content.

- The Commerce class provides a way to *price* media.

- The Usage class *sets* and *gets* the usage paradigm.

- The Delivery class *sets* and *gets* the delivery paradigm.

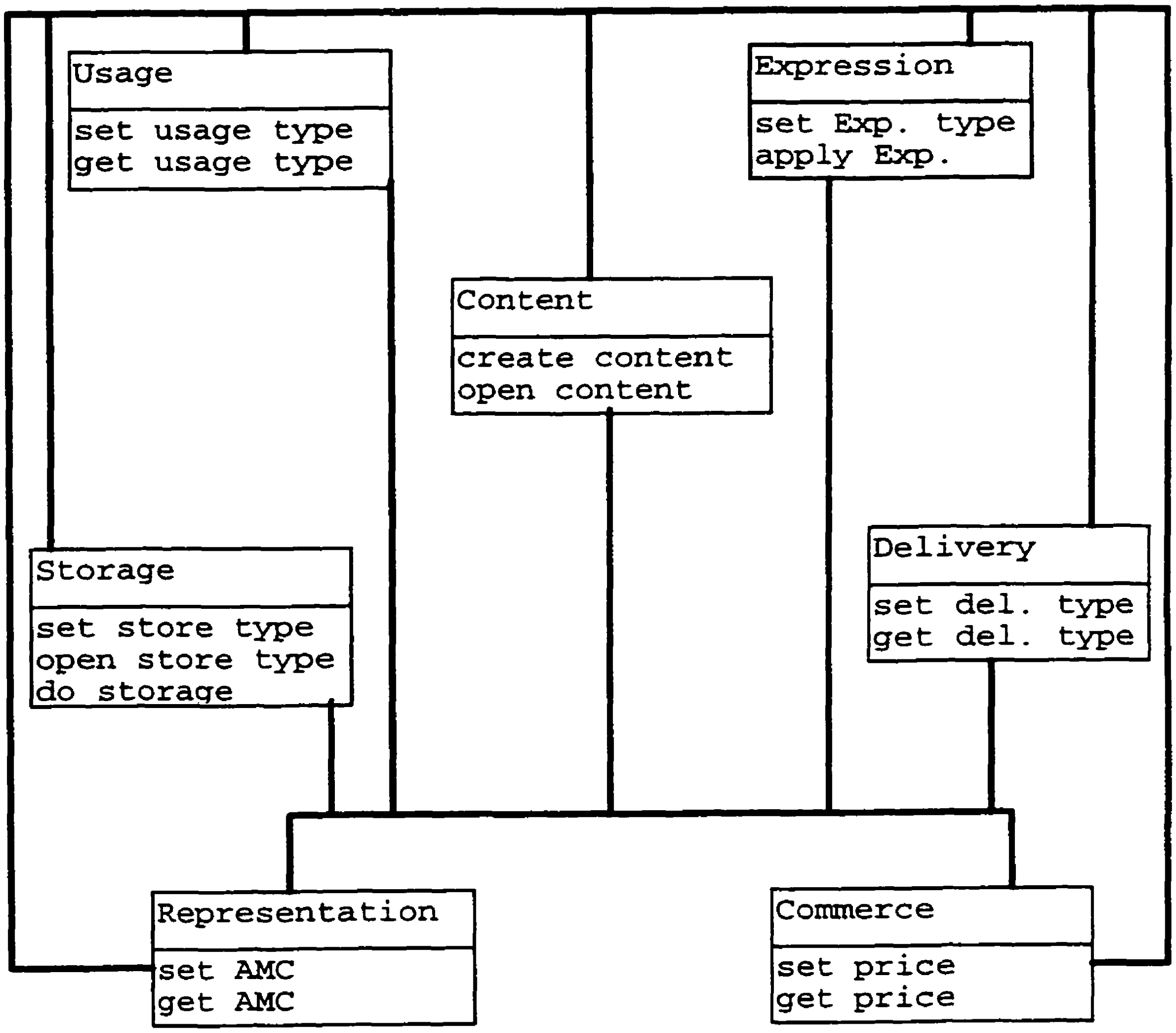
- The Storage class provides the ability to *set* the storage attribute. It must also provide the mechanism for *saving* the data stream to the selected storage medium.

- The Representation Class must allow the *setting* and *getting* of the Abstract Media Code.

5.3.2.3) Identifying Inheritance

I view each of the AMM's services as orthogonal. Therefore no inheritance relationship is necessary.

Figure 5.3.2.1: Object Model



5.3.3) Analysis: Dynamic Model

As previously indicated the dynamic model is concerned with showing the sequence of operations. It is a representation of the operation of the model over time. The dynamic model is comprised of multiple state diagrams. A state diagram is drawn using a rounded box which contains a name to represent a state. A transition from one state to the next is drawn as an arrow from the receiving state to the target state where the label on the arrow is the event that caused the transition.

Figure 5.3.3.1: State Diagrams

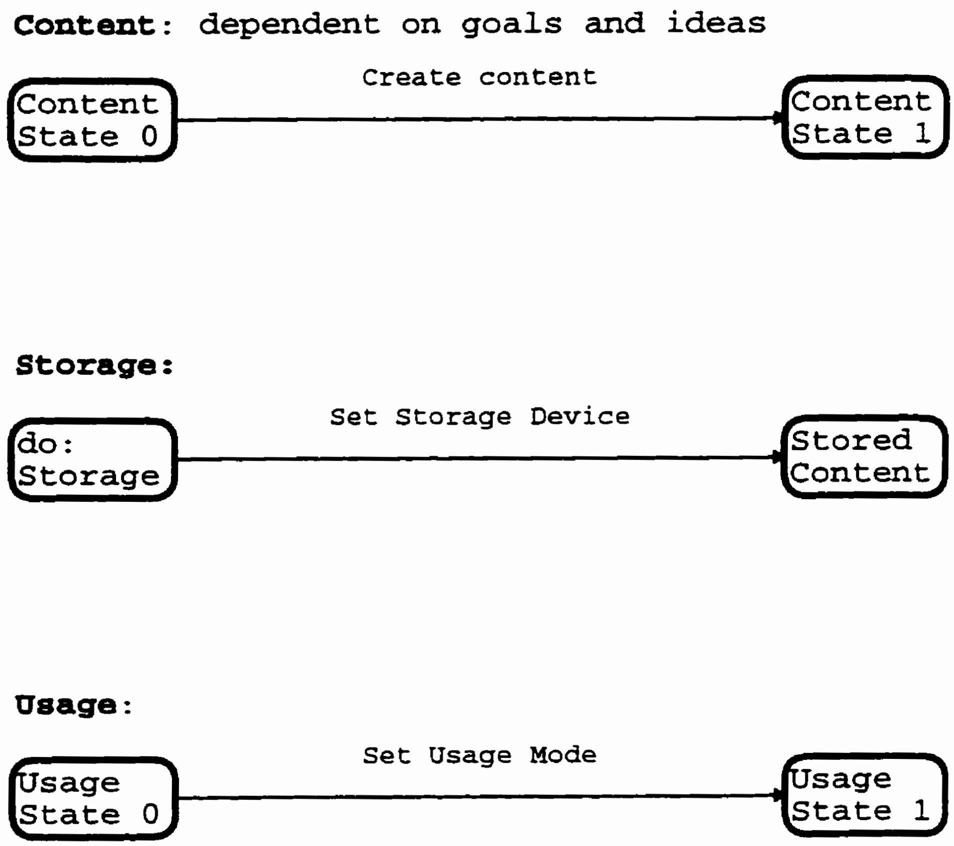
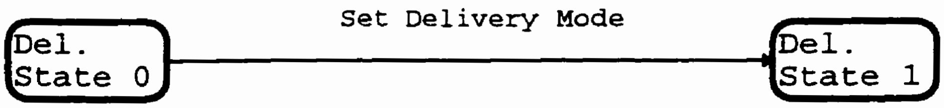
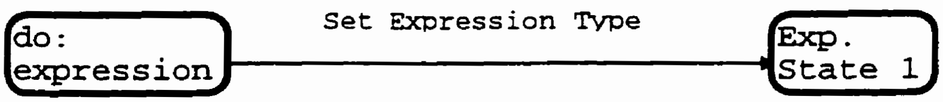


Figure 5.3.3.1: State Diagrams Continued...

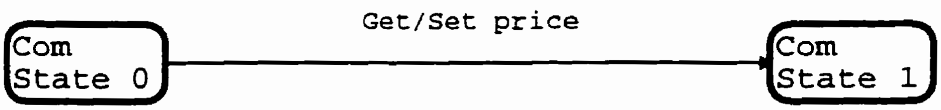
Delivery:



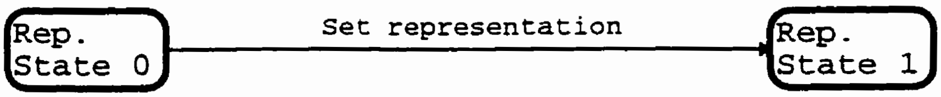
Expressions:



Commerce: dependent on market forces



Representation:



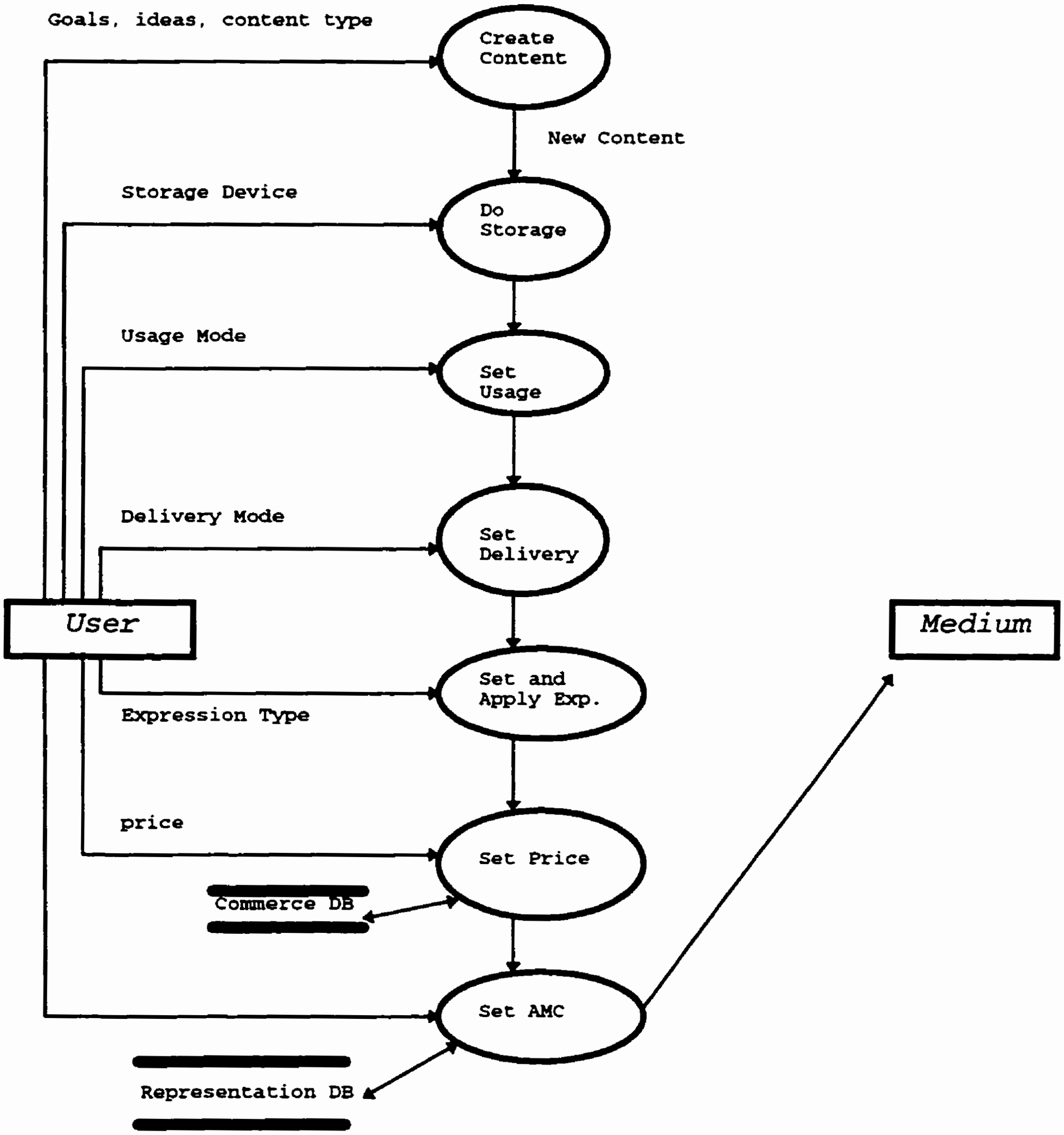
The state diagrams of Figure 5.3.3.1 show the functioning, over time, of the AMMda. The first step is the creation of content. The next is the storage of the newly created content. This is followed by the selection of the usage and delivery mode. The chosen expression is applied and the new medium is priced. Finally the AMC representation is saved to the representation data store.

5.3.4) Analysis: Functional Model

The functional model is the third leg of the modelling triad. Its role is to specify what happens. The functional model consists of multiple data flow diagrams. A data flow diagram(DFD) is a graph showing the flow of data values from their sources in objects through processes that transform them to their destinations in other objects. A DFD contains processes, data flows, actors and data stores. A process is drawn as an ellipse containing a description of the transformation. A data flow is drawn as an arrow between the producer and consumer of the data value. An actor is drawn as a rectangle. It is an active agent that produces and consumes values thereby driving the data flow diagram. The two actors for this functional model are the user or creative agent and the medium. A data store is drawn as a pair of heavy parallel lines containing the name of the store. It is a passive object, storing data for later access. The functional model is found in Figure 5.3.4.1 below.

The functional model depicts the situation where the user has goals, ideas and a content type in mind. The user wants to convert this into a medium. For this to happen the new content must be created and stored. A usage and delivery mode must be set. The chosen expression must be applied and an economic value must be assigned to the newly minted medium. Finally an AMC representation will be saved.

Figure 5.3.4.1: Functional Model



5.3.5) System Design

In the OMT, the system design phase provides a high-level strategy for solving the problem and building a solution. I will now perform the system design which requires decisions about organising the system into subsystems, allocating subsystems to hardware and software components and making conceptual and policy decisions for the design framework.

5.3.5.1) Subsystems

The AMM is organised as a sequence of partitions(Rumbaugh, 1991) where each partition represents an independent or weakly coupled subsystem. Each of the subsystems provides a service. In this case the services to be provided are: Content, Expression, Commerce, Usage, Delivery, Storage and Representation.

5.3.5.2) Concurrency and Control

No objects have to be active concurrently. Complete sequentiality is possible thereby allowing the use of a single thread of control.

5.3.5.3) Allocation of Subsystems

The single thread of control can be distributed. This will be implemented via the distributed object technology which will allow many distributed general purpose processors to provide the required services. This scenario is the scenario presented by the DAMM. The single control thread can also be implemented on a single general purpose compute node so that a distributed paradigm is not mandatory.

5.3.5.4) Management of Data Stores

The data stores will, for this prototype version of the AMMda, be implemented via files.

5.3.5.5) Global Resources

The AMMda is a prototype system running on a PC. For a non-prototype production system, many resources would have to be managed. These include: robotic tape libraries and tape drives, printing presses, delivery vehicles, storage warehouses, archive and retrieval operators, telecommunication facilities such as satellites and shared data repositories such as database systems to name a few. For those resources which are liable to conflicting access, a locking mechanism will have to be introduced. This locking mechanism will provide the lock holder the right to access the resource.

5.3.5.6) Control Implementation

The AMMda is an event driven system(Rumbaugh, 1991) where different objects are invoked by a creative agent.

5.3.5.7) Boundary Conditions

When the AMM services are started, internal variables particular to the service will be initialised. When a user creates new content or opens existing media other initialisation particular to those operations occur. The creation of new text content causes a connection to the media bus to be established. Opening existing media causes the price of the media to be returned.

5.3.6) Object Design

The object design sees a shift in focus from the "...external requirements to the computer implementation."(Rumbaugh, 1991, p. 426). This object design will, as is frequently the case, be a simple, direct implementation of the analysis model.

5.3.6.1) Content

In general, the content object should provide access to classes capable of working with text, continuous or still imagery, audio or virtual environments. There could be multiple video classes where each is a service offered by different vendors. For example there would be an AVID video class, a MATROX video class where each object would be loaded on a per-use basis. For the AMMda prototype, the content object only provides a very simple text processing and creation object.

5.3.6.2) Expression

An expression is applied to a content stream. For example if a series of TARGA images are sent to an MPEG compression engine then an MPEG stream will be produced. If a stream of ASCII text is placed into an HTML style sheet then a web page is built. Therefore the process provides a set-expression and apply-expression capability. For the prototype system only the ASCII-to-HTML expression was implemented.

5.3.6.3) Commerce

The commerce service is used to set the price of media and to get the price of media. Therefore this process will see an initial price being set via a set-price operation. The price can be retrieved via a get-price operation.

5.3.6.4) Usage

The usage class allows the usage paradigm to be set to, for example, book paradigm, hypertext paradigm, or continuous media paradigm. Therefore there will be a set-usage paradigm operation. There will also be a get-usage paradigm operation.

5.3.6.5) Delivery

The delivery class provides the facility for setting the delivery channel. There is, therefore, a set-delivery-mode operation and a get-delivery-mode operation.

5.3.6.6) Storage

The storage class provides the facility for setting the storage attribute. This requires the existence of a set storage mode operation. There is also a store data stream operation which allows the data stream to be stored to the chosen storage device.

5.3.6.7) Representation

The representation class implements the Abstract Media Code as I have defined it. It provides two operations: set-the-AMC and get-the-AMC.

5.3.7) Implementation

The AMMda is implemented using IDL and C++. The motivation for using IDL and C++ was provided at the beginning of this chapter. The AMMda client application, developed using Microsoft Windows NT, demonstrates how a client application, could use the services of the AMM. The AMM services are delivered via a shared CORBA server. This means that one process provides the AMM services. In the explanation that follows I will be referring to certain member functions with only a sub-set of all the parameters needed by the function. I am doing this to keep the discussion focused and clear. For the reader interested in the complete parameter specification of a member function, the complete IDL code for the AMM classes, class member functions and their parameters can be found in Appendix A.

To demonstrate a typical session with the AMMda consider the following situation where a fictional author called Bob Jones wants to write an article on activity in the real estate market for a business publication. Bob is going to create a new article comprised solely of text called realEstateActivity via the text creation tool available to the ammDa. This tool is activated, ultimately, via the following call:

```
ammContent->textToolDistributed( byte stream )
```

where 'byte stream' is the byte stream being processed by the tool. Note that prior to this call a number of housekeeping functions needed to move the byte stream along the mediaBus were performed.

After Bob has finished writing he wants to store his newly created text content on disk. In order to store Bob's work on disk the following call to the storage service will be performed:

```
ammStorage->doStorage( byte stream )
```

where 'byte stream' is the byte stream being stored. Again this call was preceded by housekeeping functions needed to place the content on the storage device of choice via the mediaBus.

Now assume Bob wants the piece to be accessible via the WWW. He knows that HTML is the mark-up language of choice for documents delivered via the WWW. Therefore he converts his document by applying the HTML expression. This is done by the following series of calls to the expression service:

```
ammExpression->setExpression( "HTML " );
ammExpression->applyExpression( "realEstateActivity ");
```

The result of this last function call is to create a file called realEstateActivity.html.

Bob then sets the delivery mode to the WWW via the delivery service:

```
ammDelivery->setDeliveryMode(WWW);
```

Bob then wants to attach a price to his work. Given his knowledge of the publication market, he wants $C0.50 every time his realEstateActivity.html piece is accessed. In order to set the price the following member function of the commerce service is called:

```
ammCommerce->setCommerce( $0.50 );
```

Finally Bob will set the representation on his piece of work. He will do this for two reasons: 1) using AMC provides a rich set of searchable keys and 2) other media creators like using this industry standard, time-code-based representation to incorporate other peoples' work into their own. Note that the Abstract Media Code was being built up as Bob performed the previous set of actions. It is set via the AMC service with the following call:

```
ammAmc->setAMC(Abstract Media Code)
```

In my prototype version, this call places the AMC parameters into a flat file called amc.dat. This file is basically a content database. If the content was placed in a relational database management system(RDBMS) users could then access the database to search for media which would be of interest to them using the well known, very expensive tools that accompany any commercial RDBMS. I have developed an alternative database engine utilising neural network technology, a sub-set of the Abstract Media Code and the concept of mark-up free content. This database system is the subject of the next section.

5.4) Neural Database System

The initial and only requirement of this version of the neural database system(NDBS) is that it be query-able after it has been trained on data records where the data records are made up of a word and numbers; i.e., a sub-set of the AMC. By query-able I mean that a user can submit a target record to the NDBS which will in turn identify the records from the training set that are similar to the target record.

5.4.1) Neural Database Systems: The Motivation

Why even consider a NDBS when very capable, stable and reliable technologies such as Relational Database Management Systems(RDBMS) and Object-Oriented Database Management Systems(ODBMS) with powerful query capabilities currently exist? To answer this question let us briefly consider the benefits and costs of using Database Management Systems(DBMS). A DBMS can(Loomis, 1987):

- store, organize and represent an organizations data resources for selective, efficient access.
- provide a secure, consistent and reliable repository for data resources.
- allow data sharing whereby users have concurrent access to data resources.
- facilitate the management of changing information

requirements.

These benefits, which should serve as goals for any database technology regardless if it is a RDBMS, an ODBMS or a NDBS, are not free. In order to provide these benefits costs must be incurred which include:

- *Costly DBMS technology:* DBMS software and the hardware to run it on must be purchased. There are also training and operation costs.

- *Planning and conversion costs:* Planning costs include building a data model while conversion costs involve taking data from either conventional file systems or other DBMS products and placing them into the new DBMS.

One of the initial motivations for developing the NDBS is found in the conversion costs where data must be fed into or placed in the DBMS from a traditional file system or another DBMS. This data insertion process assumes that a data model was developed and implemented in the new DBMS. Also once data is placed in a DBMS, it becomes only accessible via vendor-specific DBMS tools or specially built application programs. The act of placing the data in the DBMS results in much information being added to the data in order to support the DBMS' promised functionality. I view this insertion process as marking-up the data just as the HTML marks-up text, images and sound for dissemination on the WWW. This mark-up is necessary for the DBMS to function whether it is a relational or object-oriented database management system. This is also a characteristic of many of the current generation of strategic information systems. This problem has been previously discussed in Section 4.9, "Content-Free Expressions" so I will not repeat myself. I view marking-up data as a strategic mistake for organizations. Therefore I suggest that a database system which does not require the data to be marked-up in any vendor or system specific fashion as a positive development. Attempting to determine if a rudimentary database management system could be built that does not require either:

- the insertion of strategic data into vendor-specific DBMS products
- the marking-up of strategic data

is one motivation for the NDBS concept.

The second motivation was the development of a tool that could cluster similar media together where the clustering is based on a sub-set of the AMC. In particular I use the who and when attributes of the ETC. Once I have the clusters created they are used to allow queries like: "I have a data record with the following attributes, "weyth 1996 11 31 23 01 01 29" what do you have that is similar? ". The NDBS would then take in the query and respond by identifying which cluster the query best matches. These two motivations led me to develop the NDBS which I will present in Section 5.4.3. As I have indicated I am not using the complete AMC representation. However this work lays the foundation for the complete AMC to be used. Before presenting the system, I will discuss similar work in the field of Neural Database Systems.

5.4.2) Similar Work

The only other effort in the area of what could be called neural databases is the work by Caudell, Smith, Escobedo and

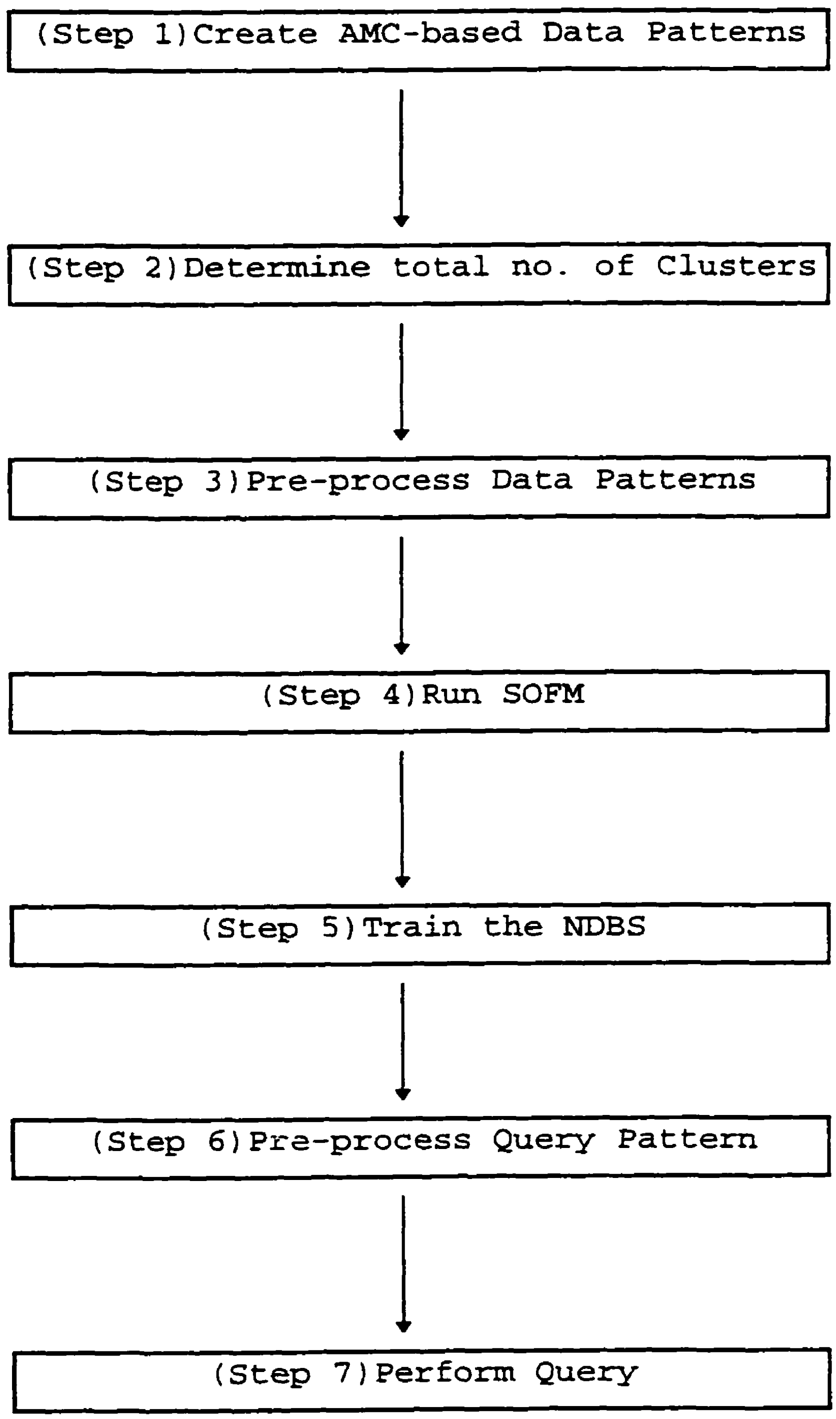
Anderson entitled "NIRS: Large Scale ART-1 Neural Architectures for Engineering Design Retrieval". This work involves using the Adaptive Resonance Theory of Carpenter and Grossberg to create a neural database system for engineering design. The ART-1 network is trained to cluster engineering designs into families. When queried with a new part design, similar designs will be returned. The number of clusters is not preset but is determined by the input patterns and parameters specific to the ART-1 network.

5.4.3)The Neural Database System

5.4.3.1)General Architecture

I have implemented a NDBS by combining a Self-Organizing Feature Map(SOFM) and the Linear Vector Quantization(LVQ) technique. Although the SOFM algorithm was initially developed for the "...visualization of non-linear relations of multidimensional data."(Kohonen, p. 191), the role of the SOFM in the system is to assign the input patterns to classes or clusters. The benefit of doing this is that I do not have to go through the input data and assign each input pattern to a class. Rather this assignment is performed by the SOFM. Once the input data have been assigned to clusters, I then feed this input data to the LVQ system whose only purpose "...is to define class region s in the input data space."(Kohonen, p. 175). Given these class regions, I can then query against them; i.e., submit a query record to determine which class the query record belongs to.

Figure 5.4.3.1: Block Diagram and Algorithm for The Neural Database System



5.4.4) System Specifics and Implementation Details

The SOFM is implemented in C++ using the code base found in "Pattern Recognition with Neural Networks in C++ " by Pandya and Macy. This SOFM can handle both single and two dimensional Kohonen layers. When I run the SOFM I always use a one dimensional Kohonen layer. The reason I do this is that I want the input data to be mapped to clear, distinct clusters. If I use a two dimensional Kohonen layer, many clusters result where a cluster may have one member. Therefore for the purpose of this thesis I will use a one dimensional Kohonen layer where the dimension of this layer is determined by the number of clusters in the data. The total number of clusters is determined by inspecting the data beforehand. This pre-determined number of clusters is why I am interested in using ART2 or Fuzzy ART as this type of technique determines the number of clusters and the clusters themselves. After the SOFM has been trained, I then determine which cluster each input pattern belongs to via a Euclidean distance measure calculated using the weights produced by the training phase and the input pattern values. Once this process is completed I then save the input patterns and their assigned clusters to a file. This file is then used by the LVQ system.

Before I run the SOFM software, a very important data pre-processing stage occurs. As Kohonen says "Natural signals, which are often dynamic, and the elements of which are mutually dependent, must therefore be preprocessed before their presentation to ANN algorithm."(Kohonen, p. 192). Accordingly I pre-process the data. The goal of the pre-processor is to orthogonalize the input patterns as much as possible. Therefore the pre-processor first translates from symbol to number space. It then moves the input patterns away from each other. The method I chose for pre-processing the data is ad-hoc. It is not perfect and can lead to data patterns not being separated into distinct clusters. A fix for this is to increase the number of data patterns for the cluster that we want to exist. Experimentation shows that this will in most cases result in a distinct cluster. See Kohonen, 1995, p. 112 for a further discussion of this.

The LVQ system is also based on the code base found in "Pattern Recognition with Neural Networks in C++ " by Pandya and Macy. This system is applied in a straight-forward fashion to the data produced by the pre-processor and the SOFM. It will, given a data record, identify the class to which the record is most similar. It is in this sense that the LVQ is a database engine because it responds to queries. Demonstrating how this whole system functions is the role of the next section.

5.4.5) Sample Sessions

This section will show the whole system running; that is, it will demonstrate how to use the Symbol-To-Number(STN) or pre-processor program, SOFM and LVQ programs to create a neural database system. The first tests will demonstrate how the NDBS functions and will verify that it is functioning correctly. The next set of tests will in turn demonstrate how the NDBS can deal with noisy data while the third set of tests will show how the knowledge base can be re-trained to accommodate the current state of the user's knowledge base.

5.4.5.1) Basic System Operation

The first step that must be performed is the data pre-processing stage. This is performed by the STN program. It processes a file that contains a five letter word and eight numbers. The five letter word is the media creator's name. Seven of the eight numbers represent the year, month, day, hour, minute, second and frame. The eighth number is just a place holder for identifying the cluster to which the data record belongs. The data file used for the following example is found in Table 5.4.5.1.1.

Table 5.4.5.1.1: First Training Set

```
20 12 5
weyth 1996 12 30 10 12 59 3 0
weyth 1995 12 30 12 59 01 30 0
weyth 1995  2 30 11 31 11 22 0
weyth 1996 11  3 12  1  1  1 0
vault 1996 1  11 11 10 10 10 0
vault 1995  1  1 11 33 33 28 0
vault 1996  1  7  3  4 41 21 0
vault 1986 11 10 11  5 32  3 0
didan 1979 10 22  4  5 57 30 0
didan 1989 10 22 23 15 17 10 0
didan 1989 11 21  4  5 57 30 0
didan 1990 11 12 14 15 57 03 0
loren 1979  5  1  2 16 58 21 0
loren 1997  1 22 12  6 58 21 0
loren 1987  1  1  2 16 58 21 0
loren 1981 11 11 22  6 58 21 0
freud 1996 12 11 11 10 10 10 0
freud 1988 10 11 11 33 33 28 0
freud 1986  3  9  3 41 41 21 0
freud 1991 11 11 11 55 32  3 0
```

The very first row of the data file found in Table 5.4.5.1.1; i.e., 20 12 5 is interpreted as follows: 20 is number of input patterns, 12 is the dimension of each input pattern and 5 is the pre-determined number of clusters. Suppose the data in Table 5.4.5.1.1 is in a file called data3.1. The file data3.1 is processed by the STN program which outputs the results to standard out. In order to capture these results use the redirection operator and provide a filename which will contain the results. Therefore a sample session would be executed by typing the following at the command line:

STN data3.1 0 1 0 > out

The first parameter to the STN program is the name of the file to be pre-processed. The second parameter indicates which letter of the word in file data3.1 forms the basis of the preprocessing. This parameter's role will be made clear when the net is being re-trained to accommodate the user's knowledge base. The third parameter indicates if we will be using the SOFM as the cluster identifier. The fourth parameter indicates if this run of the STN program should be performed on all the input patterns of the input file or should only the first data pattern be used. This type of functionality was put in place because when a query is to be performed the record we want to query with must also be pre-processed with the STN program. After all this, the output is found in the file called out. It is the contents of this file that are then fed to the SOFM program.

The SOFM program takes six parameters. The first parameter is the name of the file to be processed which in this case is the file out which captured the results of the STN program. The next parameter is the number of epochs during which training will occur. The third parameter sets the dimension of the Kohonen layer which is also the number of clusters. The next parameter sets the second dimension of the Kohonen layer. The fifth input parameter is the learning rate while the sixth is the radius for weight updating. Therefore using the file out, with training happening over two hundred epochs for a Kohonen layer which has five columns and one row, a learning rate of 0.1 and a radius of one, the SOFM program would be executed as follows:

SOFM out 200 5 1 0.1 1

This step's results are output to a file called dac2.dat. The first row of dac2.dat contains the number of different patterns, the length of each vector and the number of clusters. Each row after the first contains data identical to the content of the out file except that whereas out had twelve columns, dac2.dat has thirteen. The thirteenth contains the cluster to which the input pattern belongs. Note that the assignment of each input pattern to the clusters seems to be invariant to the learning rate. I ran the SOFM program ten times with the learning rate set to 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99 and the assignment of the patterns to the input clusters remained the same. However this requires a good pre-processor.

After the data have been clustered, the next step is to train the LVQ system. The program that does this is called trainNDBS. It takes three parameters. The first is the training pattern for the LVQ. The second parameter is the number of epochs during which learning takes place while the third parameter is the learning rate. Using dac2.dat, which is generated by the running the SOFM program with the following parameters, out 200 5 1 0.1 1, we run the following command:

trainNDBS dac2.dat 50 0.1

This command will place the weights calculated in the training phase into a file called wf.dat. This file is then read by the queryNDBS program which is used to query the neural net created by the trainNDBS program. The queryNDBS program takes one parameter, the query pattern that is to be classified where this query pattern is pre-processed by the

STN program. To demonstrate that the NDBS functions properly where properly in this case means that the database can correctly identify a query pattern that it was trained on, take the second record from Table 5.4.5.1.1 and place it into a file called dac4; i.e., place the pattern into the dac4 file:

weyth 1996 12 30 10 12 59 3 0

The dac4 file must then be pre-processed by the STN program. This is done via the following command:

stn dac4 0 1 1 > xxx

Running this command pre-processes the contents of dac4 and places the result into the file xxx. This file contains the pre-processed version of the query pattern. This query pattern is then submitted for classification to the NDBS with the command:

queryNDBS xxx

The query process identifies the query pattern contained in xxx as belonging to cluster number 4 which is the correct classification. This can be verified by comparing the contents of xxx with the file dac2.dat. By repeating this process four more times by replacing the contents of dac4 with the following query patterns:

vault	**1996**	**1**	**11**	**11**	**10**	**10**	**10**	**0**
didan	**1979**	**10**	**22**	**4**	**5**	**57**	**30**	**0**
loren	**1979**	**5**	**1**	**2**	**16**	**58**	**21**	**0**
freud	**1996**	**12**	**11**	**11**	**10**	**10**	**10**	**0**

Each will be correctly classified; i.e., the query pattern beginning with vault belongs to cluster 0, the pattern beginning with didan belongs to cluster 2, the pattern beginning with loren belongs to cluster 3 and the pattern beginning with freud belongs to cluster 1. This process demonstrates that the NDBS can correctly classify those patterns it was trained on.

The next test to be performed will demonstrate that the NDBS can correctly identify patterns with noise in them. This is demonstrated by seeing if the following query pattern is correctly classified:

weith 1995 12 30 10 12 59 3 0

This pattern differs from the original in that the name weyth has been misspelled and 1996 has been changed to 1995. This pattern after the usual pre-processing is identified as belonging to cluster 4. This makes sense as weith is very close to weyth. If the query pattern is set to: **xeith 1995 12 30 10 12 59 3 0,** it is also identified as belonging to cluster 4. Again this is plausible as xeith is closest, in an alphabetic and Euclidean sense, to weyth. Therefore these test cases demonstrate that the NDBS is capable of dealing with noise.

The next functionality I would like to demonstrate is that the NDBS can be trained to accommodate the current state of the user's knowledge base. For example, suppose a user only knows that the five letter name they are interested in ends in 'an'. The name could be zeean or meean or piian or xiian, they are not sure. Therefore given a data set, the neural net should be trained to exploit the user's knowledge base which in this case is that the five letter word in the training patterns ends in a 'an'. This means that the pre-processor program, STN, which has a parameter that allows the user to set the starting target in the name, should set that parameter to 3 since the five letters in a five letter word are one of 0,1,2,3,4. Therefore suppose we are working with the data found in Table 5.4.5.1.2 below.

Table 5.4.5.1.2: Second Training Set

```
20 12 5
smith 1996 12 30 10 12 59 3 0
smith 1995 12 30 12 59 01 30 0
smith 1995  2 30 11 31 11 22 0
smith 1996 11  3 12  1  1  1 0
abbey 1996 1  11 11 10 10 10 0
abbey 1995  1  1 11 33 33 28 0
abbey 1996  1  7  3  4 41 21 0
abbey 1986 11 10 11  5 32  3 0
xiian 1979 10 22  4  5 57 30 0
xiian 1989 10 22 23 15 17 10 0
xiian 1989 11 21  4  5 57 30 0
xiian 1990 11 12 14 15 57 03 0
goren 1979  5  1  2 16 58 21 0
goren 1997  1 22 12  6 58 21 0
goren 1987  1  1  2 16 58 21 0
goren 1981 11 11 22  6 58 21 0
cabot 1996 12 11 11 10 10 10 0
cabot 1988 10 11 11 33 33 28 0
cabot 1986  3  9  3 41 41 21 0
cabot 1991 11 11 11 55 32  3 0
```

The parameter set needed for the STN program in order exploit the knowledge base is:

stn data3 3 1 0 > out

After the pre-processing stage, the clusters must be identified using the SOFM program and the neural net must be trained via the trainNDBS program. This is accomplished via the following two commands:

SOFM out 200 5 1 0.1 1
trainNDBS dac2.dat 50 0.1

Now that the net is trained, it can be queried. Suppose the person thinks the name they are looking for sounds like zeean. Therefore the query pattern, which is again placed in a file called dac4, would be:

zeean 1991 10 22 4 5 57 30 0

This file must be pre-processed by the STN program. The command for doing this is:

stn dac4 3 1 1 > xxx

The next step is then to query the NDBS with the query pattern, xxx. The result of this is that xxx is identified as belonging to cluster 2. This cluster contains the training patterns which start with the word 'xiian' which is the name the user was looking for.

Therefore examples presented in this section demonstrate the functioning of a NDBS. The NDBS innovations are:

- Using a SOFM to assign clusters to data patterns.
- Using the LVQ algorithm as the basis of a NDBS.

5.4.6) Benefits of a NDBS

Section 5.4.1 presented the motivations for using a NDBS. It only hinted at the benefits of using a NDBS. Now that we have had a chance to see a NDBS in action, I would like to use this section to clearly specify what I consider the benefits.

1. *Re-trainable:* given the appropriate pre-processing tools, which is not a trivial matter, a NDBS can be re-trained to accommodate the user's knowledge base.

2. *Mobile:* given that the database is contained in a very small parameter file, very small when compared to the disk space occupied by a traditional database management system, it can be moved around very easily. This means it can be transmitted very quickly over modest communications links or placed in, for example, the ROM of a smart card.

3. *Economical Data Representation:* a NDBS is an economical way of representing data and working with that data. Just as a wire frame can be used in CAD systems, a NDBS could be viewed as the wire frame equivalent in the database world.

4. *Modest Computer Resources Required:* a NDBS can be trained on a very modest computer. This work was done on a Pentium 100. Furthermore a traditional file system can be used for a NDBS. This means that unlike traditional database management systems which require special file systems totally under the database management system's control, a regular file system can be used for the NDBS and other applications. This makes for a more cost

effective database system.

5. *Cheaper Database System:* a traditional database management system which entails an expensive data modeling stage prior to the insertion of the user's data into the database system. A NDBS has no such costs. It also leaves the user's data free of mark-up. Therefore using a NDBS avoids the data modeling and insertion costs making the NDBS cheaper to set-up.

6. *Utilises the AMC:* a unique media representation can be used to not only create media but access it from the NDBS. This will greatly reduce training costs.

5.4.7) Costs of a NDBS

Section 5.4.6 presented the benefits that I can foresee using a NDBS. However there are costs. The first is that an adequate and robust pre-processing mechanism must be developed. The second is that users must be trained how to use this type of database technology. The third, and not un-important, is that data placed in expensive traditional database systems must be extracted and made available to a NDBS. This operation would be expensive. Furthermore all that high-priced database management system software would have to abandoned. Many organizations would be very reluctant to do this. The next section will present how this work can be extended.

5.4.8) Future NDBS Work

This prototype effort should be extended in five directions.

The first will be to determine the scalability of the NDBS; i.e., can it handle 50000 training patterns? The second, not un-related to the first, is the development of a general pre-processor so that training and query patterns based on a full AMC specification can be used.

The third direction should be to implement the Hypermap-Type LVQ as it has been demonstrated to be a more accurate classifier than LVQ1(Kohonen, 1995, p. 188). The central principle of the Hypermap, which could be applied to both LVQ and SOM, is the selection of candidates for the "winner " in sequential steps. Recognizing that a pattern occurs in the context of other patterns, the context around the pattern is first used to select a subset of nodes in the network. Then from this subset of nodes, the best-matching one is then identified using the pattern part.

The fourth direction should be to investigate the Adaptive-Subspace SOM(Kohonen, p. 161). The Adaptive-Subspace SOM or ASSOM is interesting in that it will detect features even when the source patterns are under-going transformations in space or time. I suggest that this may be a powerful technique when applied to the data that will comprise a database.

The fifth direction for this work is to implement the neural database system using either ART2 or the Fuzzy ART of Carpenter and Grossberg. The reason for wanting to do this is that the ART neural networks determine the number of clusters. This is different from SOFM where the number of clusters is pre-specified.

5.5) Conclusions

This chapter demonstrates the AMM's power in two ways:

1. It provides a general system architecture that was used to develop the AMMda.
2. Its representation scheme, the AMC, has been used to create a Neural Database System.

The AMMda was implemented using CORBA's Static Invocation Interface(SII). The implication of using the SII means that new object types or services cannot be accessed as they become available. However if CORBA's Dynamic Invocation Interface(DII) is used, this will allow any new service to be accessed as soon as it is added to the ORB. The DII allows the implementation of a dynamic distributed object system. The AMM was conceived to function in an ever changing media environment where new services arrive and old services fall by the wayside. The next AMMda will be implemented using the DII as it becomes more widely available in ORB products where the first new service to be added will be a Neural Database System.

Chapter 6: Conclusions and Future Research

6.1) Conclusions

This thesis work was initiated with the primary goal of developing a framework that could provide a disciplined way of thinking about, working with and understanding media. I believe this goal has been realised in the form of the Abstract Media Model(AMM) as presented in Chapter 2 with Chapter 4 demonstrating how the AMM can be used as a design and analysis tool. This primary research goal has generated several other benefits. The first, presented in Chapter 3, is a new media representation, the Abstract Media Code(AMC). The AMC is important as it can represent many media types. The AMC, because of its time code foundation, also serves as a bridge. It plays this role by linking the existing media production community with its huge investment in analogue-based tools, techniques and methodologies to the emerging digital world. The AMC is also an example of convergence in media representations and as such can support the technology convergence we are currently witnessing. Chapter 5 contains the second and third benefits: i) a prototype implementation of the AMM and therefore a design pattern for distributed and non-distributed media systems and ii) a new type of database system, the Neural Database System which utilises the AMC.

Each of these accomplishments was made possible by the fundamental design of the AMM; i.e., a partition-based design(Rumbaugh et al, 1991, p. 201). This type of design has resulted in the AMM providing a fragmented and decontextualised view of media. This fragmenting and decontextualising of media is important as it is the basis for recombination which in turn presents opportunities for technological, social and cultural innovation(De Kerckhove, 1995, p. 201). This thesis has provided several innovations including:

- the decontextualised, service oriented, media model
- the extended time code
- the mediaBus for accessing media services.

These innovations are accompanied by the invention of the neural database system. However these are only the beginning because the AMM has now made explicit the previously hidden media attributes and structures.

In addition to playing the role of an innovation/invention catalyst, the AMM can also play the role of facilitator. The AMM facilitates the discussions of media theoreticians, practitioners, performers and technologists by providing a meeting ground for these groups. This last point is especially important since these groups tend to use different terms and measures. The digital technologists speak about "bits" and "bytes" and "free, ubiquitous Internet" access whereas traditional media practitioners speak about time code, tape and "pay-per-view" scenarios.

In terms of solving issues identified by the media research community, the Abstract Media Model(AMM) provides a framework and paradigm to support distributed multimedia applications(Muhlhauser, Gecsei, 1996). The framework, by capturing the design decisions common to the media domain, supports design reuse over simple code reuse. Within this context the AMMda not only becomes a sample implementation of the AMM but, as mentioned earlier, a design pattern for media applications. As discussed in Gamma, Helm, Johnson and Vlissides this will allow the media application designers and creators to focus on application development. The Abstract Media Code presented in chapter 3 is of use to the hypertext community as they seek ways of working with time-based media(Gronbaek, Trigg, 1994). The AMC could also make a contribution to the "Multimedia Content Description Interface" which is also known as MPEG-7. MPEG-7, the latest member of the MPEG family, has the task of specifying a standardised description of various types of multimedia information to facilitate content searching and retrieval (http://drogo.cselt.stet.it/mpeg/mpeg_7.htm).

As previously stated this thesis was devoted to the development of a model or construct for working with and understanding media. The development of such constructs is not new. One of the earliest examples is Aristotle's model of drama which is now being used to describe, study and develop human-computer activity(Laurel, 1991, p. 49). A model that can be re-deployed approximately 2350 years after its invention is a very useful and robust construct. I believe the AMM has a similar re-deployment potential. The reason for this statement is that the AMM was designed to function within a non-static technological, economic, political, human and social context. Since media, like drama, will always exist and given that the AMM is not dependent or wedded to any particular element of the context, it has the potential to be a useful tool despite the many innovations and changes to the techno-socio-human-political-economic landscape that will take place over time.

6.2) Future Research

This thesis has demonstrated how the newly developed Abstract Media Model can be employed both as an analysis tool and a software design tool. Another role it could perform is that of a re-engineering tool. As a re-engineering tool it could be applied to a corporation's media strategy and how the company works with its media base. It could also be used to re-engineer the design of media software. This leads to my next area of research; i.e., I want to further investigate and develop the distributed symmetric content tools that allow a user to create their media tools and media by selecting the services, as described by the AMM, that they require. I would next like to use these distributed symmetric content tools to implement a media creation environment where the constraints of the seven media attributes can be:

- dynamically imposed during the creation process,
- statically imposed on an existing work.

The former could be realised by placing the AMM in, for example, a digital camera's firmware which would allow a media creator to set the constraints on the creation process. The former could be realised by re-processing an existing media base given an instantiation of the AMM.

Another area of research would be to determine which parts of the Portable Network Graphics(PNG) standard could be included into the AMC. I suspect that my Extended Time Code(ETC) could borrow some elements of the PNG. With these modifications to the AMC in place, my next research target is to modify an MPEG encoder so that it would allow the encoding of the AMC into an MPEG stream. This evolutionary step, as the UC Berkeley MPEG encoder allowed traditional time code to be encoded into the MPEG stream, would be a movement towards creating the foundation of a distributed media edit system as the representation capable of supporting distributed editing would then be in place. I would also like to further research the Neural Database System so that: 1) it is trained on the full AMC; and 2) it becomes an on-line expression service invoke-able via the Dynamic Invocation Interface.

Appendix A: The AMM IDL code

This appendix contains the IDL code for the AMM. This IDL code was used to implement the AMMda. The AMMda only supports text content. It also only allows: storage onto disk, a WWW delivery mode and a hypertext usage paradigm. It supports an ASCII-to-HTML expression and a compression expression. It provides a commerce service for setting and retrieving the price of a medium. It also allows the AMC to be set and retrieved.

The procedure for accessing a service is to first attach the client to the AMM server. This procedure is dependent on the ORB product utilised and returns a pointer. After this attachment activity, use the returned pointer to access the desired service of the AMM. This is where the member functions of the AMMda class are used. To access the storage service perform the following call:

```
storage_service_ptr = ptr->getStorage();
```

```
//
// Commerce structure
//
struct commerce {

string name;    //content or media name.
string owner;   //copyright holder
float  price;   //price of media
string infoTimeStamp; //date and time of commerce info.
string exchange; //name of exchange listing the media.
};
```

```
//
// Extended Time Code Structure
//
struct extendedTimeCode { // extended time code

string who;  // content creator
string what; // content type
string where;// geographic location
string when; // time code given as year, month, day,
             // hour, minute, second, frame
string why;  // description field

};



//
// all bulk data transfers across the mediaBus
// are performed using the byteStream construct.
//

typedef sequence<octet> byteStream;

//
// This class implements the mediaBus.
//
interface mediaBus {

//
// Allocates memory buffer for bytes being placed
// onto the mediaBus.
//
short mbAllocateInBuffer (
                    in unsigned long value  // buffer size.
                    );
```

```
//
// Allocates memory buffer for bytes being taken
// off of the mediaBus.
//
short mbAllocateOutBuffer (
                    in unsigned long value  // buffer size
                    );

//
// places a byte stream on the mediaBus.
//
unsigned long mbByteStreamOut (
     out byteStream data,// the byte stream containing the
                         // data to be sent on the mediaBus.
     in unsigned long value // the size of byte stream
     );

//
// removes a byte stream from the mediaBus.
//
unsigned long mbByteStreamIn  (
     in  byteStream data // byte stream containing data
                         // removed from the mediaBus.
     );

//
// retrieves a pointer to the buffer allocated via
// mbAllocateInBuffer.
//
void  mbGetInBuffer( out string data );

//
// retrieves a pointer to the buffer allocated via
```

```
// mbAllocateOutBuffer.
//
void  mbGetOutBuffer( out string data );

//
// Frees the allocated In buffer.
//
void  mbFreeInBuffer( );

};


//
// This class works with the Abstract Media Code.
//
interface ammAmc {

    //
    // sets the Abstract Media Code.
    //
    void setAMC     (
        in string      usage,
        in string      delivery,
        in string      storage,
        in string      content,
        in string      expressions,
        in extendedTimeCode etc,
        in commerce    com
    );
    //
    // gets the Abstract Media Code.
    //
    void getAMC     (
        in  string     desiredContent, // search key
```

```
        out string     usage,
        out string     delivery,
        out string     storage,
        out string     content,
        out string     expressions,
        out extendedTimeCode     etc,
        out commerce   com
    );

};


//
// The Commerce class
//
interface ammCommerce {
    //
    // sets the commerce attribute
    //
    void setCommerce(
        in commerce com // filled-out commerce structure
    );

    //
    // gets the commerce attribute
    //
    void getCommerce(
        in string name, // search key name
        out commerce com // filled-in commerce structure
    );
};

interface ammStorage; // pre-defined for use in the content
                      // class.
```

```
//
// The content class
//
interface ammContent {

      //
      // Sets the mediaBus pointer. This is needed for
      // the distributed text tool.
      //
      void setTheBus(
            in mediaBus mb // mediaBus pointer
      );

      //
      // Sets the storage attribute pointer. This is also
      // needed by the distributed text tool.
      //
      void setTheStorage(
            in ammStorage s // storage pointer
      );

      //
      // Moves a byte stream to be processed by the
      // distributed text processing engine.
      //
      unsigned long textToolDistributed(
            in byteStream data // byte stream to be processed
      );
};


//
// Delivery class
```

```
//
interface ammDelivery {
      //
      // sets the delivery mode
      //
      short setDeliveryMode(
            in any type // sets delivery type
      );

      //
      // applies the delivery mode.
      //
      short applyDeliveryMode( );
};


//
// Expression Class
//

interface ammExpression {
      //
      // sets the expression
      //
      short setExpression (
            in any type // the expression type
      );


      //
      // applies the expression
      //
      short applyExpression( )
```

```
};


//
// Storage Class.
//
interface ammStorage {

//
// retrieves a byte stream.
//
unsigned long doRetrieval (
out byteStream data, // contains the retrieved byte stream
in unsigned long value // the size of the byte stream
);

//
// stores the byte stream
//
unsigned long  doStorage (
     in byteStream data // byte stream that is to be stored.
);

//
// opens the device holding the byte stream to be retrieved
//
short openRetrievalDevice (
     in any type // retrieval device
);
//
// opens the storage device
//
short openStorageDevice (
     in any type // storage device
```

```
);

//
// sets the retrieval device
//
short setRetrievalDevice (
     in any type // the retrieval device type
);

//
// sets the storage device
//
short setStorageDevice (
     in any type // the storage device type
);

//
// sets the mediaBus pointer for use by the storage engine
//
void  setTheBus (
     in mediaBus mb // the media bus pointer
);

};





//
// Usage class
//

interface ammUsage {
```

```
    //
    // sets the usage mode
    //
    short setUsageMode  (
         in any type // the usage type
    );
};


//
// The ammDa class is needed in order to
// get access to the services of the AMM. This class
// is offered by a server. A client application then
// attaches itself to the AMMda server and accesses the
// needed service.
//
interface ammDa {

ammAmc          getAmc    (); // get AMC class pointer
ammContent      getContent(); // get content class pointer
ammCommerce     getCommerce();// get commerce class pointer
ammDelivery     getDelivery();// get delivery class pointer
ammExpression   getExpression();// get expression class
                                // pointer
ammStorage      getStorage();// get storage class pointer
mediaBus        getBus();// get access to mediaBus

};
```

References

1. Adelstein, P.Z.(1996). " ***STANDARDS ON THE PERMANENCE OF RECORDING MATERIALS"*** in **Standards for Electronic Imaging Technologies, Devices, and Systems**, Michael C. Nier ed., Volume CR61. SPIE Optical Engineering Press: Bellingham, Washington, USA.

2. Anderson,D.P.(1993). " ***Metascheduling for continuous media"*** , ACM Transactions on Computer Systems, vol.11, no. 3, Aug 1993, p. 226-252.

3. Avid (1993). ***Open Media Interchange Specification Version 1.0.***, Avid Technology Inc., March 1993.

4. Bahr, L.S., Johnston, B., eds.(1992). ***Collier's Encyclopedia.*** Vol. 19, p. 30D, Toronto:Maxwell Macmillan, 1992.

5. Becker, H.S.(1995). " ***Library of Congress Digital Library Effort"***, Communications of the ACM, Vol. 38, No. 4, April 1995, p. 66.

6. Begault, D.R.(1994). ***3-D Sound for Virtual Reality and Multimedia***. Academic Press: Toronto, 1994.

7. Berners-Lee, T., Cailliau, R., Luotonen, A., Frystyk Nielsen, H., Secret,A.(1994). " ***The World Wide Web"*** , Communications of the ACM, Vol. 37, No. 8, August 1994, pp. 76-82.

8. Berners-Lee, T., Fielding, R., Frystyk, H.(1995). ***Hypertext Transfer Protocol -- HTTP/1.0,*** http://ds.internic.net/internet-drafts/draft-ietf-http-v10-spec-04.txt, 1995.

9. Bharath, R., Drosen, J.(1994). ***Neural Network Computing***. McGraw-Hill: New York, New York, 1984.

10. Blair, G.S., Coulson, G., Davies, N.(1994). " ***System Support for Multimedia Applications: an assessment of the state of the art"*** , Information and Software Technology, vol. 36, no. 4, 1994, pp. 203-212.

11. Black, U.(1991). ***OSI, A model for Computer Communications Standards.*** Prentice-Hall Publishing Compnay: Englewood-Cliffs, New Jersey, 1991.

12. Borenstein, N.(1989). ***Multimedia Applications Development with the Andrew Toolkit.*** Prentice-Hall: Englewood Cliffs New Jersey, 1989.

13. Bradner, S. and Mankin, A.(RFC 1752). *The Recommendation for the IP Next Generation Protocol,* RFC 1752.

14. Bridgwater, W. and Kurtz, S. eds., ***The Illustrated Columbia Encyclopedia.*** New York: Columbia University Press, 1956.

15. Britnell, J., McCarthy, C., Samuels, S.(1993). " ***Interdisciplinary Collaboration: The development of an Interactive Multimedia Module to Teach Communication Skills"*** , Interpersonal Skills Teaching Centre Group, Ryerson Polytechnic University working paper, November, p. 3.

16. Brown, A.J.(1995). " ***What is a Data Warehouse?"*** , Unix Review, August 1995, Vol. 13, No. 9, pp. 39-44.

17. Buchanan, M.C., Zellweger, P.T.(1992). " ***Specifying Temporal Behaviour in hypermedia documents"*** , in **Proceedings of the Fourth ACM Conference on Hypertext**. D. Lucarella, J. Nanard, M. Nanard and P. Paolini, eds. ECHT'92 (Milano, Italy, Nov. 30 - Dec. 4, 1992), pp. 63-70.

18. Buford, J.F.(1996). " ***Distributed multimedia information systems"*** in **Standards for Electronic Imaging Technologies, Devices, and Systems**. Michael C. Nier ed., Volume CR61. SPIE Optical Engineering Press: Bellingham, Washington, USA.

19. Bulterman, D.C.A.(1993), " ***Specifying and support of adaptable networked multimedia"*** , 1, 2(1993) Springer-Verlag/ACM Multimedia Systems, pp. 68-76.

20. Bulterman, D.C.A., van Rossum, G., van Liere, R.(1991). " ***A structure for transportable, dynamic multimedia documents"***, in **Proceedings of the Summer 1991 USENIX Conference**. Nashville, Tenn., June 1991, pp. 137-155.

21. Cacace, F., Lamperti, G.(1996). ***Advanced Relational Programming***. Kluwer Academic Publishers: Dordrecht, 1996.

22. Campbell, B., Goodman, J.M.(1988). " ***HAM: A general purpose hypertext abstract machine***" , Communications of the ACM 31, 7(July), pp. 856-861.

23. Carpenter,G.A., Grossberg,S.(1987a). " ***A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine***", Computer Vision, Graphics and Processing, 37:54-115.

24. Carpenter,G.A., Grossberg,S.(1987b). " ***ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns***" , Applied Optics, 26:4919-4930.

25. Caudell, T.P., Smith, S.D.G., Escobedo, R., Anderson, M.(1993). " **NIRS: Large Scale ART-1 Neural Architectures for Engineering Design Retrieval**", *Neural Networks*, Vol. 7, No. 9, pp. 1339-1350, Pergamon Press Ltd.

26. CCITT Recommendation I.320, ***ISDN protocol reference model***, Available at http://www.itu.ch/itudoc/itu/rec/i/i320_25235.html

27. Cerf, V.G.(1991). " **Networks**", Scientific American, September 1991, Volume 265, Number 3, pp. 72-81.

28. Cheung, N.K.(1992). " ***The Infrastructure for Gigabit Computer Networks***", IEEE Communications, April 1992, Vol. 30, No. 4, pp. 60-68.

29. Colaitis, F.(1994). " ***Opening Up Multimedia Object Exchange with MHEG"*** , IEEE Multimedia, Summer 1994, pp. 80-85.

30. De Kerckhove, D.(1995). ***The Skin of Culture***. Somerville House Publishing: Toronto, 1995.

31. Delgrossi, L., Berger, L.(1994). ***Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+***, RFC 1819.

32. Deloddere, D., Verbiest, W., Verhille(1994). " ***Interactive Video On Demand"*** , IEEE Communications, Vol. 32, No. 5, May 1994, pp. 82-88.

33. Dolson, M.(1986). " ***The Phase Vocoder: A Tutorial"***, Computer Music Journal, Vol. 10, No. 4, Winter 1986, pp. 14-27.

34. Eckel, B.(1995). ***Thinking in C++***. Prentice-Hall: Englewood Cliffs, New Jersey, 1995.

35. Fausett, L.(1994). ***Fundamentals of Neural Networks: architectures, algorithms and applications***. Prentice-Hall: Englewood-Cliffs, New Jersey, 1994.

36. Fleming, C.(1996). " ***The Predator"***, Vanity Fair, February 1996, pp. 147.

37. Fortner, B.(1992). ***BRD-028 Television Technical Theory Instructor Notes***. Ryerson Polytechnic University, School of Radio and Television Arts, 1992.

38. Frye, N.(1993). ***The Educated Imagination***. Anansi: Toronto, 1993.

39. Gamma, E., Helm, R., Johnson, R., Vlissides, J.(1995). ***Design Patterns***. Addison Wesley Publishing Company: Reading Massachusetts, 1995.

40. Genesereth, M.R., Ketchpel, S.P.(1994). " ***Software Agents***", Communications of the ACM, Volume 37, Number 7, July 1994, pp. 49-53.

41. Gibbs, S.J., Tsichritzis, D.C.(1994). ***Multimeda Programming: Objects, Environments and Frameworks***. Addison Wesley Publishing Company: Reading, Massachusetts, 1995.

42. Globe and Mail, Report on Business, Jan. 22, 1996, p. 2.

43. Globe and Mail(1994). ***Three Clues about how we'll all be riding the I-way.*** July 2, p. D8.

44. Gorlen, K.E., Orlow, S.M., Plexico, P.S.(1991). ***Data Abstraction and Object-Oriented Programming in C++***. John Wiley and Sons: Toronto, 1991.

45. Gronbaek, K., Trigg, R.H.(1994). " ***Hypermedia system design applying the Dexter Model***" , Communications of the ACM, 37, 2(Feb. 1994), pp. 26-29.

46. Halasz, F.G. and Schwartz, M.(1990). " ***The Dexter Hypertext Reference Model"*** , NIST Hypertext Standardisation Workshop (Gaithersburg, MD, 16-18 January), pp. 95-133.

47. Hardman, L., Bulterman, D.C.A, van Rossum, G.(1994). " ***The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model"*** , Communications of the ACM, 37, 2(Feb. 1994), pp.50-64.

48. Heller, R., Martin, C.D., " ***A Media Taxonomy"***, IEEE Multimedia, Vol. 2, No. 4, pp. 36-45.

49. Hodges, M.E., Sasnett, R.M.(1993). ***Multimedia Computing, Case Studies from MIT Project Athena***. Addison Wesley Publishing Company: Reading Massachusetts, 1993.

50. Huber, D.M.(1987). ***Audio Production Techniques for Video***. Howard W. Sams & Company: Indianapolis, 1987.

51. Huston, G.(RFC 1744). ***Observations on the Management of the Internet Address Space***. RFC 1744.

52. Intriligator, M. D.(1978). ***Econometric Models, Techniques & Applications.*** Prentice-Hall: Englewood-Cliffs, New Jersey, 1978.

53. Jacobson, V., Braden, R.T. and Borman, D.A. (1992). ***TCP Extensions for High Performance***. RFC 1323.

54. Jezequel,J.M.(1996). ***Object-Oriented Software Engineering with Eiffel***. Addison Wesley Publishing Company: Don Mills Ontario, 1996.

55. Kaliski, B.S.(1991). " ***A Layman's Guide to a Subset of ASN.1, BER, and DER"*** , RSA Data Security Inc.

56. Kelber, W.(1995). " ***Language, Memory and Sense Perception in the Religious and Technological Culture of Antiquity and the Middle Ages"*** , Oral Tradition, Volume 10, Number 2, October 1995, pp. 409-450.

57. Kesik, T.(1996). ***Knowledge Mapping.*** http://www.ryerson.ca/~bsc/kmapmain.html

58. Kim, S.(1995). " ***Stock Imagery Surfs the Net"*** , Videography, Fall, 1995, pp. 82-88.

59. Koegel Buford, J.F.(1994). ***Multimedia Systems***. Addison Wesley Publishing Company: Don Mills Ontario, 1994.

60. Kohonen, T.(1995). ***Self-Organizing Maps***. Springer-Verlag: Berlin, 1995.

61. Laurel, B.(1991). ***Computers as Theatre***. Addison Wesley Publishing Company: Don Mills Ontario, 1991.

62. Leffler, S.J., McKusick, M.K., Karels, M.J., Quarterman, J.S.(1989). ***The Design and Implementation of the 4.3BSD Unix Operating System***. Addison Wesley Publishing Company: Reading Massachusetts, 1989.

63. Lehtinen, R.(1994). " ***Everything's Under Control: A brief history of machine control, past, present and future"***, AV Video, April, Vol.16, No. 4, 1994, pp. 24-32.

64. Lesk,M.(1994). " ***Experiments on Access to Digital Libraries: How can Images and Text be Used Together?"*** , in **Proceedings of the International Conference on Very Large Data Bases**. Santiago Chile Sept. 12-15, 1994.

65. Lison, H., Crowley, T.(1989). " ***Multimedia Systems: Sight and Sound"*** , UNIX Review, Vol. 7, No. 10, October 1989, pp. 76-86.

66. Little, T..C., Ghafoor, A.(1993). " ***Interval-Based Conceptual Models for Time-Dependent Multimedia Data"*** , IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No. 4, August 1993.

67. Loomis, M.E.S.(1987). ***The Database Book***. Macmillan Publishing Company: New York, New York, 1987.

68. Lougher, P.,Shepherd, D.(1993). " ***The design of a storage server for continuous media"*** , The Computer Journal, vol. 36, no. 1, 1993, pp. 32-42.

69. Lucky, R.(1991). ***Silicon Dreams: Information, Man and Machine***. St. Martin's Press: New York, New York, 1991.

70. Magel, M.(1994). "***Author! Author!***", AV Video, January, Vol. 16, No. 1, pp 68-77.

71. McLuhan, M.(1969). ***The Gutenberg Galaxy***. The New American Library: New York, New York, 1969.

72. McLuhan, M., Fiore, Q.(1967). ***The Medium is the Massage***. Bantam Books: Toronto, Canada, 1967.

73. McMillen, K.(1994). " ***ZIPI: Origins and Motivations*** " , Computer Music Journal, vol. 18, no. 4, Winter 1994, pp. 47-51.

74. Meyer-Boudnik, T., Effelsberg, W.(1995). "***MHEG Explained***", IEEE Multimedia, No.1, Vol.2, 1995, pp. 26-38.

75. Minton, G.(1996). " ***Programming with CORBA*** " , Unix Review, No.4, Vol. 14, pp. 29-39.

76. Muhlhauser, M., Gecsei, J.(1996), " ***Services, Frameworks, and Paradigms for Distributed Multimedia Applications*** " , IEEE Multimedia, Vol.3, No. 3, Winter 1996, pp.48-61.

77. Newcomb, S.R.(1995). "***Multimedia Interchange Using SGML/HyTime***", IEEE Multimedia, Vol.2, No.2, Summer 1995, pp 86-89.

78. Newcomb, S.R., Kipp, N.A., Newcomb, V.T.(1991). " ***HyTime: The Hypermedia/Time-Based Document Structuring Language*** " , Communications of the ACM, November 1991, Vol. 34, No. 11, pp 76-83.

79. Newell, A.(1986), ***The Knowledge Level***, Readings from AI Magazine, 1980-1985, pp 357-377.

80. Nielsen, J.(1990). ***Hypertext and Hypermedia***. Academic Press: Toronto, 1990.

81. Norman, D.A.(1991). " ***Cognitive Artifacts"*** in **Designing Interaction: Psychology at the Human-Computer Interface**. J.M. Carrol Ed. Cambridge University Press, New York, 1991.

82. Norman, D.A., Draper, S., eds.(1986). ***User Centered System Design: New Perspectives on Human-Computer Interaction***. Lawrence Erlbaum: Hillsdale, N.J.

83. Ong, W.J.(1995). " ***Hermeneutic Forever: Voice, Text, Digitization and the " I" "*** , Oral Tradition, Volume 10, Number 1, March 1995, pp. 3-26.

84. O'Shaughnessy, E.(1994). " ***The New Establishment"*** , Vanity Fair, Vol. 57, No. 10, pp. 209-240.

85. Padlipsky, M.A.(1985). ***The Elements of Networking Style***. Prentice-Hall: Englewood Cliffs, New Jersey, 1985.

86. Pandya, A.S, Macy, R.S.(1996). ***Pattern Recognition with Neural Networks in C++***. CRC Press: Boca Raton, Florida.

87. Potel, M., Grimes, J.(1991). " ***Guest Editors Introduction: Multimedia- It's Actually Useful"***, I.E.E.E. Computer Graphics and Applications, July 1991, pp. 24-25.

88. Prunty, P.F.(1994), " ***Delivery of TV Over Existing Phone Lines"***, *in* **Unveiling New Technologies and Applications - Selected Papers from the SMPTE Advanced Televisions and Electronic Imaging Conference**. Chicago, Illinois, U.S.A., February 4-5, 1994, pp. 82-99.

89. Raiffa, H.(1982). ***The Art & Science of Negotiation.*** Belknap/Harvard: Cambridge, Massachusetts, 1982.

90. Reddy, A.L.N., Wyllie, J.C.(1994). " ***I/O Issues in a Multimedia System"*** , Computer, vol. 27, no. 3, Mar 1994, pp. 69-74.

91. Roe, P., Soulis, G.N., Handa, V.K.(1969). ***The Discipline of Design.*** University of Waterloo: Waterloo, Ontario, Canada, 1969.

92. Rona, J.(1990). ***SYNCHRONIZATION from reel to reel.*** Hal Leonard Publishing: Milwaukee, Wisconsin, 1990.

93. Rose, M.T.(1996). ***The Simple Book.*** Prentice-Hall: Englewood Cliffs, New Jersey, 1996.

94. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W.(1991). ***Object-Oriented Modelling and Design.*** Prentice-Hall: Englewood Cliffs, New Jersey, 1991.

95. Samuelson, P.(1995). " ***Copyright and Digital Libraries"***, Communications of the ACM, Vol. 38, No. 4, April 1995, pp. 15-21-110.

96. Schmandt-Besserat,D.(1979). " ***L'invention de L'ecriture"***, Les imaginaires, II, Cause Commune(1979):119-30.

97. Schmidt, D.C.(1996). " **An Overview of the Common Object Request Broker Architecture(CORBA)"**, World Wide Web, http://www.cs.wustl.edu/~schmidt.

98. Schmidt, D.C., Vinosky, S.(1995). " ***Introduction to Distributed Computing***", C++ Report Magazine, January 1995.

99. Schubin, M.,(1995). "***Unreal***", Videography, fall, 1995, pp. 20-25.

100. Sherman, M., Hansen, W.J., McInerny, M., Neuendorffer, T.(1990). " ***Building Hypertext on a Multimedia Toolkit: An Overview of Andrew Toolkit Hypermedia Facilities***" in **HYPERTEXT: CONCEPTS, SYSTEMS AND APPLICATIONS, Proceedings of the European Conference on HYPERTEXT**. A. Rizk, N. Streitz and J. Andre eds., INRIA, France, November, 1990. Cambridge University Press, New York, 1990.

101. Shneiderman, B.(1992). ***Designing the User Interface: Strategies for Effective Human-Computer Interaction, Second Edition***. Addison-Wesley: Reading, Massachusetts, 1992.

102. Siegel, J.(1996). ***CORBA: Fundamentals and Programming***. John Wiley & Sons, Inc.: New York, New York, 1996.

103. Simon, H.(1981). ***The Sciences of the Artificial***. Cambridge Massachusetts, MIT Press, pp. 159.

104. Sloan,D.(1994). ***Aspects of Music Representation in HyTime/SMDL***. Computer Music Journal, vol. 17, no. 4, Winter 1994, pp. 51-57.

105. Stanford Digital Libraries Group(1995). "***The Stanford Digital Library Project***", Communications of the ACM, Vol. 38, No. 4, April 1995, pp. 59-60.

106. Stein, M.J.(1994). "***DIGITAL MULTI-MEDIA: Now That I've Got It, Where Do I Put It?***", in **Proceedings of The SMPTE Advanced Televisions and Electronic Imaging Conference**. February 1994, pp. 64-71.

107. Steinmetz, R.(1995). ***Analysing the Multimedia Operating System***. IEEE Multimedia, Spring 1995, pp. 68-84.

108. Stevens, W.S.(1994). ***TCP/IP Illustrated, Volume 1: The Protocols***. Addison-Wesley: Reading, Massachusetts, 1994.

109. Stroustrup, B.(1986). ***The C++ Programming Language***. Addison-Wesley: Reading, Massachusetts, 1986.

110. Suhanic, W.(1996). "***The Abstract Media Model***" in **Proceedings of the Seventh Annual Conference of the International Information Management Association**(IIMA'96), pp. 59-69, Estes Park Colorado, Dec 4-6, 1996.

111. Systems Design Engineering(1991). ***Graduate Studies and Research, Department of Systems Design Engineering***. University of Waterloo, April 1991, p. 1.

112. Tanenbaum, A.S.(1981). ***Computer Networks***. Prentice-Hall: Englewood Cliffs, New Jersey, 1981.

113. Tarnas,R.(1991). ***The Passion of the Western Mind***. Crown Publishers Inc.: New York, New York, 1991.

114. Technology Policy Working Group, Committee on Applications and Technology, Information Infrastructure Task Force(1994). " ***Services and the National Information Infrastructure***", Draft for Public Comment, Dec. 1994.

115. Tokuda, Hideyuki(1994). "***Operating System Support for Continuous Media Application***" *in* J.F. Koegel Buford(ed.), **Multimedia Systems.** pp 201-220. New York: Addison-Wesley.

116. Tversky, B.(1995). ***"Cognitive Origins of Graphic Productions"*** in F.T. Marchese(ed.), **Understanding Images: Finding Meaning in Digital Imagery.** pp 29-53. Santa Clara: Springer-Verlag, TELOS.

117. Ullman, B.L.(1969). ***ANCIENT WRITING AND ITS INFLUENCE.*** MIT Press: Cambridge Mass., 1969.

118. Wilensky, R.(1995). " ***UC Berkeley's Digital Library Project***" , Communications of the ACM, Vol. 38, No. 4, April 1995, p. 60.

119. Winston, P.H.(1984). ***Artificial Intelligence, Second Edition.*** Addison-Wesley: Reading, Massachusetts, 1984.

120. Wright, R.(1993). ***An Approach to Knowledge Acquisition, Transfer and Application in Landscape Architecture.*** http://www.clr.utoronto.ca:1080/PAPERS/kmap.html

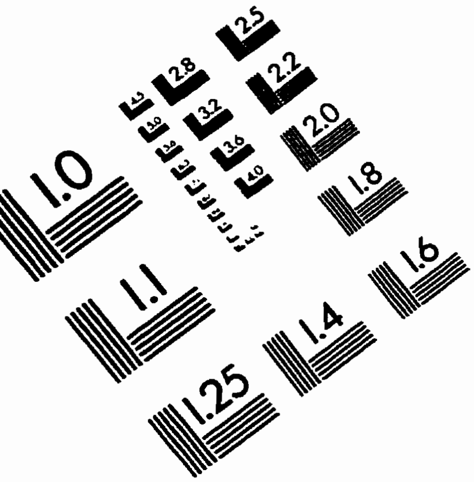
121. Woodhead, N.(1991). ***Hypertext and Hypermedia: Theory and Application***. Addison-Wesley: Wokingham, England, 1991.

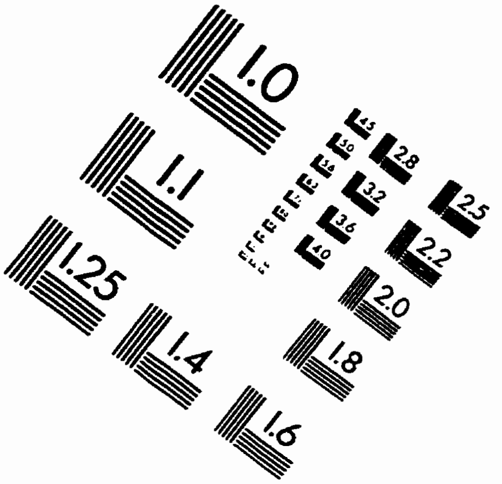
122. Yager, R.R., Filev, D.P.(1994). ***Essentials of Fuzzy Modeling and Control***. Wiley-Interscience: Toronto, 1994.

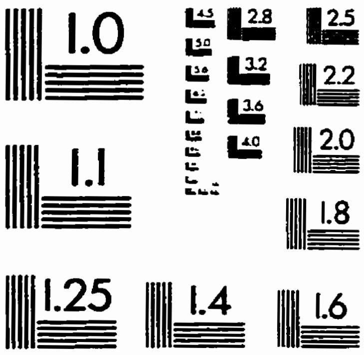
123. Zettl, H.(1992). ***Television Production Handbook***. Wadsworth Publishing Company: Belmont,Ohio 1992.

124. Zimmermann, H.(1980). "***OSI Reference Model--The ISO Model of Architecture for Open Systems Interconnection***", IEEE Transactions on Communications, Vol. COM-28, No. 4, April 1980, pp. 425-432.

IMAGE EVALUATION TEST TARGET (QA-3)

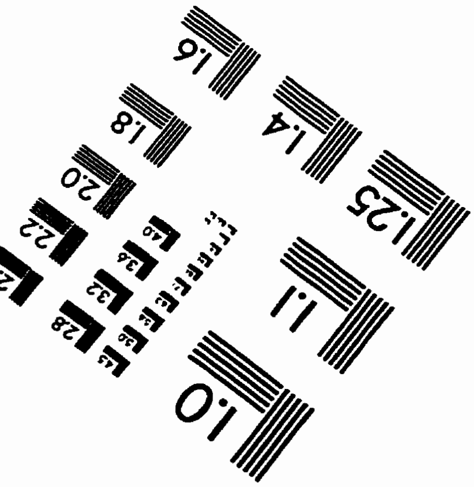


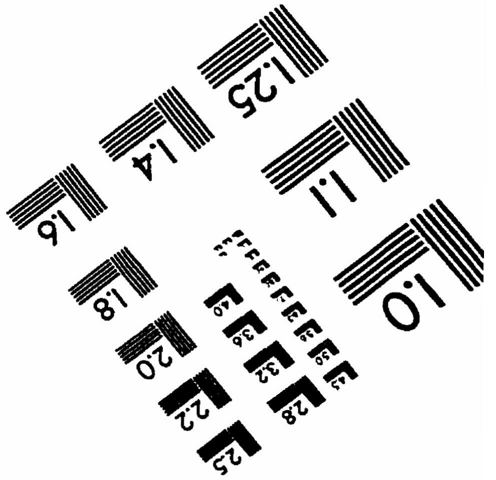




150mm

6″





APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989