Multimedia Communications

by

Terrence P. McGarty Ph.D.

Massachusetts Institute of Technology

Fall 1989

CONTENTS

PREFACE

This text deals with the many issues that relate to the field of multimedia communications. Multimedia communications is a new field that concerns the creation, manipulation, transmission and processing of complex images, text, graphics, voice and other multi media entities. Unlike the fields of computer communications or even voice communications, the multimedia environment presents a wide variety of new and complex challenges. The very nature of the message, complex images as well as voice, and the more complex issues of human interfaces, make this a challenging field.

This text develops the field by first providing a structure to the multi media environment and then developing a theory for its processing, transmission and manipulation. The field of multimedia communications includes the issues of human interfaces, characterizations of complex images, the abstractions of distributed data base and distributed operating systems as well the interaction of multi users in a multi media environment.

This text grew out of the work that the author has been involved in over a ten year period. This time period included involvement in such divers areas as CATV, personal computers, image processing, broadband communications and ultimately multi media systems. It had become quite clear over this period that multimedia communications was becoming a technology unto itself, addressing many issues that combined the interests of many existing fields but also asking questions that had not yet been posed. The key to understanding multi media communications is not to understand how computers communicate but how humans communicate. Thus the experience that the author has developed in the context of a multitude of end user applications has played a critical role in addressing the issues that led to the development of this book.

This book has been written for the use of students who are studying communications, computer systems or media applications. It is also useful for those practicing in the field and what are developing new and innovative multi media systems. It is also hoped that it can be used ti inspire many of those who have entrepreneurial tendencies to implement many of the theories discussed in this book.

The author would like to thank many of the people who have helped him in the development of this text. (etc)

Terrence P. McGarty
Waterville Valley, NH
September, 1989

CHAPTER 1

Introduction

Communications has evolved from the simplest form of telegraphy to voice and video and into the efforts associated with computer communications. The challenge of communications is now to deal with the ability to use multiple media simultaneously. For example, if we look at the way creative graphics designers function, we see that there is high interaction between the images, the spoken word, the use of facial remarks and most importantly the movement of the hands. The ability to fully communicate the creative process is the essential element of the multimedia communication design.

1.1 Definition of Multimedia

Media, for the purpose of this book, represents any form of storage or presentation means that allows for the transmission of information from one user to another. Information in its more classic sense is basically the reduction of ambiguity or entropy in the understanding of some concept. Thus the use of a picture, a a way of expressing a thousand words, is an alternative means of providing information through a specific form of media. Images, speech, text, graphics are all forms of media. The simultaneous use of all of these forms of media in an session dedicated to information transfer is the essence of multimedia communications.

The history of computer communications is based upon the less than complex transmission of information by means of simple binary messages and increasing the complexity to that of text and simple graphics. The introduction of more complex image formats is increasing and it is the use of these formats that dramatically changes the way we communicate in a multimedia fashion. In a simple text communications format, whether it be telex of even high speed computer networking, the user is forced to follow the standards of the computer network. The messages are packetized, sending a single message, a single letter or at most a single set of words at a time. The introductions of graphics packages were really a way to attempt to provide images within the constraints of the computer network. True image communications, encompassing the full interactivity of a multimedia information exchange has been severely limited.

A true multimedia multiuser information exchange occurs in a typical ad copy approval session between the client, the ad agency, the publisher and event the pre-press house. The communications involves the use of the images, the use of the hands and voice and the interaction of eyes, body language, charts, graphs, numbers, and finally even the printed words. Multimedia communications attempts to develop the theory and structure associated with such information transfer transactions between sets of individuals. Computer communications has typically focused on information transmission ans transaction between computers and infrequently tolerated the interaction with humans, but always on their terms.

At the hear of multimedia communications is the interpretation that this new media can bring to bear on our very relationship to what we perceive as knowledge. Marshall McLuhan, the academic who was made famous by his book "The Medium is the Message" is quoted by Peter Drucker at the time he was defending his own doctoral thesis. At that time McLuhan was discussing how Gutenberg had dramatically changed the character of the middle ages with the introduction of the printing press. As he developed the theses to his academic reviewers, they all shook their heads in agreement, that is until he reached his final conclusion. That conclusion, McLuhan stated was that the change in presentation of information changed not only how knowledge was transferred, but more importantly. WHAT WAS KNOWLEDGE. The medium of transfer of information actively altered what was information.

We frequently do not readily understand what McLuhan was saying and look at knowledge on a human time scale. If we were to go back 2,000 years we would see that knowledge was limited to what could be memorized or at most what was kept on the fragile documents in the few libraries such as the one at Alexandria. If we look at the knowledge of those times we find a strong oral tradition and the result was the limitation of ideas to those that had a simple oral rendering. If we look at Newton's works on gravity we find the vestiges of that oral tradition. The presentations in written form still follow the oral form with limited equations and even fewer limited diagrams. In current books on gravitation, there is a plethora of equations and the diagrams are complex and much more enlightening. As we step even further, the use of super computers can now allow these theories to be displayed on high resolution displays in dynamic form. The knowledge of such thing as fluid flow now is viewed not just as equations, graphs and figures, but as the flow of simulated fluids through simulate boundaries.

As we move towards the current days with the use of film and video, we find that knowledge is viewed as what is on film. The anecdotes, whether they are true or not, of recent presidents believing reports only if they are on film rather than written, show again how the change in available media change what is knowledge. Knowledge is the image or vision of the camera and its ability to position certain ideas in the eyes of the viewer.

As we move forward in the use of new media, especially those that allow for multimedia and multiuser interactions, we further build upon McLuhan's concept that changing media changes what is knowledge. The book as a learning medium will change. The book generally is a static medium that is linear in form, progressing from chapter to chapter. The book is built up by the author to present a sequence of concepts to the user. The book is non interactive, it does no allow for the asking of questions and in response provide a new reordering. As Nathan Felde has stated, education is a low bandwidth process, entertainment is a high bandwidth process. Education is a questioning nonlinear an iterative process that frequently uses many media. Entertainment is a nonquestioning linear process that lacks significant interactivity. We must recognize that the multimedia user frequently is trying to be educated, that is transfer and absorb information, not be passively a recipient of high bandwidth entertainment.

Figure 1.1 presents a sample multimedia communications system.

Figure 1.1 Multimedia Communications Systems

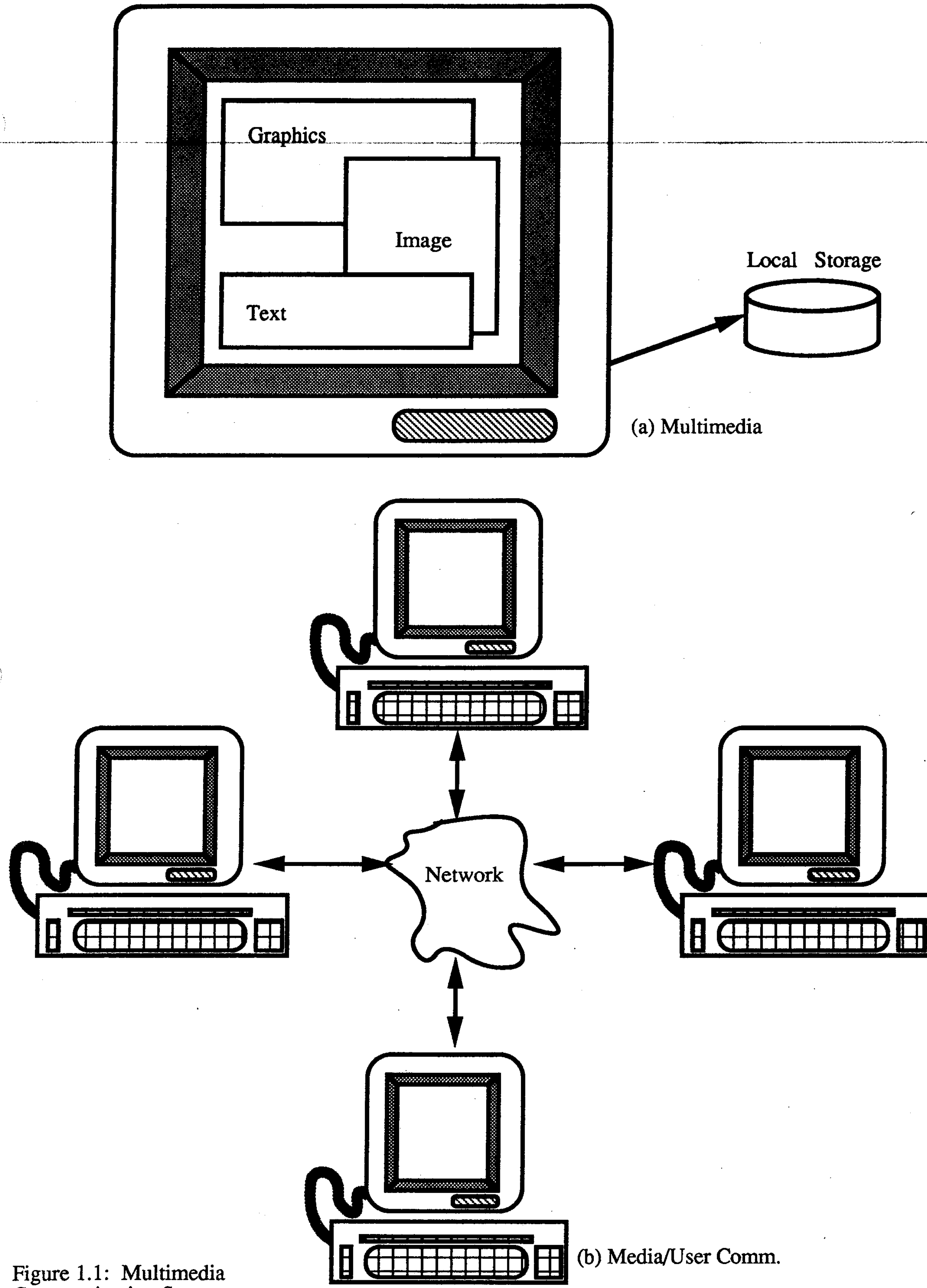


Figure 1.1: Multimedia Communication Systems

1.1.1 The Image

We shall expand the concept of an image to mean the embodiment of any media based information bearing element. For example, an image may be a picture, a segment of a voice conversation, a piece of graphics display, or even a segment of binary data. In the general sense, if we let M represent any such image, then we define the image in terms of is extent, E(x,y,z,t), its information content, I, and its location L(x,y,z,t).

The image will be at the heart of the multimedia communications environment. It is the generalization of the basic element that we shall consider at part of the transfer of information. In a voice only environment, the corresponding element is the conversation, the concatenation of the words, inflections, pauses that make up the communications from one individual to another. In a computer communications world the corresponding element is the dat file and the transactions that relate to that file. As we expand to the multimedia environment we extend beyond that to the more complex interaction of user with information in many forms.

1.1.2 The User

In a multimedia communications system, the user is generally not the large computer or other information processor but is the human. This dramatically reorients the focus on what is to be communicated and how it is to be communicated. The users objective in a multimedia communications environment is to use the information and to develop and understanding on how the many parts of the information can be used in achieving the desired goal.

1.1.3 Interaction

The interaction allowed in a multimedia environment is dramatically different than that in most other communications systems. As we noted before, the standard form of communications is highly structured and is linear in fashion. Computer communications is based upon the need for well established and agreed to protocols that permit the users to interact in a controlled fashion. The users in the computer world are generally other computers or in some cases humans who must adhere to the computers well structured dialogue. In contrast, the interactions in a multimedia environment are highly unstructured, significantly nonlinear in their form and are driven by the more creative side of the human user. Images are complex representations of knowledge elements and the ultimate information is a construct of the combination of the images, the user and the interactions that are developed.

1.2 Current Systems

There are several simple but representative multimedia system that have been developed in the past few years. Figure 1.2 depicts the overall structure of some of these systems. The first contain work stations or computers that allow for the use of many types of media, integrating graphics, still images and other types of images into the same screen. The introduction of a window environment allows for the use of many of these media on a stand alone work station. The inclusion of a networking capability further enhances the ability of these work stations to share the resources. For example, a file server is a memory storage device that allows for shared memory to be used by many users on a local area network. The server may have any image type stored on the system and one or several users may window the image onto their customized screen format.

Figure 1.2 Sample Current Systems

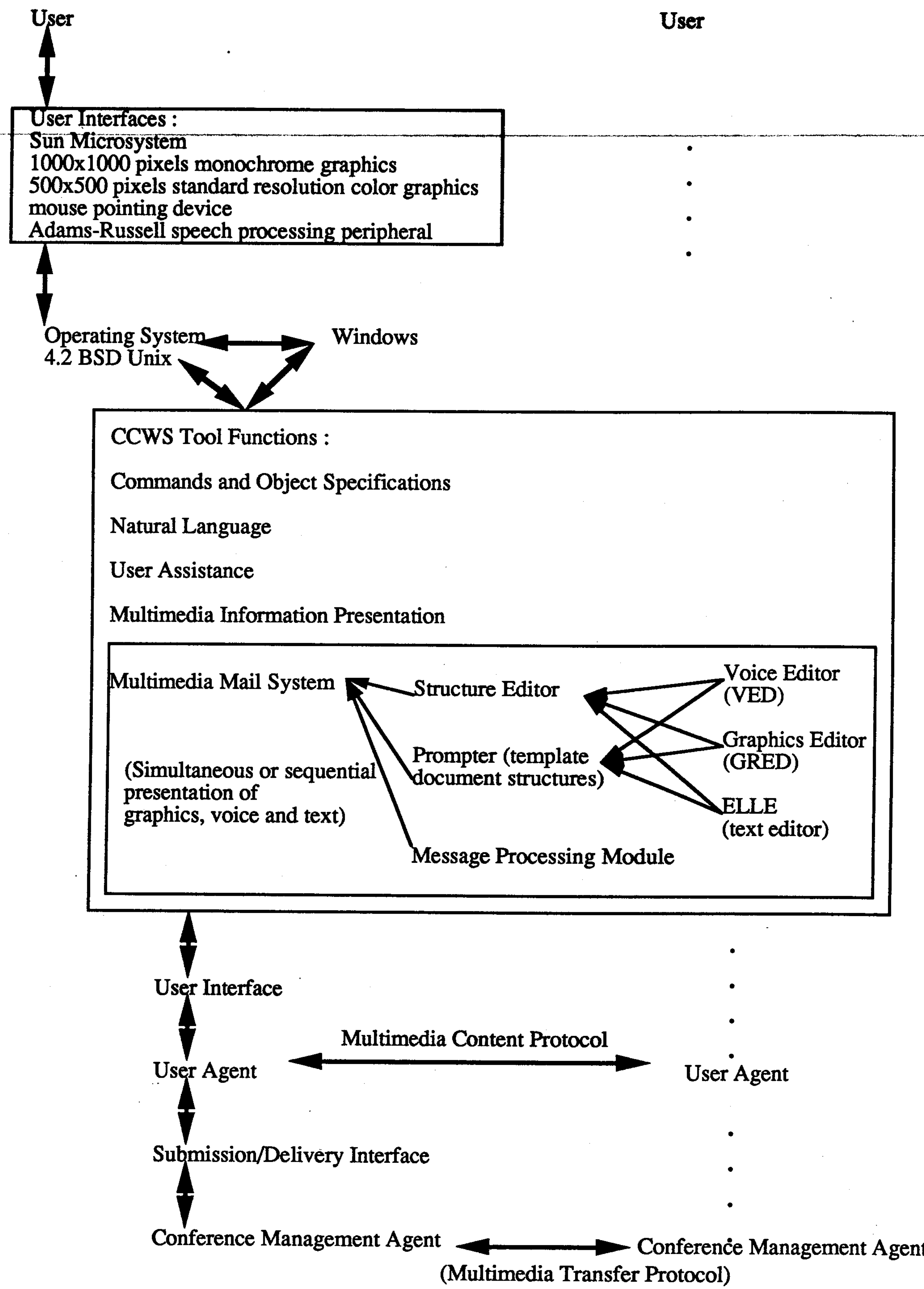


Figure 1.2.a : Sample Current Multimedia Systems - the Command and Control CCWS System.

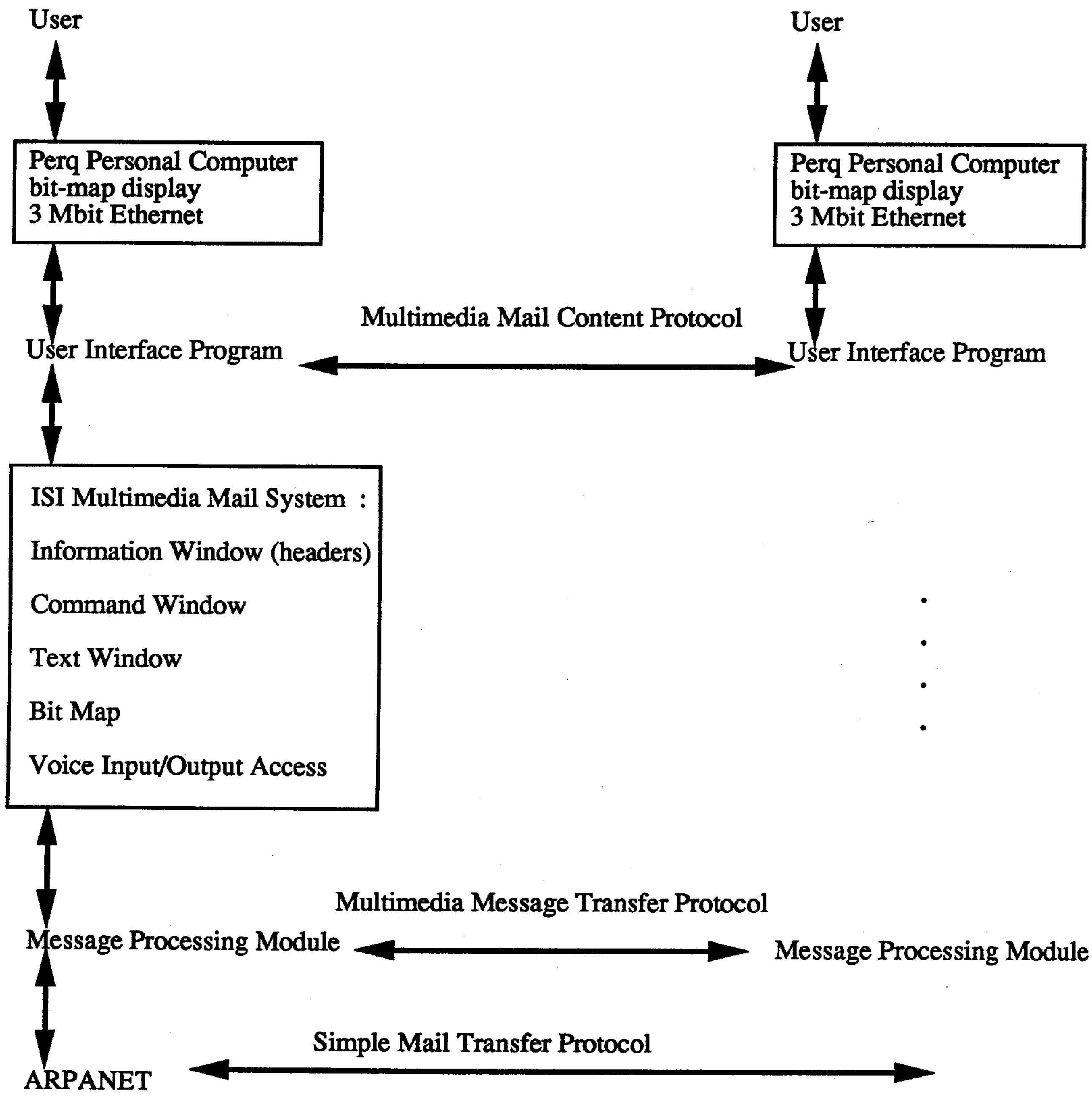
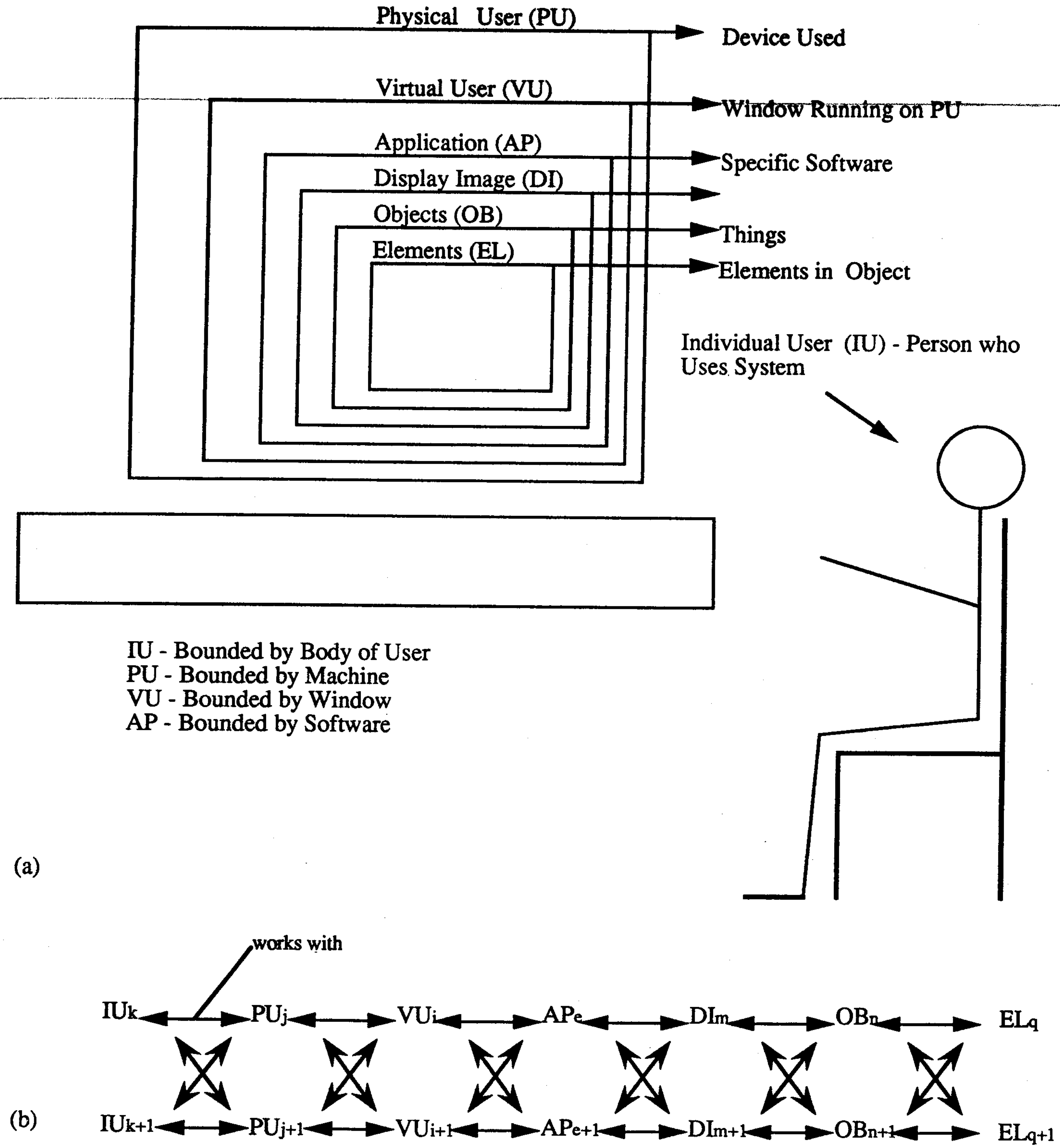


Figure 1.2.b : Sample Current Multimedia Systems - the DARPA Experimental Multimedia System.

1.2.1 Computers

The multimedia work station or computer has ben developing over the past few years and a typical design is that of the Apple MAC II system. The MAC II uses a simple window structure with a high resolution bit mapped scree for display. Figure 1.3 depicts the overall hardware and software architecture of the MAC II system. We shall see that this type of work station architecture is typical of many of the first generation multimedia work stations.

Figure 1.3 Sample Multimedia Computers



System where item can be described as:

(IUk, PUj, VUi, APe, DIm, OBn, ELq) is a seven-tuple system

(c)

Figure 1.3

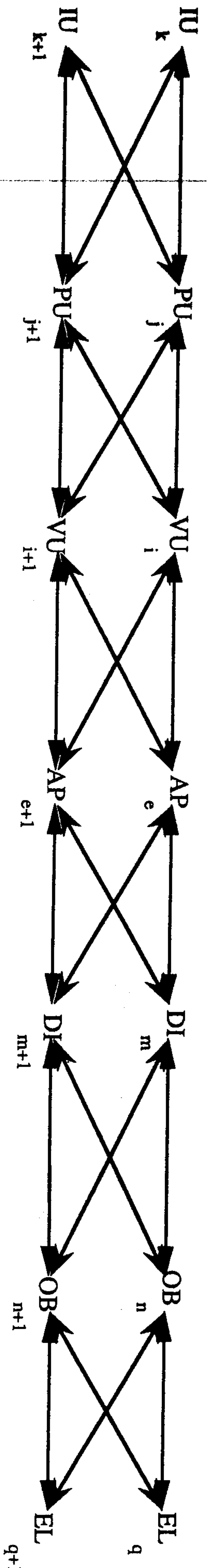


Figure 1.3b: communication between units of system

1.2.2 Presentation Formats

The first generation multimedia systems allow for a wide set of presentation formats. The systems will support graphics, text, still images, video of various forms, digitized displays, and integrated voice with the display. Also some systems under development allow for the integration of voice and image as an integrated whole. Some systems allow for the integration and manipulation of various sound elements.

Many of the efforts in the MIT Visual Language Workshop, under Prof. Muriel Cooper, show the capability of combining the visual with sound of many forms. The systems displayed in this area are presented in several cases in Figure 1.4.

In a multimedia environment, it is possible to interact with all of the human senses and more importantly to play one off against another. We can exchange sound for intensity and color for the pitch of speech. In the current first generation systems, there is considerable effort using the concept of false colors to bring out new information in visual data that was no there in a simple viewing. A typical example is what can be done using extensive color imaging in the medical area. Currently all X-rays, CAT scans and MRI scans are in gray scale images. A gray scale image quite simply is a system that put at each picture element (pixel) a dot whose intensity can vary from pure black to pure white and has many shades in between. For example the display at each pixel may have 10 bits and thus there could be 1024 different shades of gray. Depending on the human, the eye could resolve anywhere between 256 to 4096 shades of gray.

Now it is possible to expand the gray scale imaging of human tissue by adding color, wherein the color corresponds to tissue type and its intensity to type density. Thus it is conceivable that an MRI scan could look at the paranasal sinuses and determine not only the differnt masses and size of tissues but could also identify particular tissue types. The discriminants to do that are available today in the magnetic resonance imaging data.

Figure 1.4 Sample Integrated Presentation Formats (Muriel Cooper's)

1.2.3 Interaction Formats

The end user can interact with the images in many forms. The standard first generation interaction mechanisms include the keyboard, the pen and pallet, the touch screen, and even a voice activated display. Some extreme forms of interaction have included the are/hand point and speak methods developed in the MIT media Lab. The latter types represents some of the first attempts to make the interaction more in line with the activities of the human user. Remembering that multimedia communications attempts to match the end users flexible forms of inputs, the need exists for many types of advanced interaction formats. Attempts have been made to provide for speech response formats that allow for hands off interaction. The problem generally with these types of devices is that they displace the problem of input to a higher human level of making the user coordinate two separate reflexes, sight and sound.

Considerable effort is still required to develop interaction devices and systems to more effectively match the human user. The classic anecdote is the scene in the movie Star Trek, where the chief engineer Scotty, sent back sever hundred years in time, asks for a computer and is shown an Apple MAC. He picks up the mouse and speaks into it and says, "Computer". He is then told that that is a mouse and he must use it to point and the key board to enter the commands. He comments, "How quaint". Yet it is this interaction element that will be the main diver of the McLuhan change of media and knowledge.

The interaction with the current first generation systems focuses on the interaction of the human user with the stand alone work station. We can envision extending this by having a network of resources and other users and consider the interaction problems in this multimedia multiuser communications environment. We shall, in the remainder of this book, develop the concepts necessary to show how this more complex system of interaction may evolve.

1.3 The Paradigm

The development of a theory for multimedia communications requires the development of a paradigm or world view of the multimedia environment. We have defined the concept of an image and we must now combine together several key concepts that will be sued over again in the book as we develop the multimedia communications environment. All multimedia communications relate to images. Yet those images ultimately relate to the interaction of human users and their mutual manipulation of those images.

We shall now define the key elements of the multimedia paradigm and show how those elements add to a working model for multimedia multiuser communications.

Figure 1.5 The Multimedia Paradigm

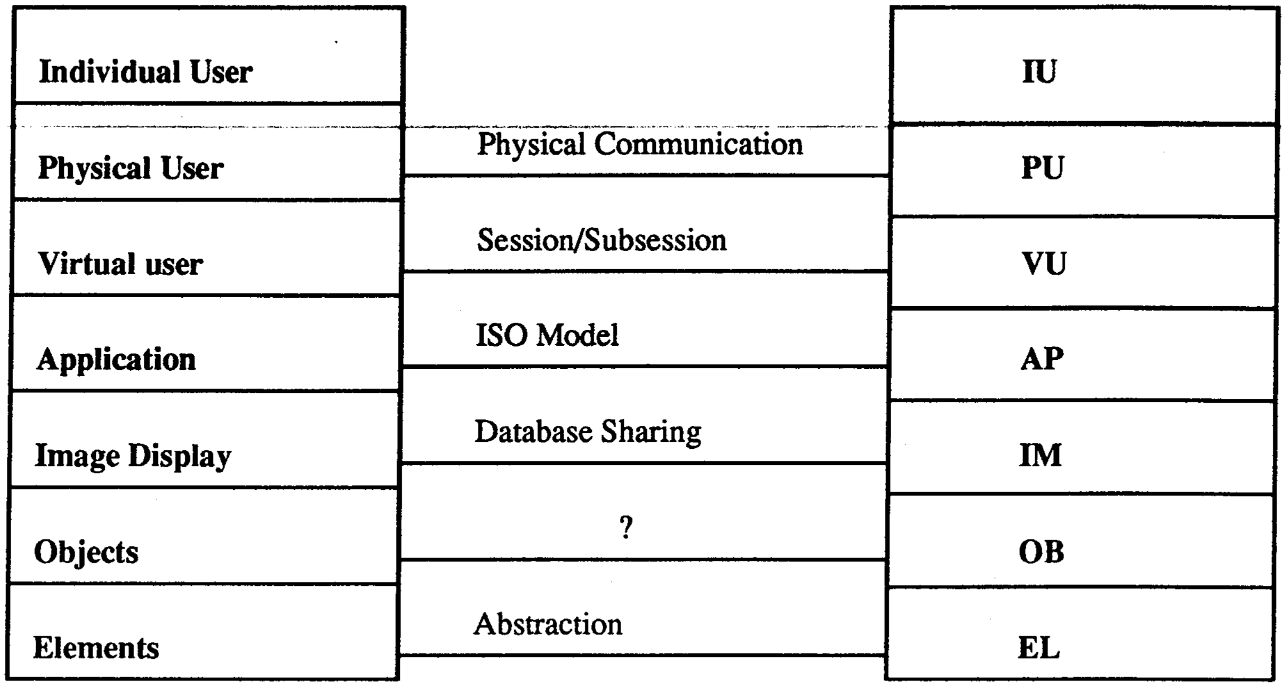


Figure 1.5: The Multimedia Paradigm

1.3.1 Individual User

The human that interacts wit information is termed the individual user, IU. In any communications interaction we can envision one or moor individual users communicating. The individual user in multimedia communications becomes a key element in the communications channel. The individual has an identity that is separate and apart from the other elements that are frequently the key elements in a computer based communications example. Even in the standard world of voice communications there is no recognition of the individual directly. The telephone directory gives only the name of the resident responsible for the payment of the bill. In multimedia communications, we recognize individuals separately and apart from the other elements in the communications system.

1.3.2 Physical User

The first element in the model is that of the physical user. The physical user is a physical device that is used by an individual for the purpose of communicating with others. Clearly, we can envision that the individual may have access to several physical devices and in turn a single physical device may relate to several individual users.

The physical user, P, may be one of several physiacal devices that are attached to the network. These devices may be terminals, file servers, imaging devices or there elements. They are unique and are specifically located at a specific location.

1.3.3 Virtual User

The virtual user, V, is a concept that becomes an important element in the multimedia environment. Any physical user may open one or more virtual users on the network. To some degree, the concept of a window is analogous to that of a virtual user. Any physical user may have several virtual users connected to the single physical device.

1.3.4 Application

An application is a specific program that is operative within a physical user environment. For example, in a CAT scan environment, the physical user is the CAT scan system, the virtual user is the specific allocation of resources to a set of scans and the applications may be a combination of both two and three dimensional CAT scan analyses. The virtual user window can open up for the analysis of a specific patient. Thus the identification of a virtual user in this case is with the patient, and within that virtual user is allocated two applications programs that process the CAT scan data and another application program that is used for the maintenance of the hospital records.

1.3.5 Image

The image is the physical embodiment of the multimedia subject to be used in the discussions between users in the session. For example the image could be a picture scanned from a video data base, a graphics image generated from a raster or geometric data format, or even a voice message that is interactively generated between the two users.

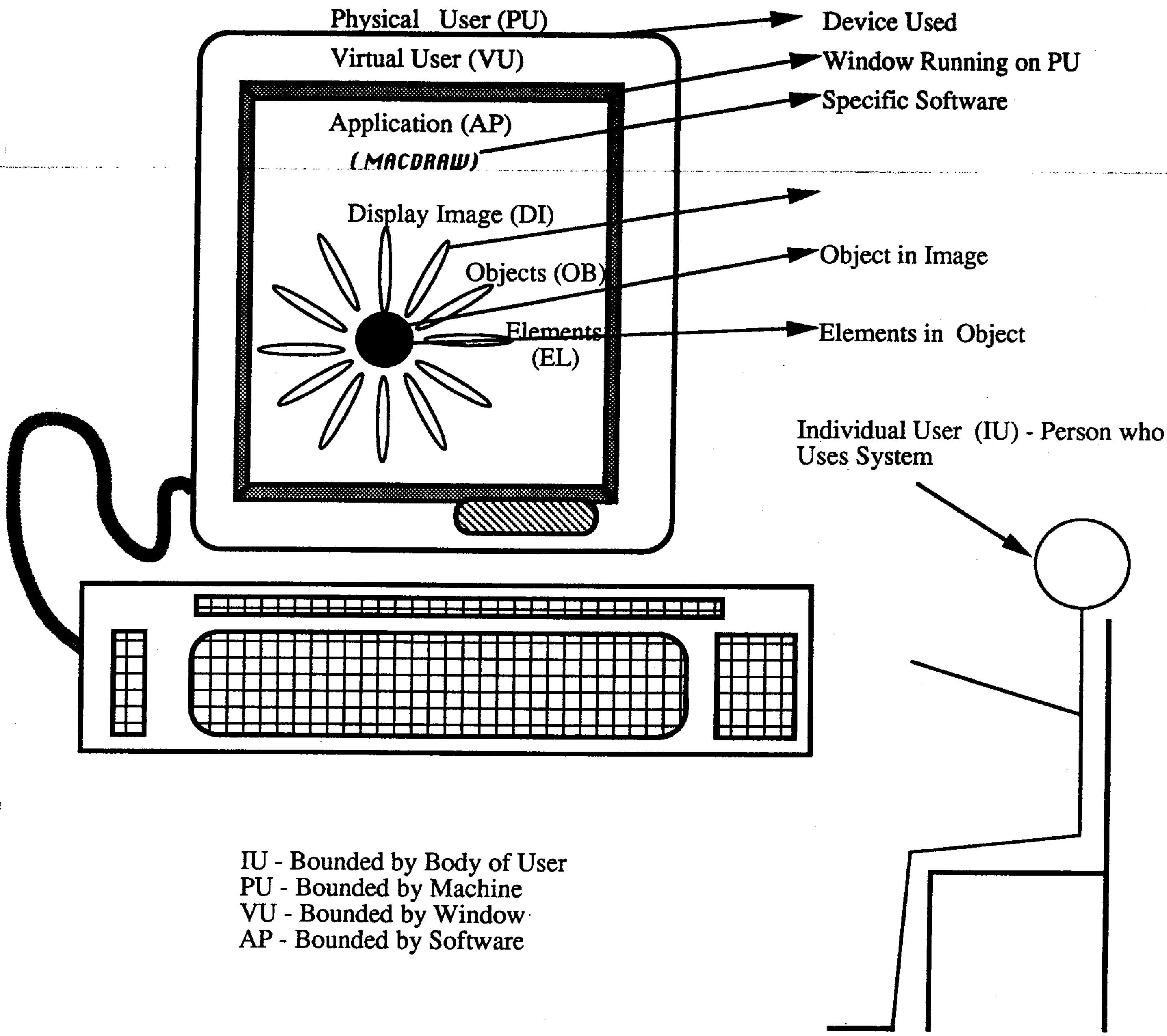
1.3.6 Object

The object is the first step towards abstraction of the image. Whereas the image is the totality of the multimedia message, the object is a bounded portion of the image that can be user definable. Thus if the image is considered to be the full landscape of a photograph, the object may be the part of the landscape that is near a stream or mountain. Objects present localization of images. A further example is that in the general sense the image may be a total conversation, whereas the object may be a portion of that conversation.

1.3.7 Element

The element is the part of an object that can be abstracted to a specific definable abstraction. For example in an image of an X-Ray, there can be a specific object called the spine, and a specific element called the fifth cervical vertebra. The ability to now abstract the fifth cervical vertebra and use this in comparison in MRI, CAT and radiographic studies is the key issue associated with elements.

Figure 1.6 Multimedia Elementals



(a) Structure of Data Management System

System where item can be described as:

(IUk, PUj, VUi, APe, DIm, OBn, ELq) is a seven-tuple system

(c) Specifics of tuple system

Figure 1.6: Multimedia Elementals

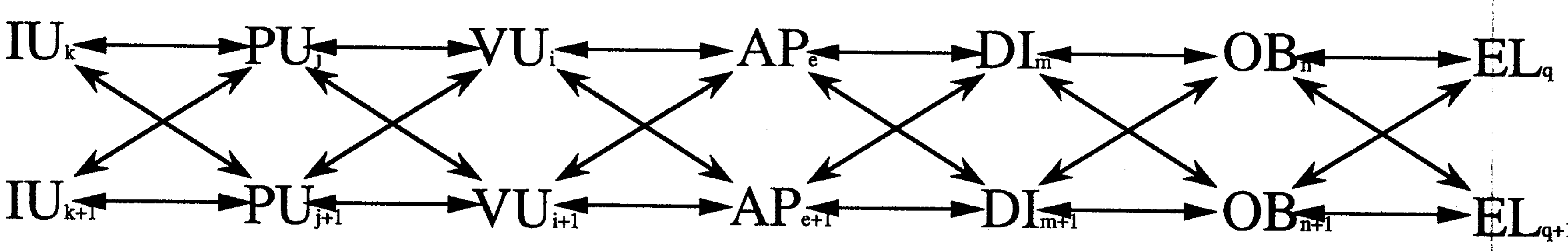


Figure 1.6b: Communication Between Units of System

1.3.8 SubSession

A sub-session is a connection between two virtual users. It is the fundamental bilding block for multimedia multi user communications. What is important

1.3.9 Session

The session is the key concept in the multimedia communications environment. A session is quite simple a collection of one or more sessions. In a session, a set of virtual users, and their associated individuals and physical users, are linked together into a common communications process. In this book we shall be developing all of the structure necessary to develop the session and to support the session as it evolve in time. A session may last an indefinite period of time and it can survive the coming and going of any of its constituent sub-sessions.

All interaction occurs within the context of a session. Figure 1.6 shows how a session is developed as a build up of many sub sessions. In addition the subsessions allow for the adding of any set of virtual users, each of whom may have their own objects, elements or even applications.

Figure 1.6 Development of Sessions and Subsessions

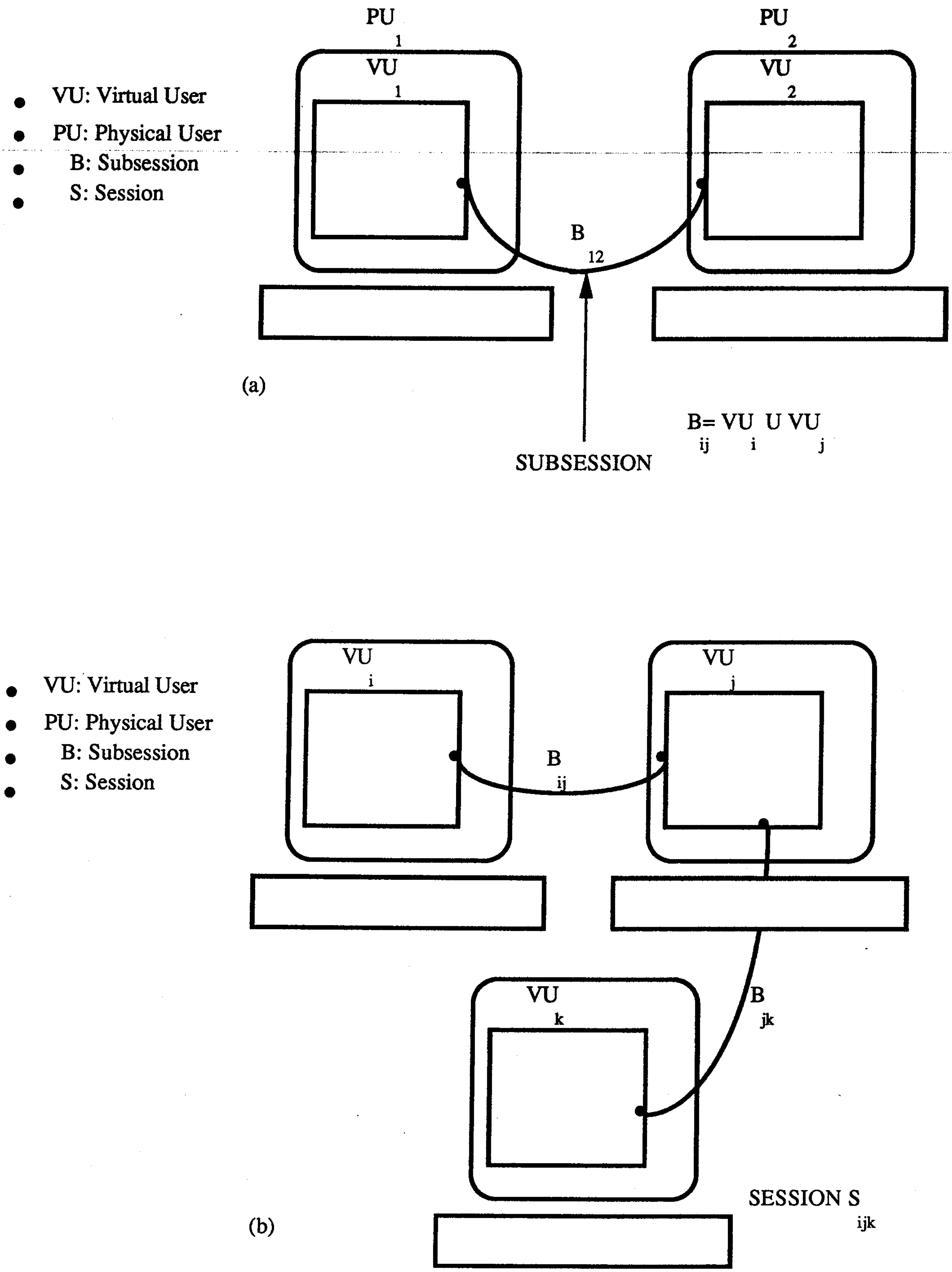


Figure 1.4 Development of Subsession and Session

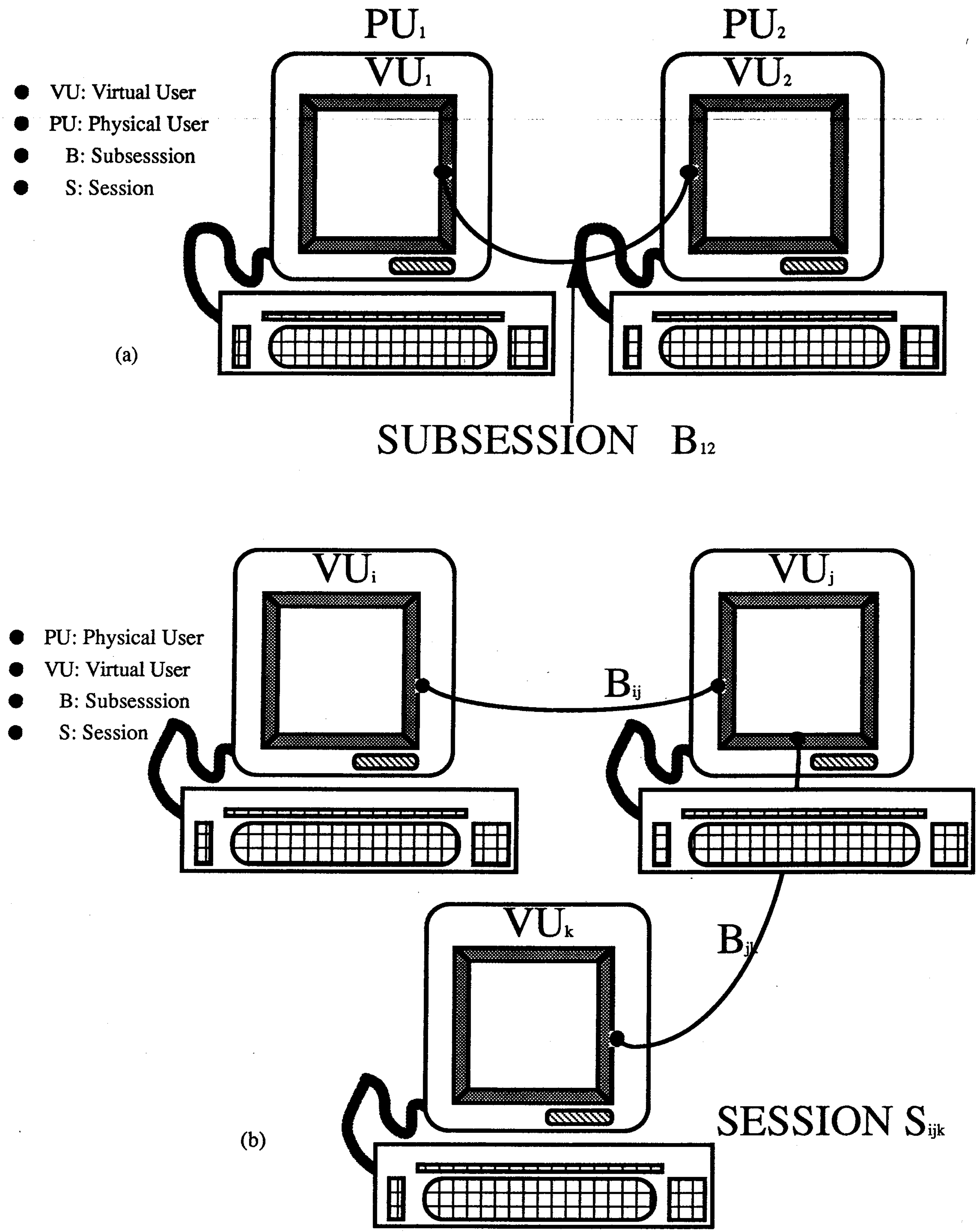


Figure 1.7: Development of Subsession and Session

1.4 Elements of Multimedia Systems

We have discussed the structure of the current first generation multimedia systems and have discussed the way in which they process data on the work station, display the images and integrate them and provide for interaction with the user. The first generation systems are generally limited in what they can do in these three areas. In this section we shall develop the basic elements of the multimedia communications systems and attempt to anticipate what is possible in the second generation of such systems.

Figure 1.7 Multimedia Layering and Interfaces

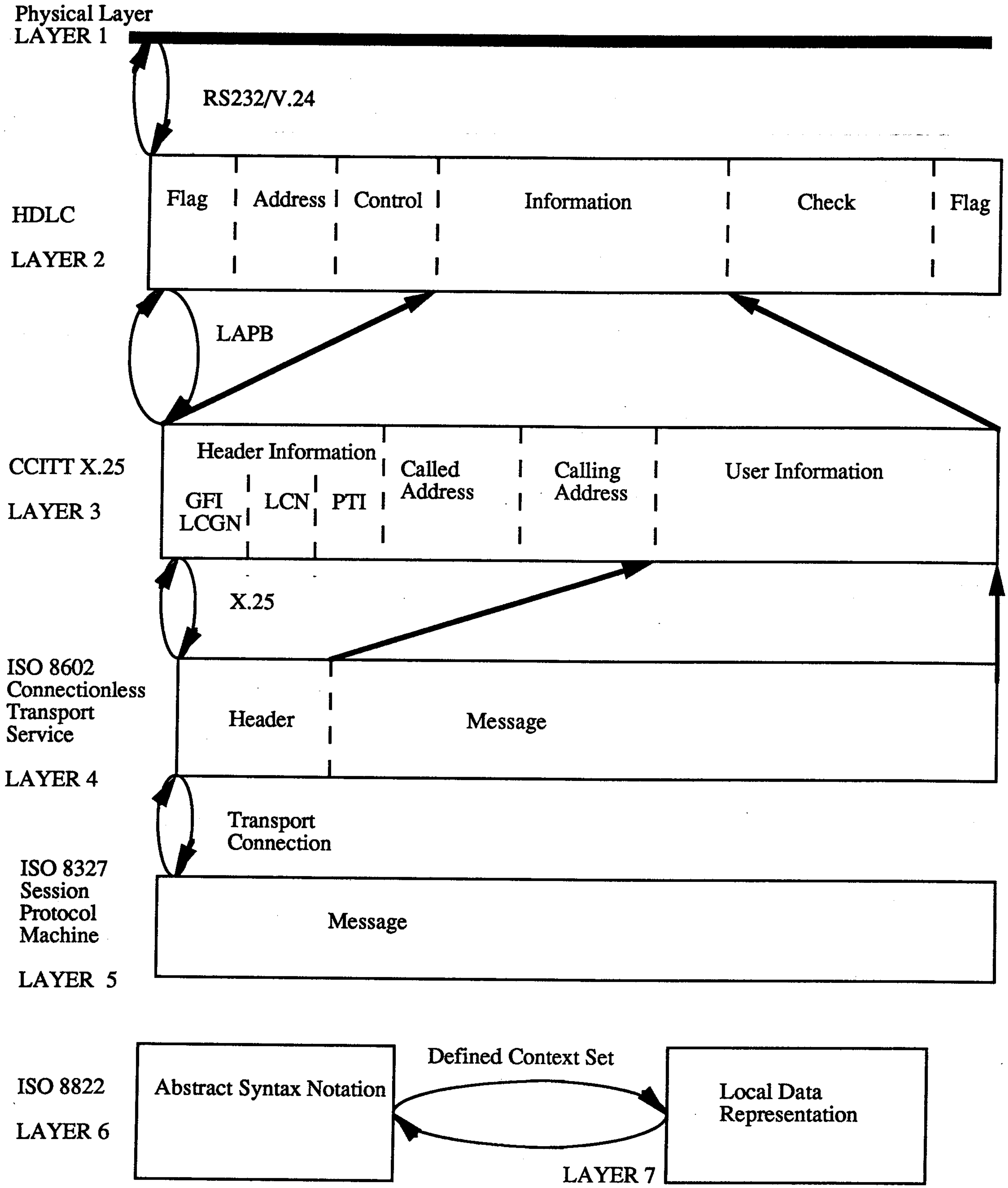


Figure 1.8.a : Multimedia Layering and Interfacing - an Example Based on a Digital Network.

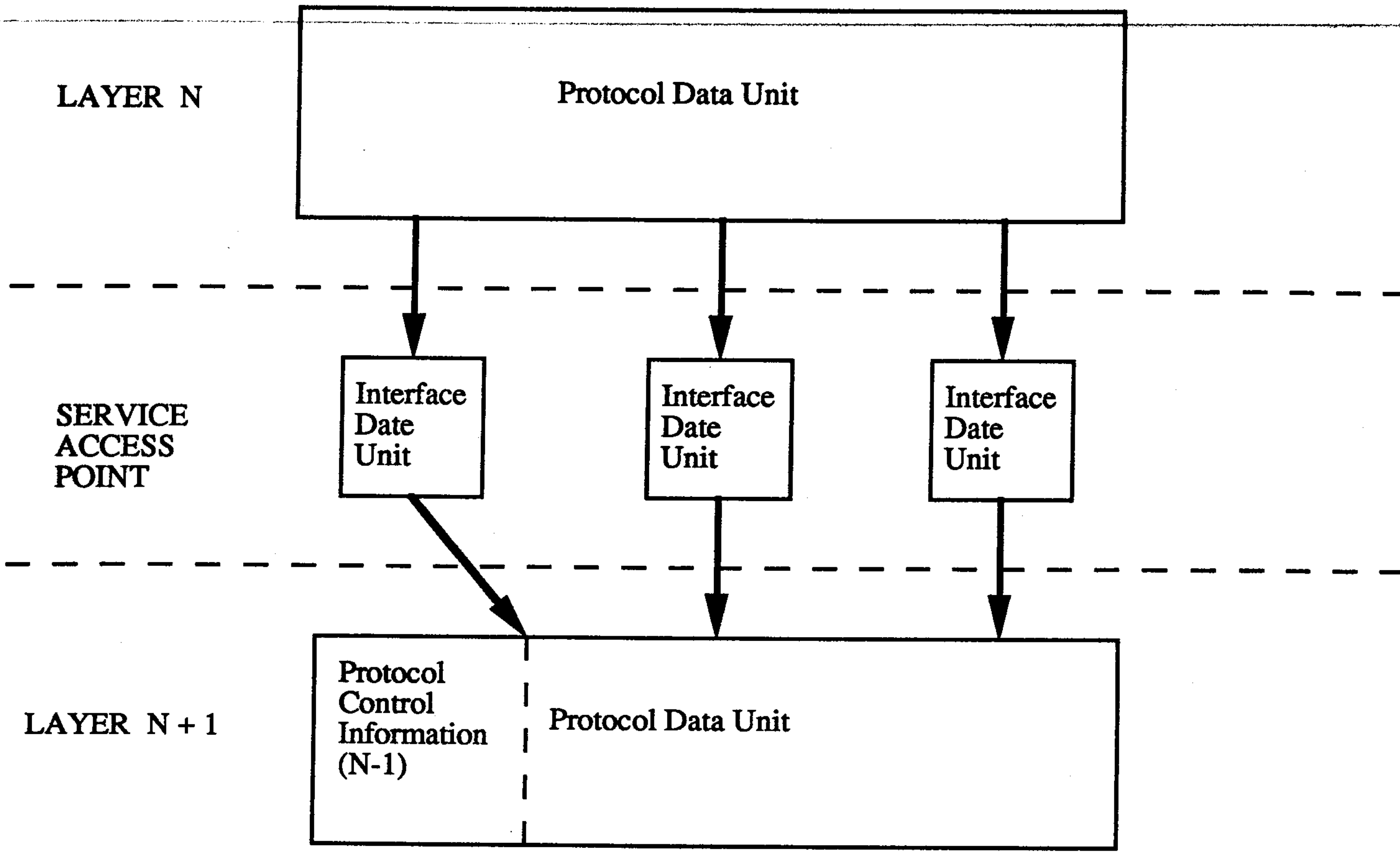


Figure 1.8.b : Multimedia Layering and Interfaces - Interface between two ISO Layers.

Link Protocol Control Information	Network Protocol Control Information	Transport Protocol Control Information	Session Protocol Control Information	Protocol Control Information and Data in Transfer Syntax	Link Check Sequence

Figure 1.8.c : Multimedia Layering and Interfaces - Layering in the Data Packets.

1.4.1 Display

The display for the first generation systems has generally been a bit mapped display whose size has been limited to about 1000 by 1000 dots. This yields about a million pixels in a display. The pixel may be up to 12 bits deep, so that a single display has 12 million bits. In current production as the first steps towards a second generation display are 2000 by 2000 units of 4 million pixels, each of 24 bits per pixel. This yields 100 million bits per screen. This approaches the display capability of a 35 mm slide.

These second generation displays now must have significant buffer storage to keep the image and must have the bandwidth at the display driver to keep the image refreshed at the rate on 30 times a second. This implies a transfer rate of 3 Gbps. Figure 1.8 shows the general structure of a display architecture, one that hold for both first and second generation multimedia displays.

Figure 1.8 Display Architecture

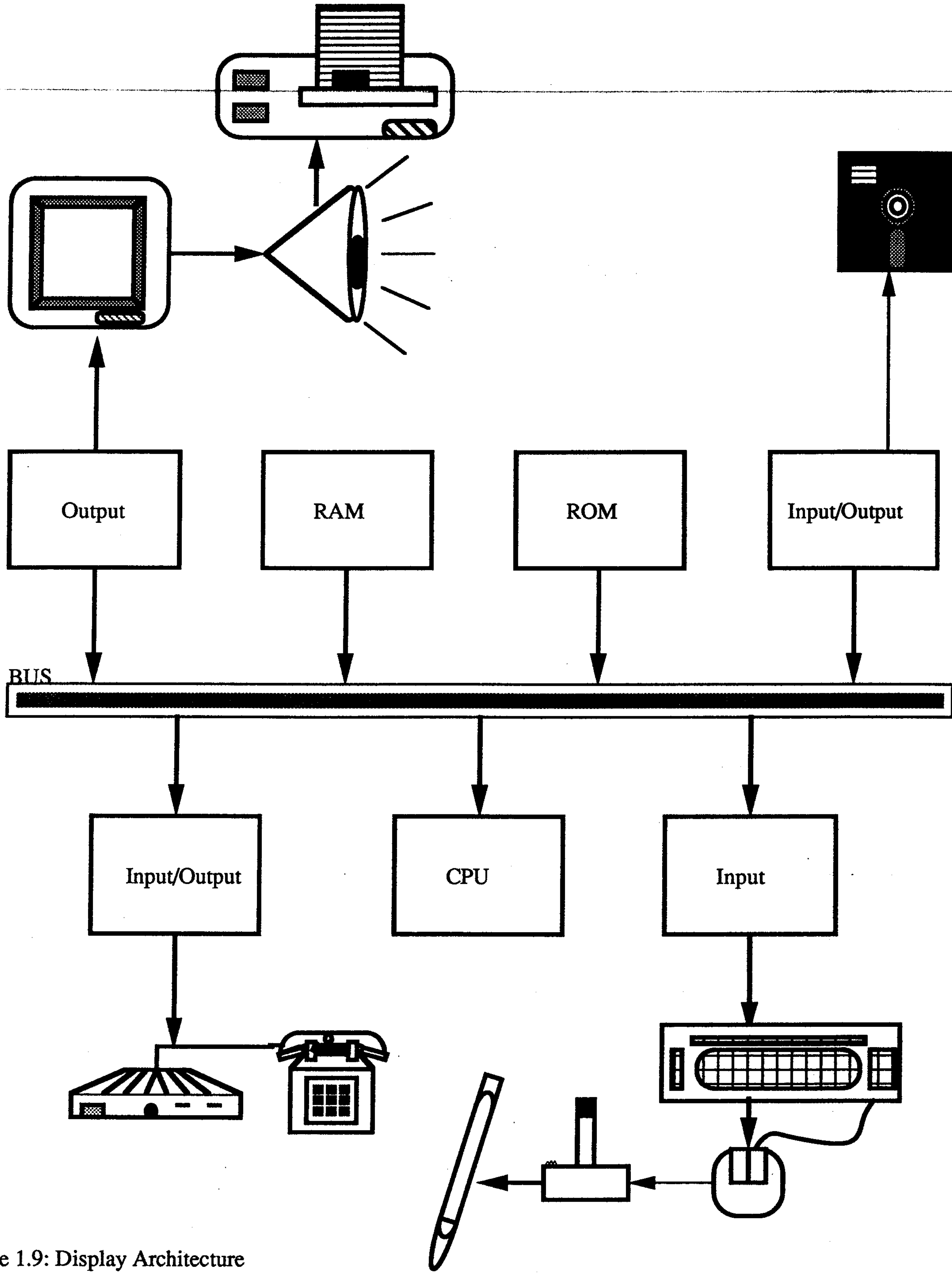
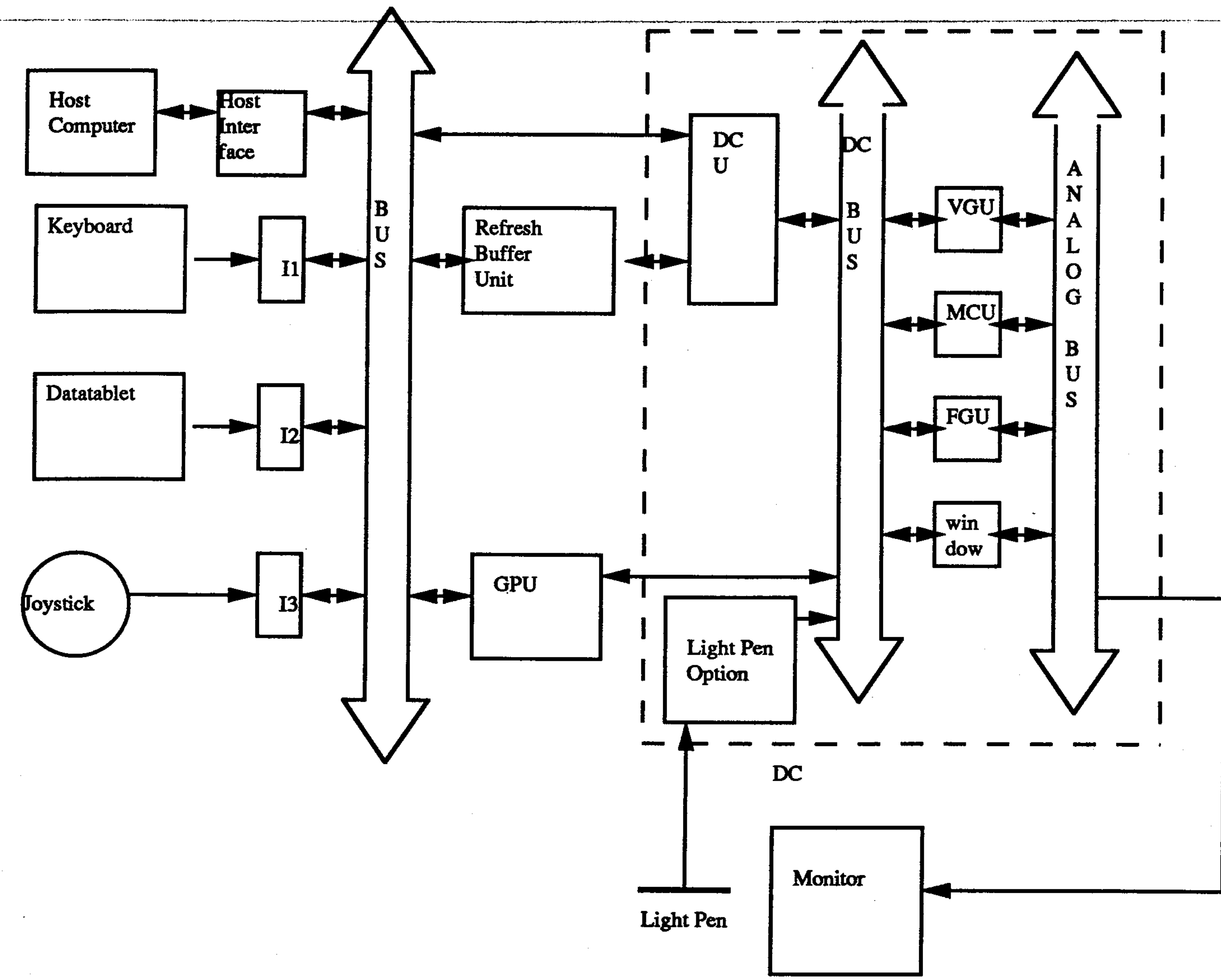
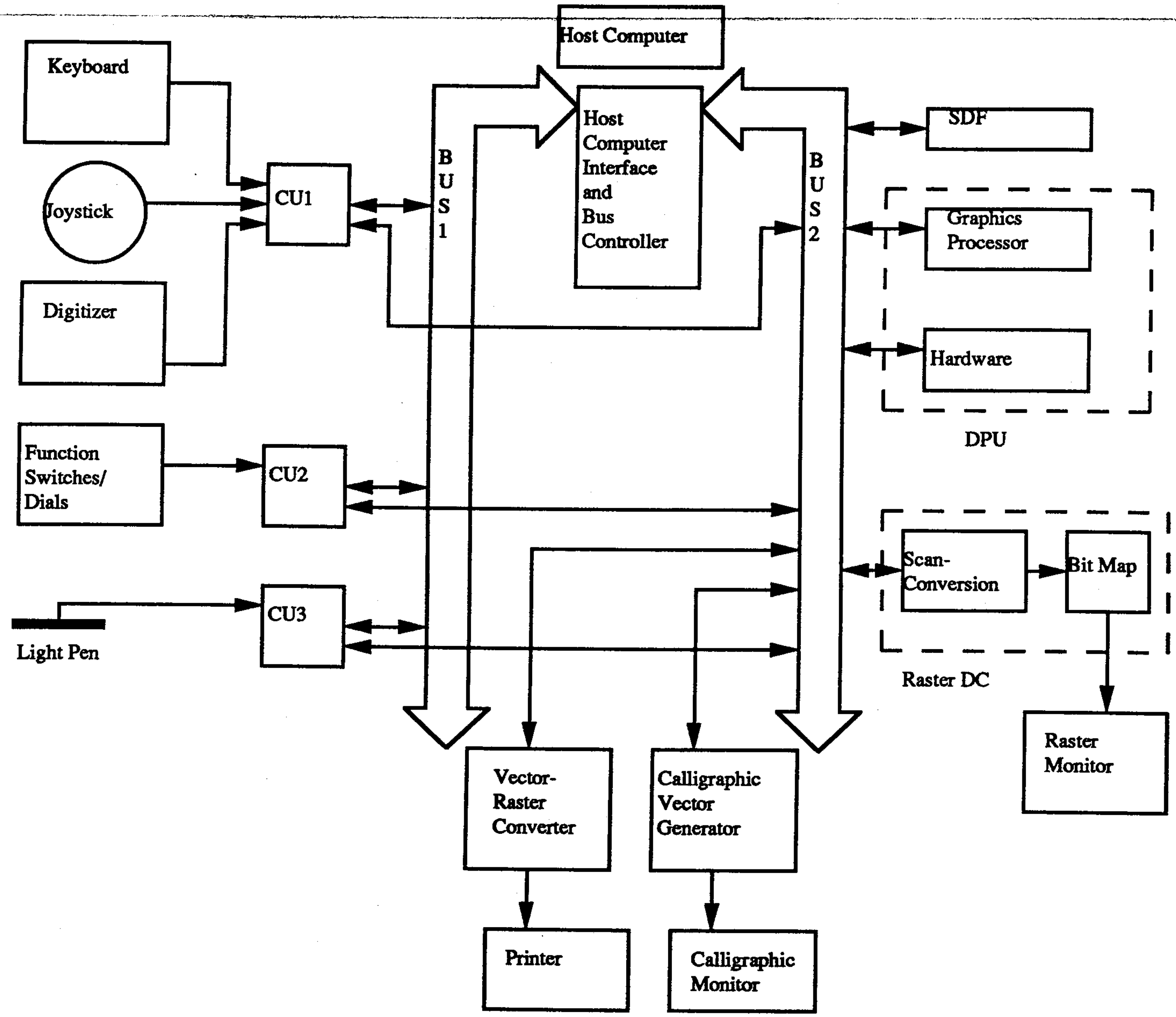


Figure 1.9: Display Architecture



Legend : - DC : Display Controller
- GPU : Graphic Processor Unit
- MCU :Monitor Control Unit
- VGU : Vector Generating Unit
- FGU : Font Generator Unit
- I1 : Digital Device Interface
- I2 : Data Tablet Interface
- I3 : Analog-to Digital Converter.

Figure 1.9.b : Display Architecture - A Simplified View of the Vector General 3400 Display Architecture.



Legend : - CU1 : Peripheral Control Unit
- CU2 : Function Switches Unit
- CU3 : Pick Module
- DC : Display Controller
- DPU : Display Processing Unit
- SDF : Structured Display File (contains the graphical descrition of an object.

Figure 1.9.c : Display Architecture - Megatek 7200 "Graphics Engine".

1.4.2 Applications Software

The applications software must handle the processing necessary for the display processing as well as for the support of the end user interaction. Figure 1,9 displays the overall architecture for the applications software. It is a layered approach to the development of the application software, the lowest layer interfacing with the operating system of the multimedia environment and at the highest layer, and end user interaction capability.

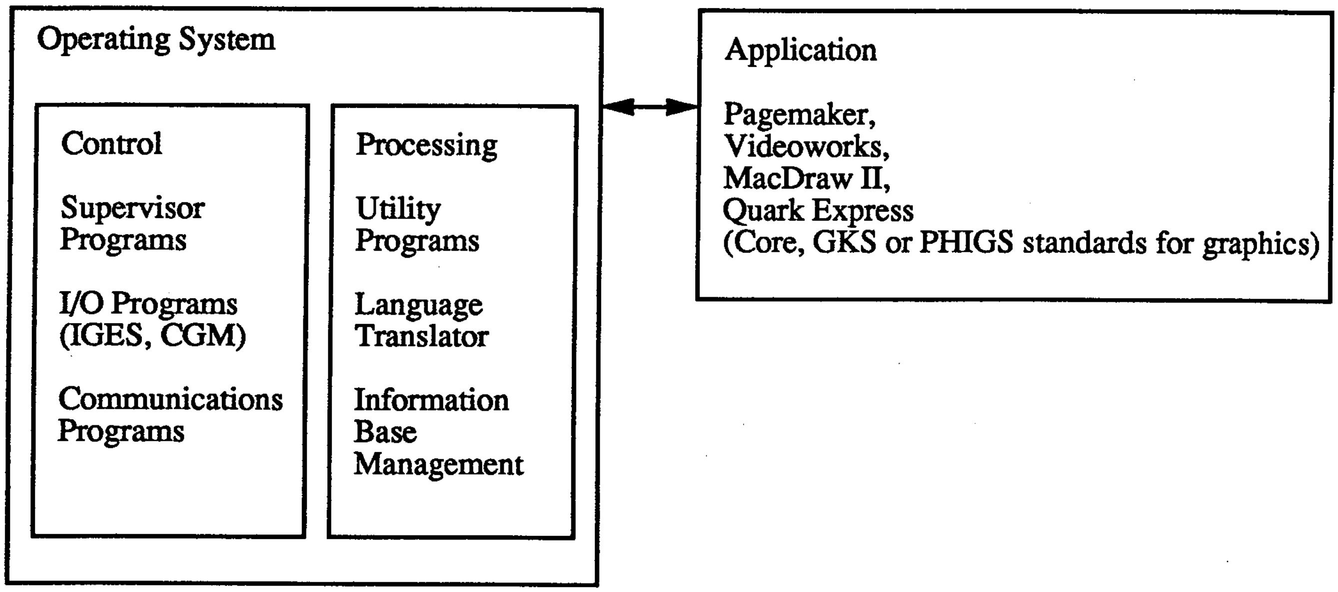
The software shown in Figure 1.9 is a canonical form for many types of early second generation multimedia systems. The layers include the following:

o User Interface Layer: This outer layer allows for support and integration of the end user into the system.

o Applications Support Layer: This layer provides for the interface to the User layer and utilization of the local work station for support of local processing.

o Data Management Layer: This layer provides for the support of the local data files and management of the images into the display. Such capability now in the X Window format can be found resident in this layer.

Figure 1.9 Software Architecture



Legend : - GKS : Graphical Kernel System
- PHIGS : Programmer's Hierarchical Interactive Graphics Standards.

Figure 1.10.a : Software Architecture - Software at Terminal for Multimedia.

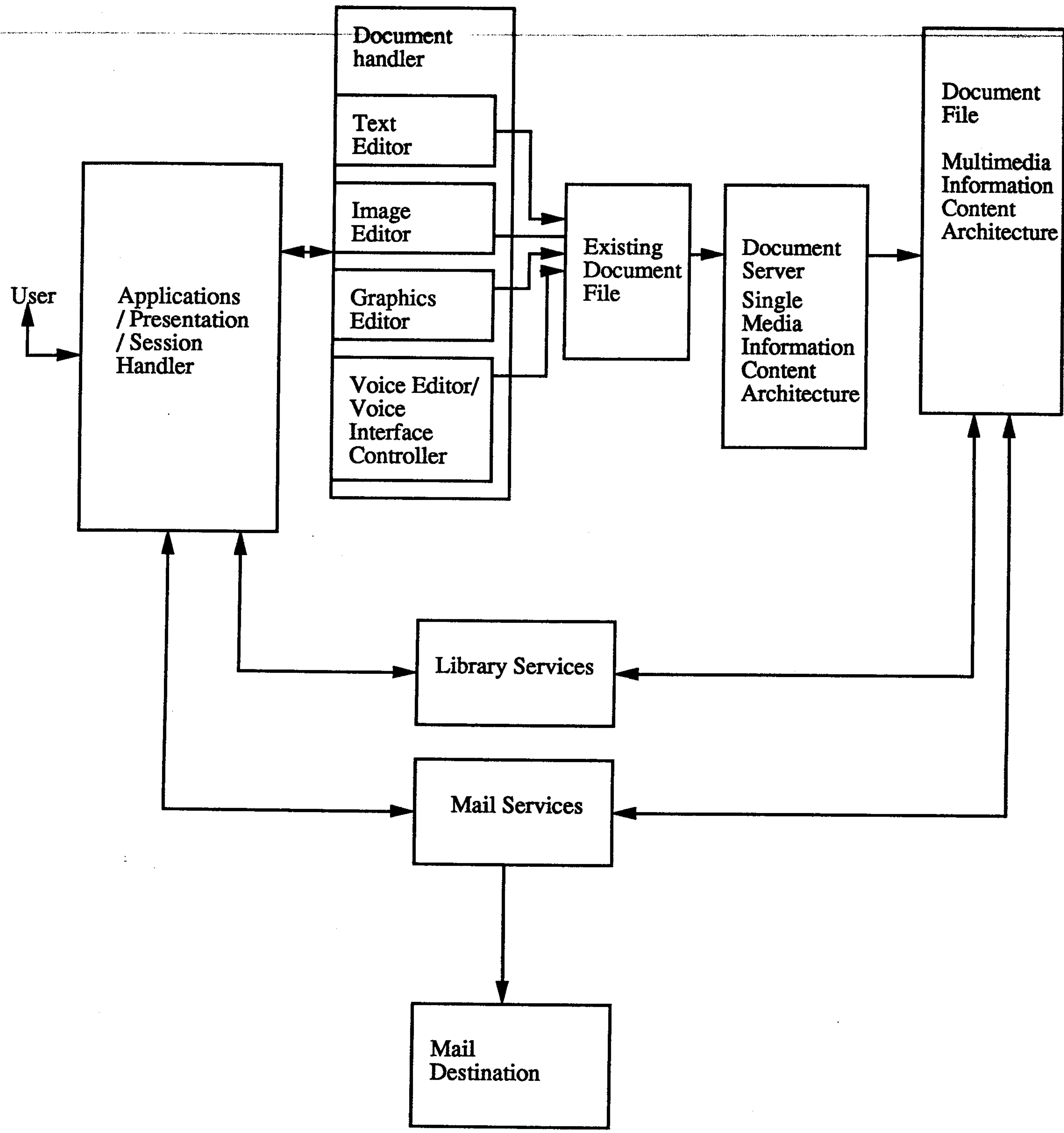


Figure 1.10.b : Software Architecture - Software Architecture for a Multimedia Messaging System.

1.4.3 Services

The services portion of the canonical design allow for the support of the multiuser environment and are the key to enabling the establishment of the session. Typical services may include:

o Session Establishment: This service allows for the generation and support of a session between any of the users. A single user may establish a session between themselves and any other virtual user available on the network.

o Mail

o File

o Directory

The services can be implemented in a layered architecture just as we had done with the applications layer. This architecture is shown in Figure 1.10. It consists of two layers, the shell and the kernel. The shall allows for the interaction between the individual terminals and the services and the kernel supports the interaction between the individual terminals and the network. The key to successful multimedia services is the ability to work in a fully distributed environment. We shall develop the distributed environment in latter chapters and shall also present a more detailed structure to these tow layers of the services.

Figure 1.10 Services Interfaces

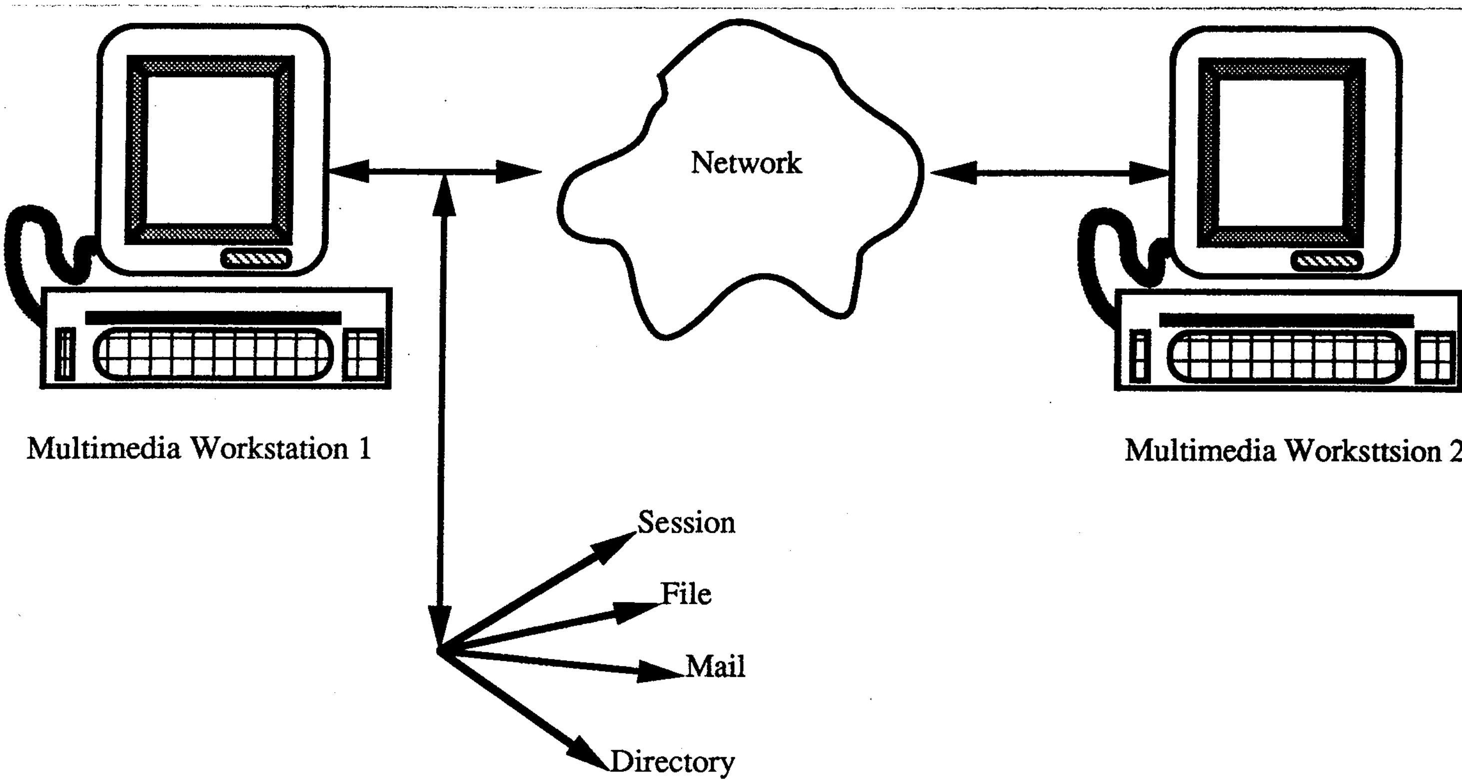


Figure 1.11: Services Interfaces

1.4.4 Network Software

The network that interconnects the end users must do so in support of all of the layered areas discussed above. The network must support the applications areas and furthermore support the services layers. The major difference between multimedia communications and computer communications is that often the computer needs to send the messages in what is called a connectionless format. Packet or datagram are sent as little letters with all the information to get from point A to B. In the multimedia environment, the need for the use of sessions requires that the sessions be supported by a connection based network service. In addition the network must be capable of providing large amount of bandwidth on demand to the user. Consider the case of the second generation high resolution display with 100 M bits of data. To transmit one image requires a 100 Mbps channel from any point to any one of a set of other point. The response time of less that one second is critical to the succes of any multimedia system.

The network architecture, in both hardware and software is shown in Figure 1.11. This architecture can be either distributed or centralized in nature and the network software must be intelligent enough to support the session based connection path. We shall be developing this concept in further detail in the remainder of the book.

Figure 1.11 Network Architecture

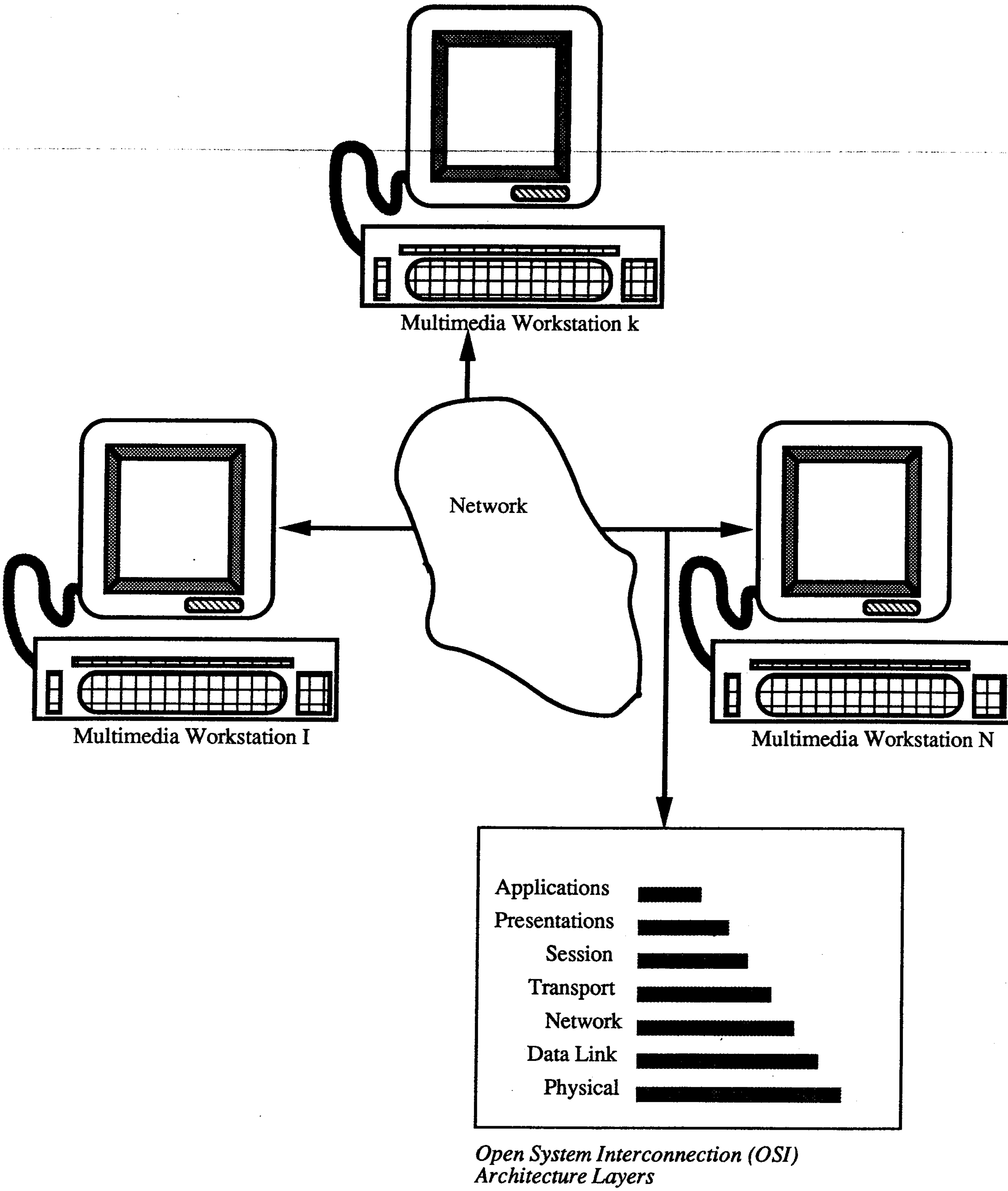


Figure 1.12: Network Architecture

1.4.5 Storage

Multimedia communications requires large amounts of complex image information for use by many users all at the same time. No matter how the cost of memory decreases, the key bit of information will not be sent everywhere at the time it is generated. Thus the network must provide access to memory that is distributed and that also can be accessed in extremely short periods of time. Consider again the example of the high resolution display that requires 100 M bits of image. The transport speed may be 100Mbps to 1Gbps but the bottleneck may be in the finding of the imaging, accessing it and loading it onto the network.

Figure 1.12 shows several of the storage alternatives available to be used in a multimedia environment.

Figure 1.12 Storage Alternatives

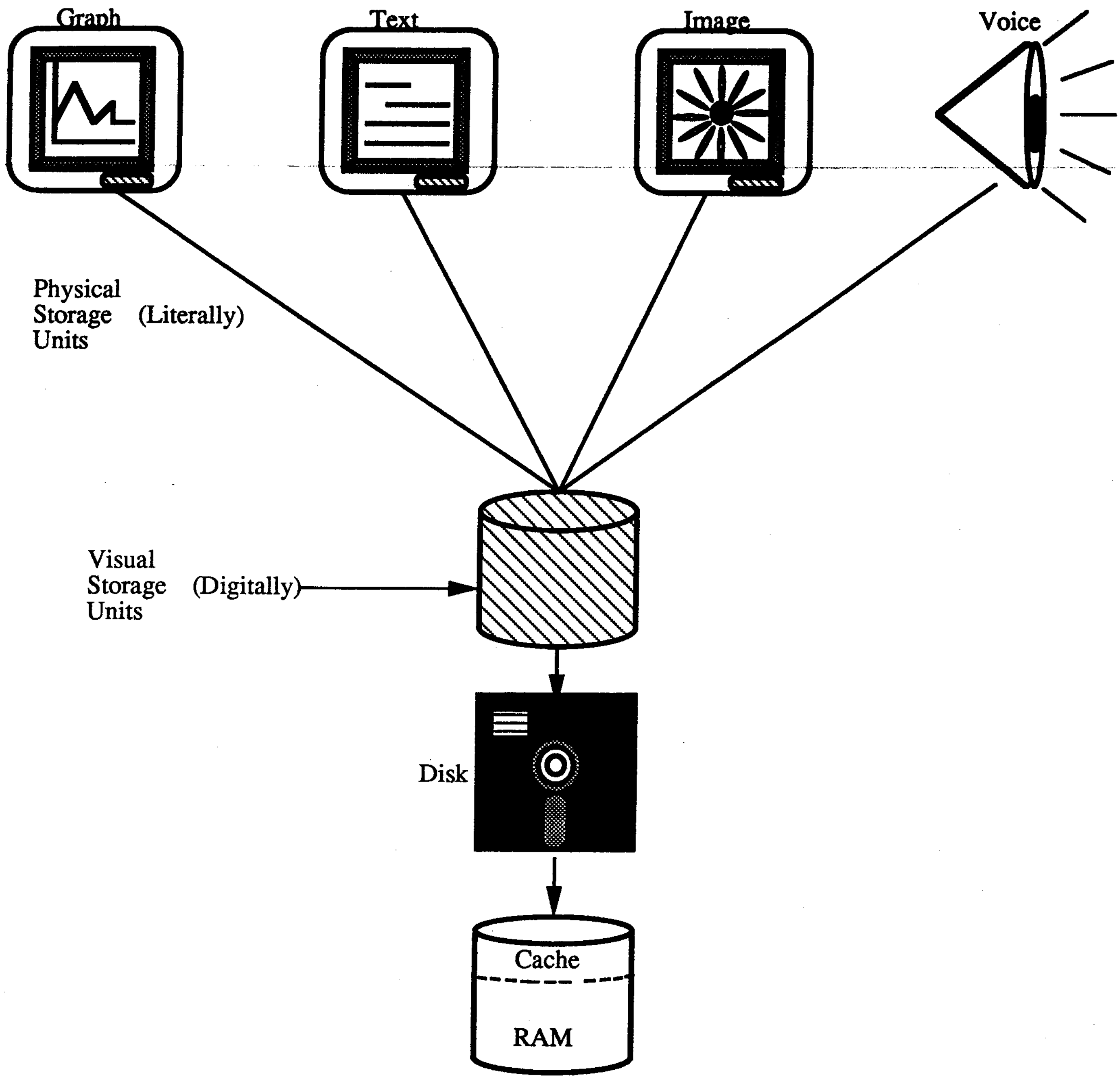


Figure 1.13: Storage Alternatives

1.5 Key Problems

One of the worlds greatest philosophers, Ludwig Wittgenstein, once stated that the essence of true knowledge of any field was the ability to pose the correct set of questions, whose answers are simply stated. Thus with multimedia communications, the development of a new body of understanding is best based upon the positing of the proper set of questions and these questions are in turn based upon a set of key problems that we face in attempting to communicate in this new environment. The following problems and the ensuing questions represent the body of the text.

1.5.1 Problem 1: Characterization

As we have developed in the early part of this chapter, the essence of multimedia communications depends on the concept of the image. The image may vary in form from a high resolution picture to a speech conversation. The first problem is the characterization of the physical image into an electrical form. For example, a speech conversation that varies over a finite period of time may be digitized in a standard form by sampling at the rate of once every 125 msec and using eight bits per sample. This will yield the standard 64 Kbps voice sample of speech.

As a second example of the characterizations problem is that applied to high resolution pictures. Again we can consider sampling the image in two dimensions and in storing the image in this highly sample digital form. The problem is again how well to sample to retain the integrity of the basic image.

Another element of the characterization problem relates to how to characterize an image so that it may be manipulated and not just stored. Consider the following example.

Example: In the medical field, there is significant use of both CAT (Computer Aided Tomography) and MRI (Magnetic Resonance Imaging) imaging. CAT scans use X rays to resolve the different parts of the body tissues based upon the density of the tissue and its ability to absorb the X radiation. By passing many different X rays through the body the image can be reconstructed to reproduce the innermost portions of the body. In MRI, the technique uses the ability of different tissues to resonate when under strong magnetic fields, the resonant frequencies dependent upon the tissue type. In the MRI case a multi dimensional Fourier transform creates images of the human body. Now the physician desires to compare the results of the two scans to determine the status of the cerebellum, the posterior portion of the brain. The characterization problem in this case relates to how do we define the cerebellum as an abstraction in both imaging schemes. We wish to do so not by just comparing bit mapped displays of the same tissue but to do so by a full element abstraction.

1.5.2 Problem 2: Utilization

The utilization problem takes the image characterization and addresses the issue of the human interface. How does the human specify the specific image, object or element, and in turn how does the human user interact with the system to allow for the manipulation of the images. The concepts of the touch screen and the mouse are but simple starts to the development of agents in this area. The general concept of an agent, that is an entity that can abstract both and element and an action are key to understanding the issue of utilization.

Figure 1.14 Utilization Application

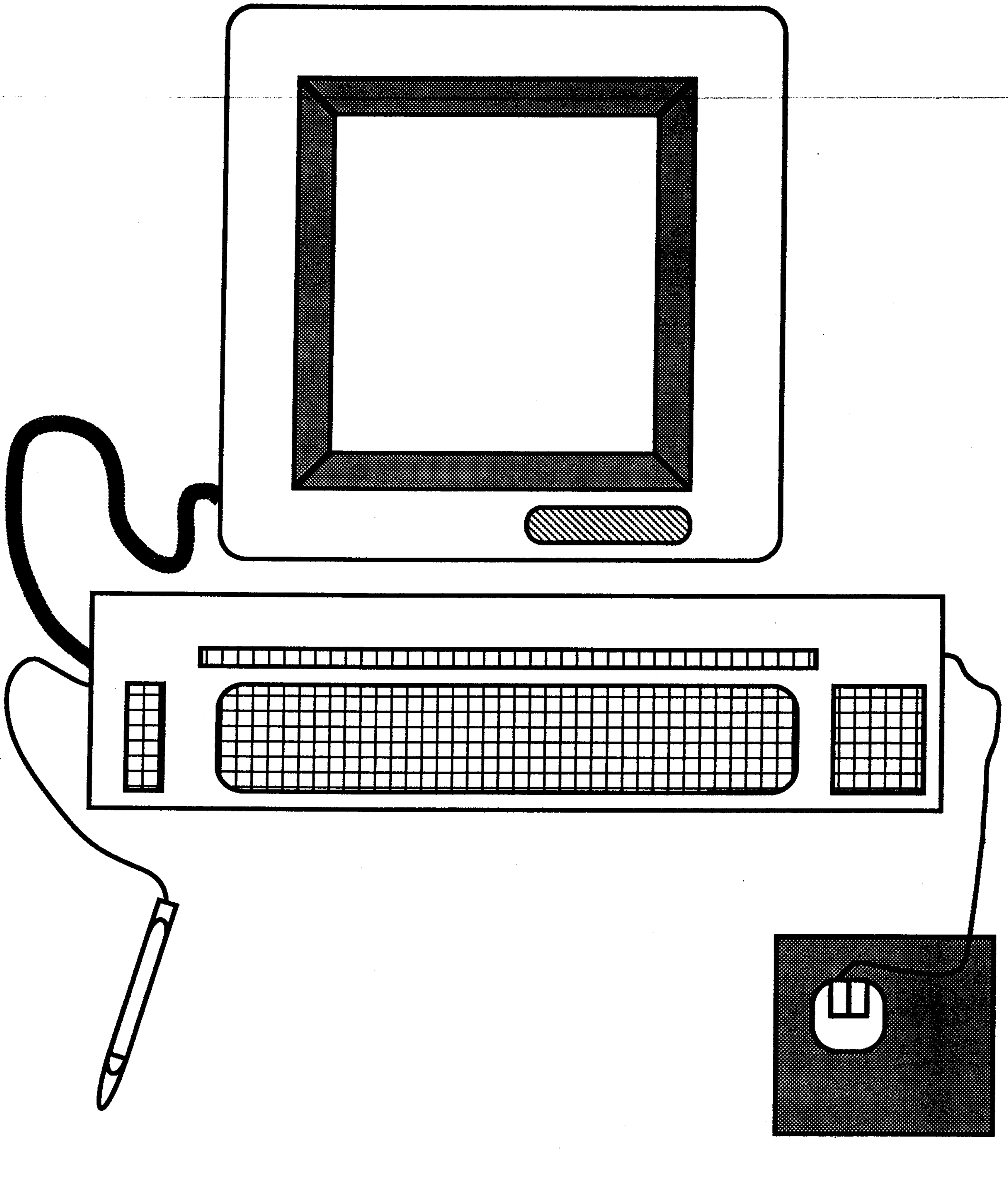


Figure 1.15: Utilization Application

1.5.3 Problem 3: Transmission

Transmission of the information relates to the problem of taking the users view of the world and allow for it to be effective in several places at the same time. In contrast to sharing, which we shall describe in the next section, transmission relates only to the transport for the images in a session context. The present means of transmission are generally quite limited. They use existing telecommunications services that in most cases are not designed for high resolution image transport or for the use of extensive session sharing. The telecommunications transmission channels allow for rates up to 45 Mbps which limit the use to small image sizes or lower refresh rates.

For example, consider a 2,000 by 2,000 pixel array using 24 bits per pixel. This amounts to about 100M bits per image. If such an image were to be used in a full motion video context at 30 times a sec refresh rate, then we would need 3 Gbps transmit capacity. That is almost 100 times greater than the current offered tariffs for transmission services. Thus the transmission problem focuses on two issues, the first is developing higher data rate transmission channels and the second is the developments of compression techniques for the reduction of the data required to transport the true information content.

The true transmission dilemma that faces many users is to wait for the greater data rates or to incorporate the use of often high cost compression technology. The date rates that are now being implemented are moving to the 500Mbps range for local access and to 1.5 Gbps for long distance transport. The problems still exist that one cannot share the data links and thus one is compelled to use the data channels as if they were fully loaded.

Figure 1.15 Transmission Example

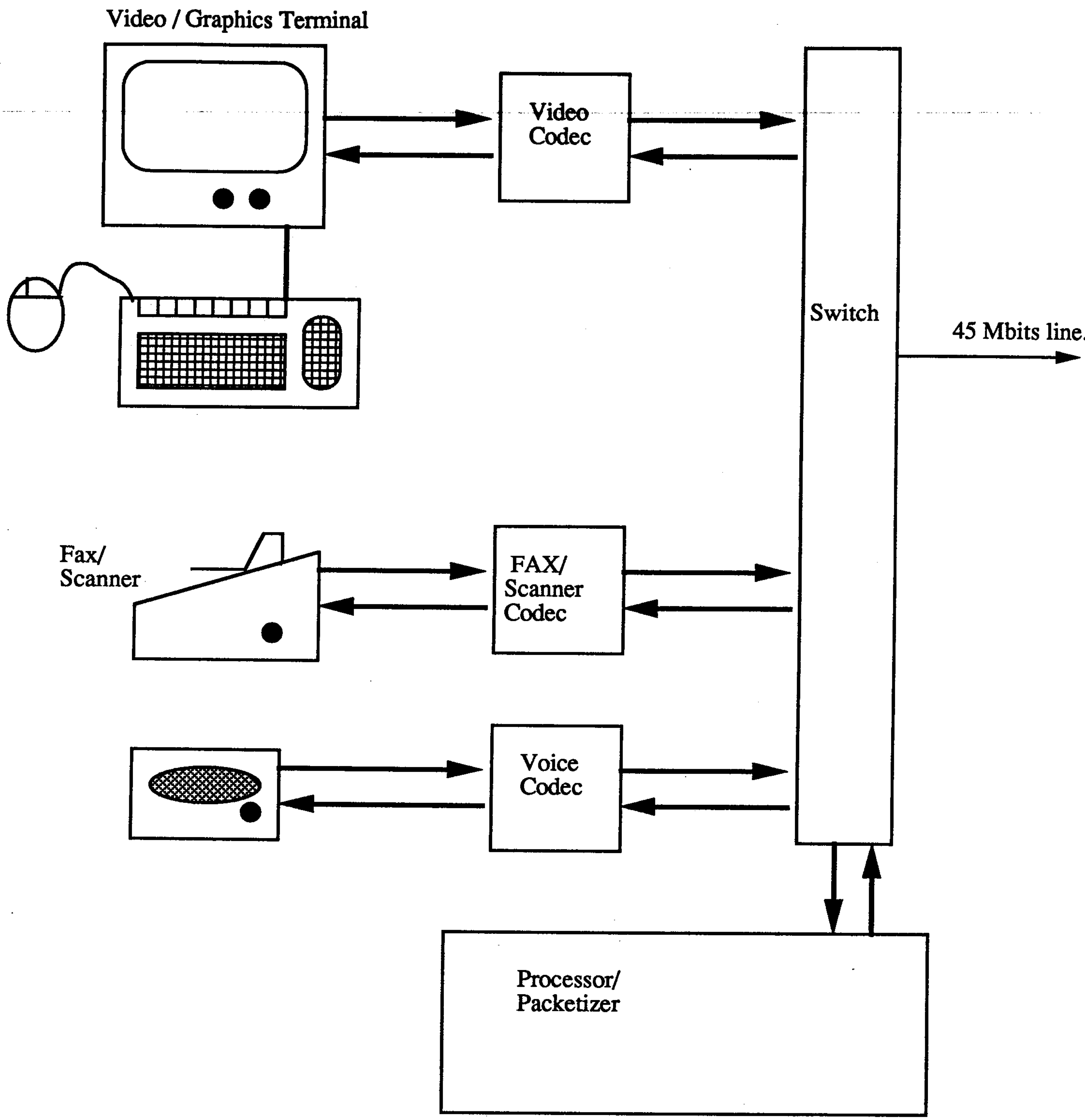


Figure 1.16 : Multimedia Transmission Example.

1.5.4 Problem 4: Sharing

The sharing problem relates to how best to take multiple users in different modalities of use and to allow them o share the communications and end users services in a user friendly fashion. The essence of sharing is essentially the development of the session concept that we have developed. As we have stated, the session concept was developed to support the virtual user layer of our multimedia architecture and allows for the interface and support at that level. The applications layer interface will be shown to have the session capability already present in what is called the OSI Layer 5 Session protocols.

The com,plexity of the sharing problem or that of VU Sessioning relates to the need for connecting diverse end user devices together with diverse end user interfaces. If there were standards that existed for doing so the problem would be generally easy. Such standards exist for the applications to applications interface but fail to exist for the VU to VU interfaces. We spend a great deal of time in this book focusing on the issue of sharing.

Example: Consider the medical application again wherein the user has to work in a complex environment of CAT, MRI and Nuclear medicine scans. In Figure 1.x we show the many devices in this network and show the many interfaces that are available.

Figure 1.16 Sharing Concept in Medical Environment

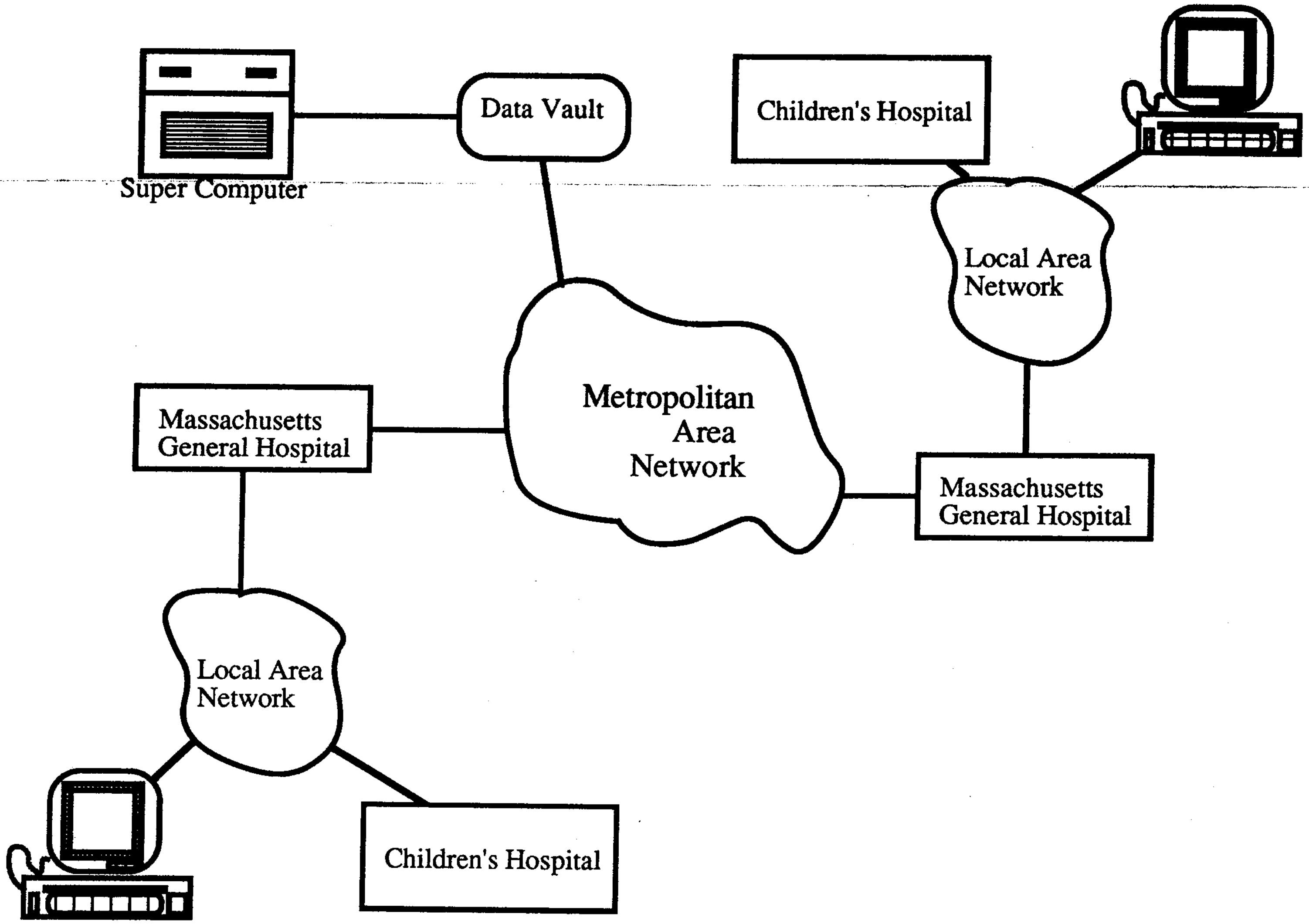


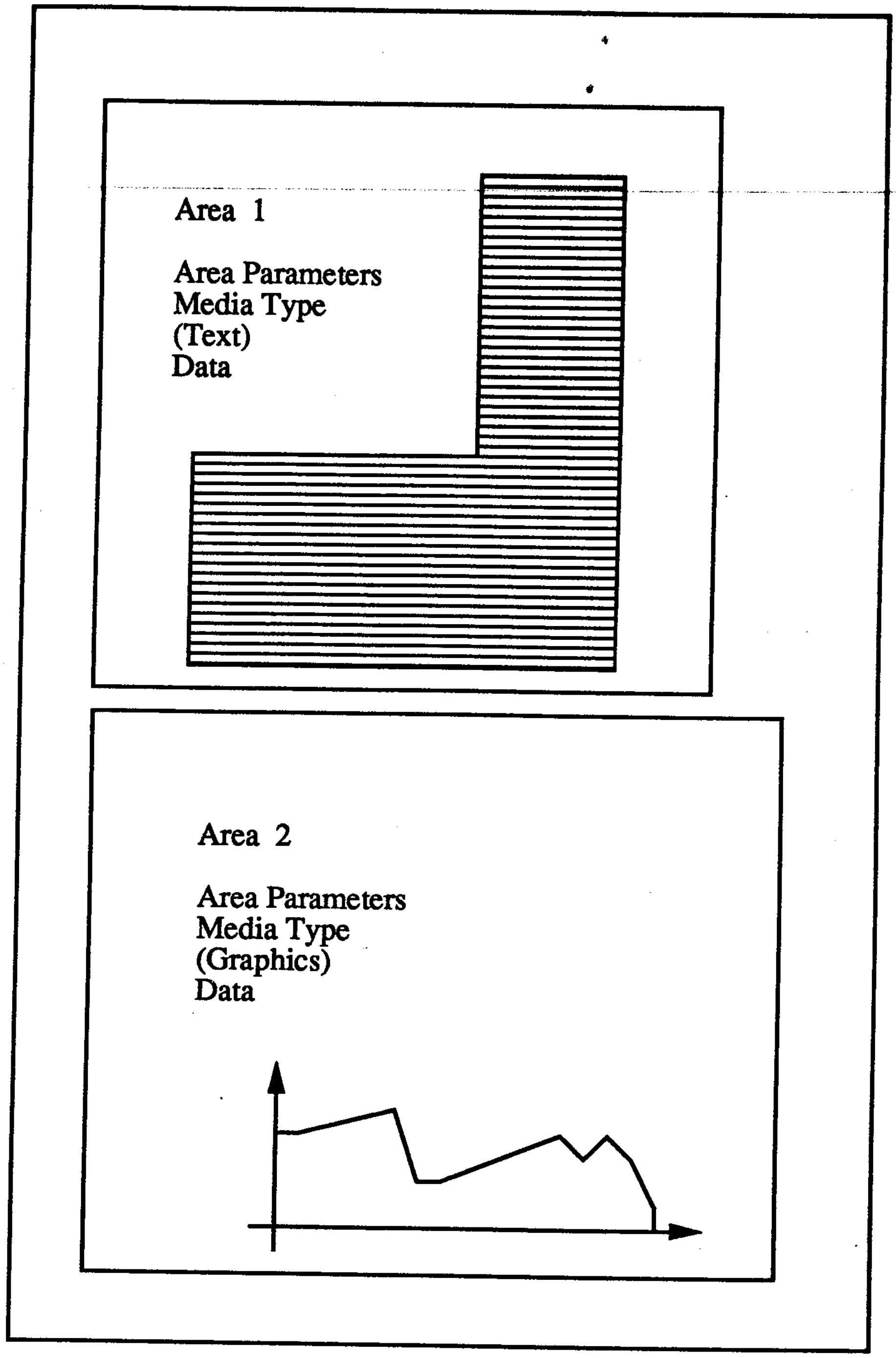
Figure 1.17:Sharing Concept in Medical Enviornment

1.5.5 Problem 5: Mixing

The applications to multi user systems is characterized in the sharing problem but the multi media multi user system problem is best characterized by the mixing problem. The mixing prob;lem is the one that occurs when different forms of images and media are to be shared and manipulated by multiple users at the same time. These users may have different types of equipment, have their images on different types of databases and different interfaces.

Example:

Figure 1.17 Mixing Concept in Printing Environment



Area Parameters : physical location of the area

Media Type : characterization of the type of information

Data : binary description of the segment of the document.

Figure 1.18.b : Mixing Concept in Printing Environment -Mixing in Multimedia Documents Using an Area Control Mixing Type.

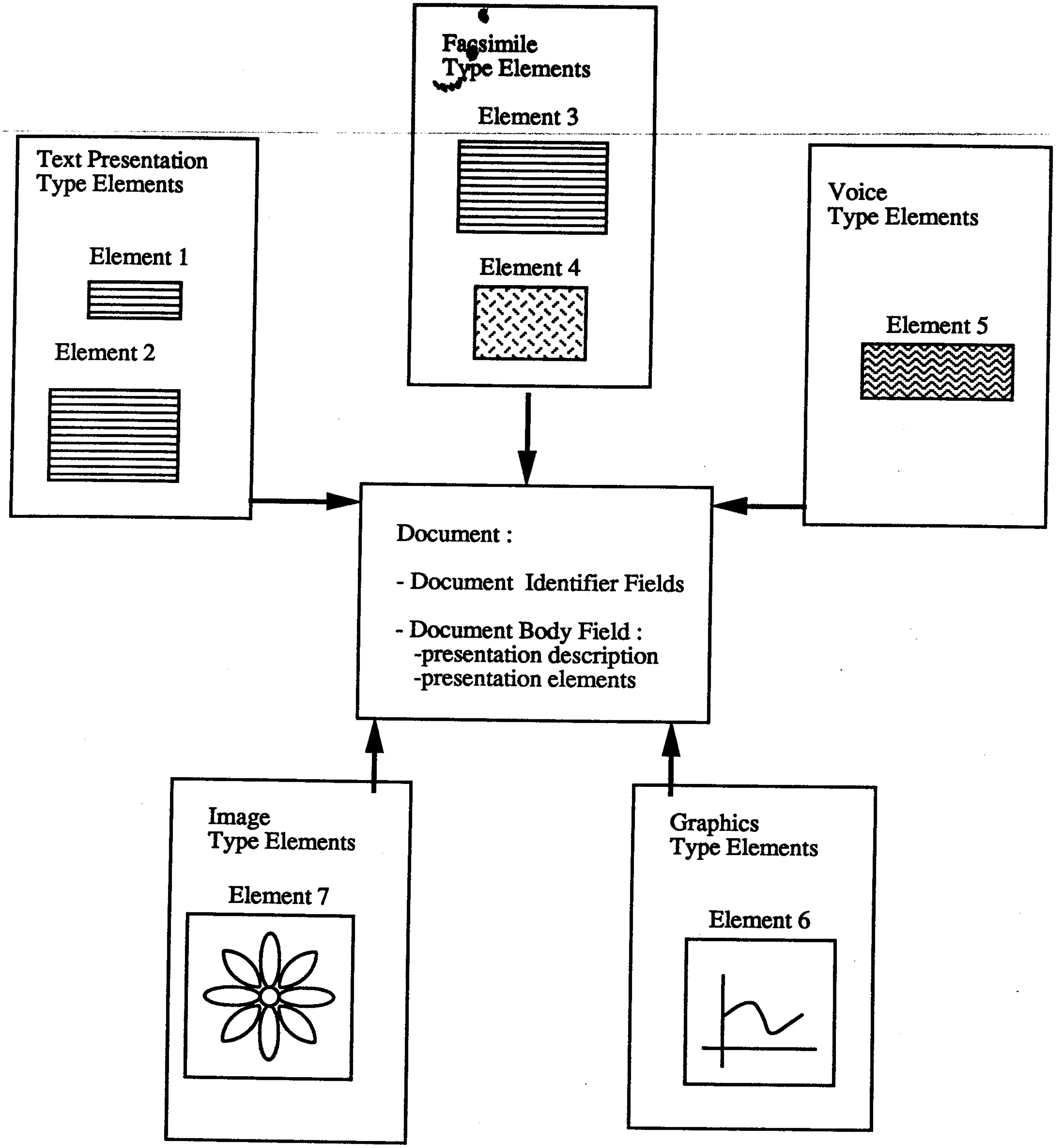


Figure 1.18.c: Mixing Concept in Printing Environment - Mixing in Multimedia Documents Using the DARPA Multimedia Mail Content Protocol.

1.6 Outline of Book

The book follows the discussion on the key problems that frame the multimedia multi user communications area. In Chapter 2 we spend a great deal of time discussing the characteristics of many forms of multimedia images. Specifically we discuss the way in which users of still images, graphics and video, for example characterize their media. This is important since when we take these existing media that are in a physical and non electronic form and turn them into electronic, we must do so so as to preserve their original frame of reference. For example, if we develop a system to display radiological studies from file to a digitized form, then we must preserve the same image density, intensity and responsivity as does the film radiograph.

Chapter 3 discusses the many types of multi media interfaces. These interfaces range from the ways in which we transform the physical media of the image into an electronic image to the ways in which we characterize images in abstraction forms. As we have developed in this chapter, abstraction of images is one of our major concerns. The approach to abstraction may be structural or connectionist. Our major focus in this text is that of a structural approach, leaving the connectionist to those other works that cover them more fully.

The issue of storage of multi media information is contained in Chapter 4. In this case we are concentrating on multimedia databases that are significantly more complex than those found in normal data processing environments. The multimedia database problem revolves around the issue of the scale of the image and its need for integrity and simultaneity. Another element of the multimedia communications that this chapter considers is that of performance and sizing. These latter two items are of critical importance in designing and developing any new multimedia system.

Chapter 5 addresses the issues of communications. We focus on the standard approach to communications by working with the OSI layers. Our approach is different than most others by starting with the higher layers, viz. applications and presentation and session, and working downward. The issue here is that instead of looking at the physical communications link as transport and then finding what is missing, we focus on the session and the essence of the multimedia environment and works downwards. This approach provides new insight into the communications link. We also develop an understanding of the systems and issues associated with broadband communications systems.

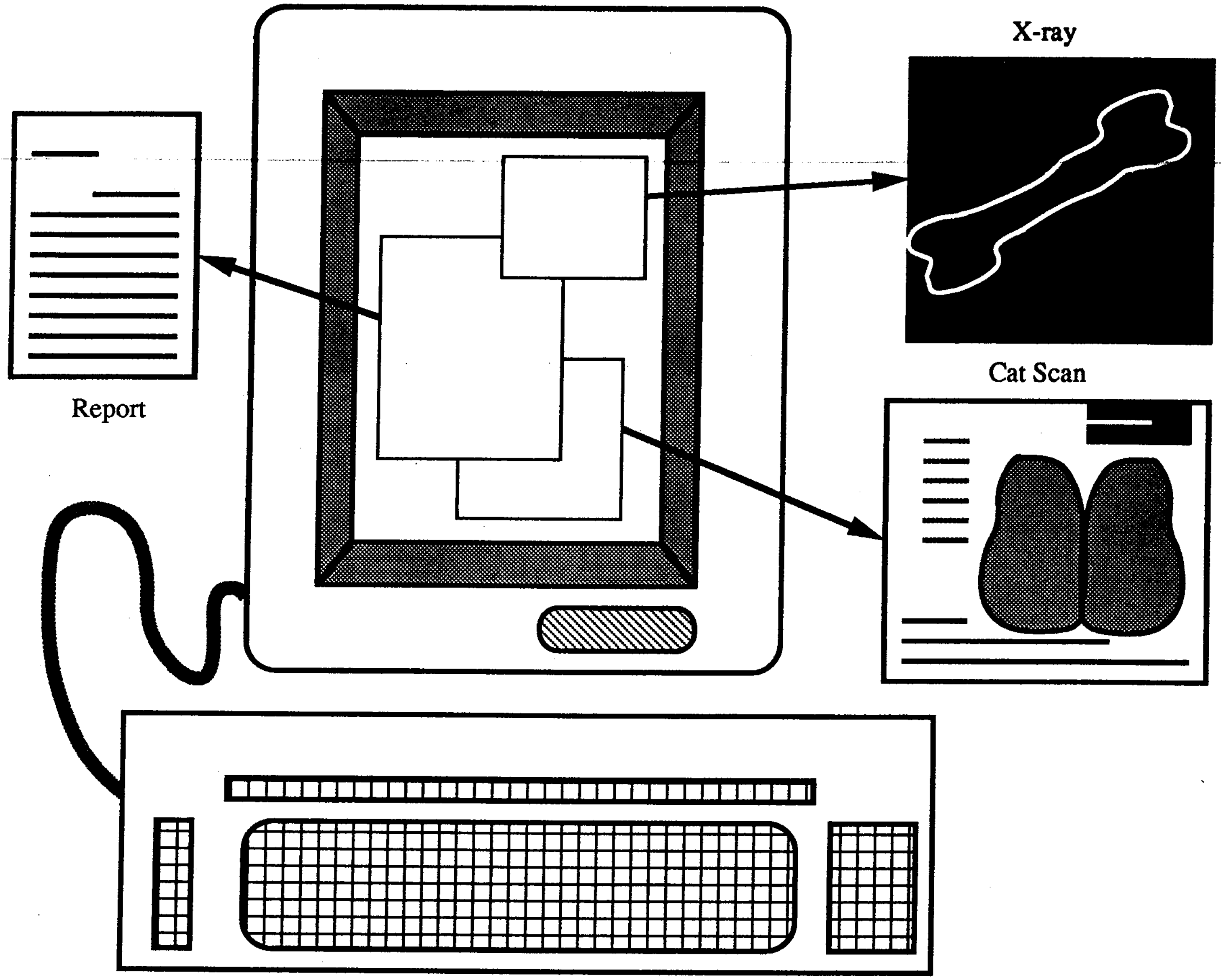
The essence of any multimedia system is the development of an overall architecture. The first few chapters focus on the major problems that we have developed earlier in this chapter. Chapter 6 develops the overall architecture for such systems. In particular we focus on developing a methodology for architecture development, including the issues we have developed on characterization, processing, storage and communications.

Chapter 7 develops the theory of multi user systems, showing how to extend the elements of the multimedia environment to that of the many user system environment. We spend a great deal of time discussing the performance and sizing efforts of such system designs.

Chapter 8 develops in details many of the issues associated with the implementation of distributed systems. In the multi user multi media environment, we are dealing with a fully distributed environment. In this environment we must focus on distributed databases and distributed operating system. As we have discussed the multimedia database we must now focus on how it can be expanded to a distributed environment. The issues of distributed operating systems allows for a better understanding on how multi users may now deal with the issues of sessions in a shared synchronous environment. One of the major issues is that of understanding the overall synergies of a shared network.

Finally Chapter 9 presents the overall conclusions and the directions some of the multimedia system are taking in today's environment. In particular, we discuss many of the issues that will become research issues in the years to come.

Figure 1.13 Medical Application



Fi gure 1.14: Medical Application

List of figures.

Chapter 1.

(1) More Detail on Displays,
(2) Details on RGB Guns & Displays

CHAPTER 2

Multimedia Characteristics

This chapter discusses the different media elements that encompass the multimedia domain. Specifically we develop the detailed characterizations of the various media types, developing an understanding of the issues of how they map into communications and reproduction formats. One of the issues discussed is the manner in which the media are presently presented and the transition from what is frequently a mechanical process to and electronic process.

The two issues that we focus on in this chapter are those relating to the characterization of the overall source in a multimedia environment. Those two issue are the characterization and the specific sizing of the multimedia image in terms of bits. We know that the bit sizing is essential for the source characterization and the we shall latter be using it as an integral element of the system sizing and performance.

2.1 The Media Construct

The main issue discussed in this chapter is how we represent the reality of the observed world in a form that can be stored permanently and processed at latter times. A simple example is how we take a picture of a particular person or place. The picture is merely a representation of the person or place and in many ways does not duplicate all the features of the person or place. We may then take that film medium and digitize it for storage in a random access memory to be post processed to either extract features or enhance the image. We can then readily ask the question of, "How well does the digitized image reflect reality?".

Thus the key element in understanding the use of media for the processing of information relates first to how we store the information on the different media and second how well the stored representation relates to the original reality. Thus media characterization starts with the following paradigm:

o Reality exists and does so independently of the observer. The reality has an extent that can be projected onto an observer through many means. The representation of the abstract reality is unique at any one instant and this uniqueness can be represented onto each of the observers senses in a definable and measurable fashion.

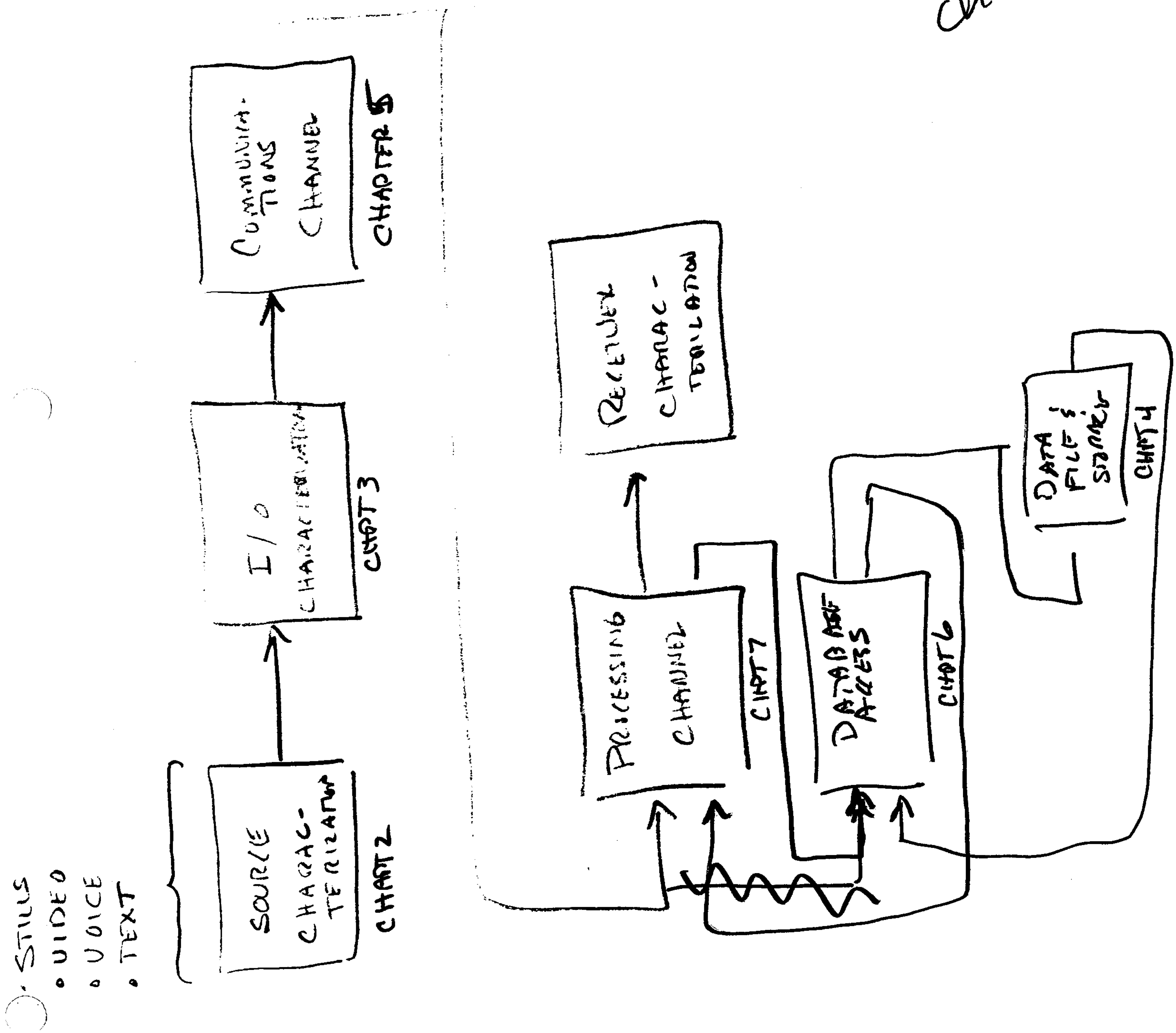
o Senses are the major inputs to the human for the purpose of transferring information about reality. The senses common to the human are sight, sound, touch, smell, and taste.

o Observations are the results of reality projecting itself onto an observer through their senses. For example, there may be a representation of a reality in terms of the electromagnetic waves that is generates and there may be a model for the optical response of the human. The combination of this quantifiable reality, with this observation system can be further mixed with

In Figure 2.1 we depict this paradigm of the world that we have viewed with an exposition of the example that we have just discussed. In this chapter we shall assume that we deal only with the senses of sight and sound and that we shall discuss the different media that can be used to store the information in a specific form of representation. We further are concerned with the accuracy of that representation adequately matching the information presented from the reality. It should be noted that we nowhere deal with relating ourselves to the ultimate reality. This is a very Platonic approach to understanding our existence but the reader should not read any significant philosophical positions in this paradigm of a world view.

Figure 2.1 Paradigm for Multimedia Representations

Move this up to Chpt. 1 and Use in each Chapter



o Medium Characterization Effectiveness is a measure of the accuracy in which the representation matches the information. Since both of these are quantifiable we can analytically measure the effectiveness of the medium in term of the ability to reconstitute the original information bearing signals.

Thus the two major efforts in this chapter are the ability to understand how information is reduced to storage on different media and in turn to quantitatively determine the effectiveness of that medium in retaining a close proximity to the original information.

Consider the model shown in Figure 2.2 of a reality of a tree and the reduction of that reality through the process that we have just discussed. We define the information of this reality as the electromagnetic wavefront generated by the reality, projected along the visual senses. Let this information quantity be given by:

(2.1) $E(x,y,z,t)$

Figure 2.2 Image Reconstitution and Performance measures

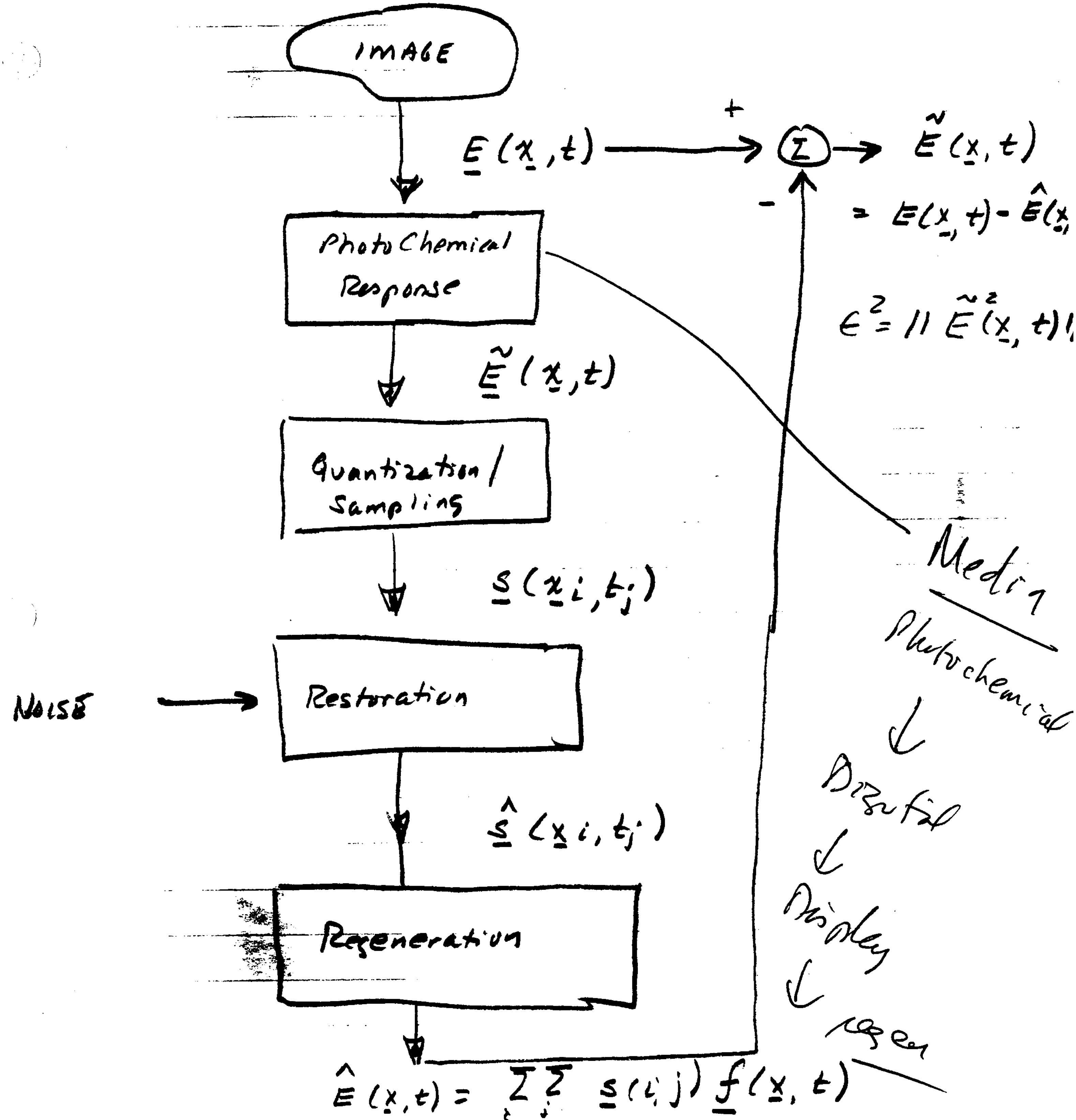


Figure 2.x Image Quality and Resolution
(Processing, Sampling and Digitization)

We now take a picture of this reality and generate a photoemulsion that can also generate an observation or an information based image of its own. This information image can also be represented by an electromagnetic field:

(2.2) E1(x,y,z,t)

Now it is possible to measure the error induced between the emulsive image capture medium and the original electromagnetic information image. We definie a measure ||E-E1|| as some metric on a three dimensional space (see McGarty) that allows for the specification of errors. Thus it is quantitatively possible to measure the effectiveness of medium characterization.

Now we can further take the photographic image an digitize it. To digitize an image we must perform several tasks. These are typically:

o Time sample the wavefront at t(k) seconds, where k+1 to N.

o For a single sampled wavefront, sample the wavefront in space at point xi and yj. Call the samples E(i,j,k).

o Now quantize the samples of E(i,j,k) into quantized values called S(i,j,k), such that the quantization is close to the original signal.

o Now let us assume that we are sampling the picture at the rate of thirty times a second, and that we have 1 thousand by one thousand element display, that is 1 million pixels, and that we quantize in one of 1024 levels (e.g. 10 bits) so that we are generating 300 million bits per second of data on this storage method.

Now let use move one step further in Figure 2.2. We have now stored the data digitally and we wish to retrieve it. To do so we need a display that reproduces the image using the stored data files. Let us assume that the display is 512 by 512 in dimension and can respond to only 8 bits of phosphor response. The image generated and transmitted towards the human eye is the electromagnetic field E2(x,y,t). We can now describe how the system can effectively reproduce the image by the error:

$$(2.3) \quad ||E-E2|| = ||E-G(E1)|| = ||E-G(F(E))||$$

where G and F represent the transformations of the two media storage methods.

This simple example of using three different media raises the two points that we wish to present in this chapter. First, we must be able to characterize how primary reality is stored and transformed by the storage media, and second what the effect of this transformation is on the overall restoration of reality.

Media characterization starts with a specific reality. That reality, for our purposes, is delimited to the visual and oral areas. Specifically it is related to what we see and hear. Expanding the reality to the other three senses (smell, touch, and taste) have not been focused on in the present are of media communications, however, they surely will evolve as we come to better understand how to transfer information through these senses.

There are multiple types of information bearing media that are used. Several of the more common forms are still images, full motion video, and voice. In this section, we develop models for the characterization of these three types of media and provide a basis for the development of generalizations for many other types of media.

2.2 Images

The still image is the most common form of display format for Pictorial information. The still image generally tires to capture some form of reality that is represented, for example by a scene in nature. That scene in nature may in this simplest form be a view of a tree on a hill with the clouds in the background. We can see in Figure 2.3 a that this 115 year old picture has s significant degree of clarity and compares well with many of today's best cameras.

The first question is how do we take this complex still image and store it in a finite amount of memory for latter retrieval and display. That problem had been answered over a hundred years ago with the development of the photograph, a physical/chemical storage medium. The photograph is only an approximation of the true reality however. It uses a form of silver salts that are responsive to light falling on them and darken to different degrees. The resolution of such images is that of the underlying silver nitrate solution and it generally is better than that of the human eye. Thus the viewer of one hundred year old photographs will not see any degradation of the image but will perceive a clear and well preserved representation of the scene. The key observation of this fact is that the eye has a key role in determining an acceptable degree of image resolution and with such forms of mechanical/chemical memories as 35 mm film, there is more than adequate resolution.

With the use of computers however and the needs of other media such as the newspapers, there are lesser degrees of resolution required in image display. This type of resolution is what many of use see in the print out of our computer terminals or in the daily newspaper. In the case of the computer printer the image is a composition of small black dots, the density of the dot proportional to the intensity of the image. We can see from Figure 1 b that this image is generally unacceptable.

2.2.1 Means of Characterization

We can now proceed to discuss the details of the characterization of the different forms of images. We shall tacks as the starting point, a reality of a fixed scene and from that scene develop the ways in which we first characterize achromatic (eg black and white) and then chromatic (eg color) still images.

Before beginning with the detailed discussions of the two major types of images; achromatic and chromatic, let us first present a brief discussion of the way the human eye responds to images. In Figure 2.x, we present the human eye in some detail.

Figure 2.3 Human Eye Anatomy

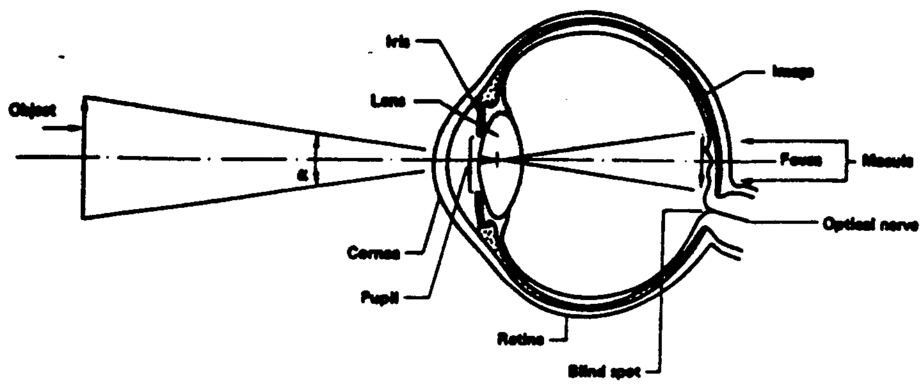


Fig. 7. Schematic view of the eye.

Fig. 8. Pathways in the human visual system.

vision. This imperfection is the source of the spherical aberration which appears as a blur in the focal plane. Such a blur can be modeled as a two-dimensional low-pass filter. The pupil's diameter varies between 2 and 9 mm. This aperture can also be modeled as a low-pass filter. The highest cutoff frequency corresponds to 2 mm. Continuous enlargement of the pupil's diameter decreases the cutoff frequency.

C. The Retina

The retina is the neurosensorial layer of the eye and its area is about 12.5 cm^2. It transforms the incoming light into electrical signals that are transmitted to the visual cortex [redacted]

The anatomy of the retina shows five types of cells organized in layers as schematized in Fig. 9. Notice here an illustration of the general wiring principles of Fig. 6. The furthest layer from the incoming light is that of photocells. There are two types of photocells: rods and cones. A normal eye contains about 130 million rods and 6.5 million cones. Rods and cones are different enough to be examined separately. Rods are sensitive to shapes and need low luminance (scotopic vision). In contrast, cones need daylight (photopic vision). They detect color and distinguish details.

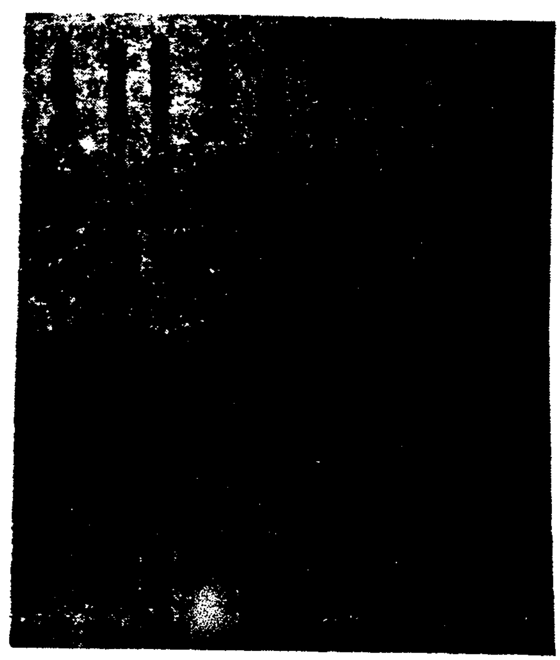


Fig. 9. Cells of the retina (after [29]).

Their distribution in the retina is highest in the vicinity of the optical axis of the eye. That is why a precise detail vision is obtained only when the eye "fix" them, in other words, when their image is formed at the fovea. In this region there are about 120 cones per degree which fix the visual resolution to 1 min of arc.

Photoreceptors are responsible for transforming the in-

The eye is composed of the following elements:

o Cornea: This is the outer surface of the eye and generally encompasses the overall outer surface of the organ.

o Iris: This is a circular and pigmented membrane behind the cornea that allows for the entry of light. It opens and closes in one of its parts, the pupil, upon response to the total intensity of the light incident on the eye.

o Pupil: This is the specific opening at the center of the iris that responds to the light level.

o Lens: This is element of the eye and is adjusted by muscle movement to allow for the positioning of images on the rear part of the eye.

o Retina: This is the most sensitive part of the eye. It contains the sets of nerves that allow for the reception of the light and its transmittal to the brain for processing.

o Optic Nerve: This is the connection point between the eye and the never pathway to the brain.

Within the retina are two major types of nerves cells; rods and cones. They are called such by their general nature since cones have a cone shaped head and the rods are generally round with no tapering toward their heads. We have shown several of these in Figure 2.x.

Figure 2.4 Rod and Cone Cells in the Human Retina

There are 130 million rod type cells in the typical human retina and there are about 6.5 million cone cells. The cone cells are distributed at about 120 cones per degree of arc which allows for a resolving capability of 1 min of arc (see Kunt at. al. p 175). Rods are the sensors for the shapes of objects whereas the cones are the sensors for color and light intensity. The eye responds to light over the range from 350 nm to 750 nm. The mechanism for this response is the photochemical responses of the molecules called carotenoids. These molecules are coiled tightly until impacted upon by photons. Then they rapidly uncoil and when recoiled release electrical energy that triggers the nerve cells to transmit their messages to the visual cortex. Thus vision is a complex optico-chemical-electrical process.

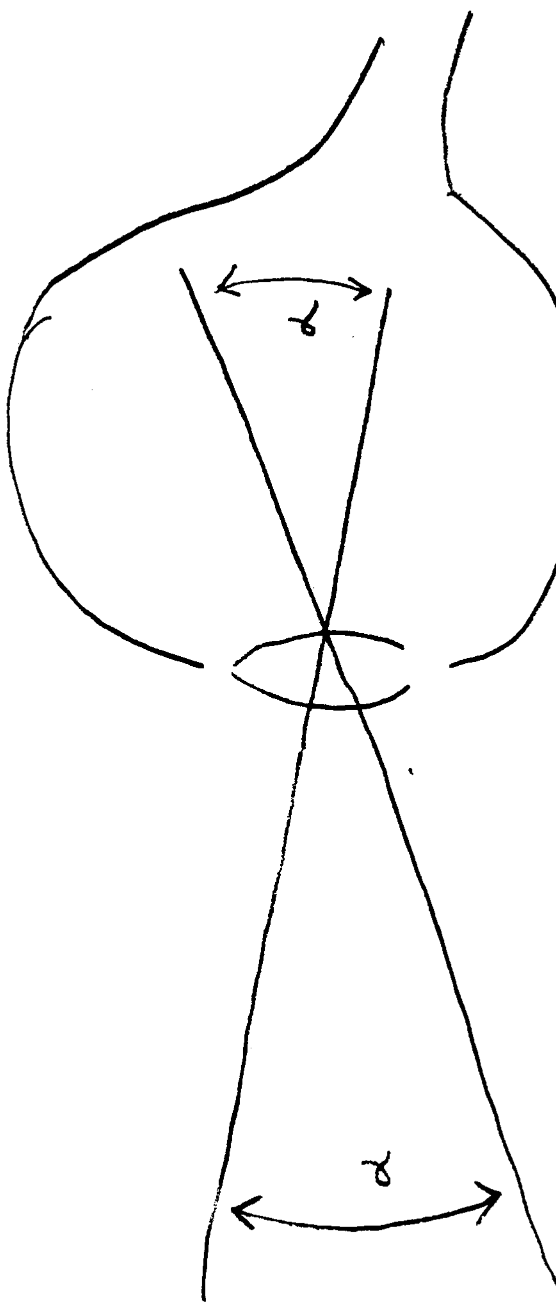
The resolving power of the eye is one of the more critical factors that we shall be focusing on in this chapter. Let us take a closer look at the 1 min of arc number and see how this effects types of images that we view.

Let d equal the distance between two images at the image plane and let A be the angular distance in degrees of arc between the two points. This is shown in Figure 2.x. Let r be the range from the eye to the image plane. hen we have:

$$d = r A$$

and we can substitute A of 1 min of arc to this equation. Recall that 1 min equals 1/60 degree, and one degree is (2 pi/360). Thus we have one minute is equal to 0.0003 radians. At one meter viewing distance we have a resolution of 0.3 mm. At 0.3 m or about one foot, we have a resolution of 0.1 mm. We can compare this to a typical television set of 1 foot in height. If there are 525 lines per scan then there are 1mm between the lines. If we sit 1 foot from the set we can resolve 0.1 mm or ten times the resolution of the set. However if we view it a 10 feet then we are at the resolving power of the human eye.

Figure 2.5 Resolving power of the Human Eye



Thus we can see that the issue of better resolution of the human eye to such technologies as HDTV may be fundamentally specious in that unless we view such system at close range, we cannot hope to obtain an fundamental improvement.

In a fashion similar to the spatial resolving power of the eye we can examine the fundamental color perceiving power of the eye also. This has been discussed by Shepherd and others and will be left to the reference materials.

The eye has six major modalities that impact on how it is effective in multimedia communications. These modalities are:

o Photosensitivity: This is the sensitivity of the eye to the intensity of the light. In particular it is important in the eyes ability to receive light from diffuse source that reflect light from nondirect elements. A typical example is the reception of light from movies theater screen which have marbled surfaces to create diffuse reflections. Without those reflections, we should have a specular reflection which would be visible;e to the viewer in the direct reflection direction.

o Form Discrimination: This factor relates to the resolving power that we have discussed. The discriminating capability os dominated by the position of the cone neurons in the retina and thus limit the resolving capability of the eye.

o Movement Discrimination: This is the ability of the eye to detect change. This factor is important in such areas as film and television. The eye has a time constant on the order on 0.1

second, such that any events that are shorter than that are not discriminated by the eye.

o Binocular vision: This factor allows the human to discriminate depth in a spatial context. It is a function of the two eyes in resolving the dfferential responses on the retina and the ability of the visual cortex to process the images and resolve depth.

o Polarized Light Discrimination: There is a limited ability of the eye to discriminate in the polarization of light.

o Color Discrimination: The eyes of humanoids are one of few anthropoid eyes that can respond to color. The color response is viewed as a evolutionary response to the limitation of night vision in the species.

2.2.1.1 Achromatic Images

Achromatic images are those that are either black or whit, or even mixed with shades of gray. More specifically, achromatic images do not have any color. An achromatic image is generated by the response of the eye to electromagnetic waves that are reflected from a surface and received by the eye. If we have a surface that reflects light at all frequencies equally well, and that total reflection occurs at the surface, then that surface is perceived as white. On the other hand if we illuminate a surface with a source of light that has all frequencies and the surface reflection characteristics are such that it reflects all incident light with zero reflectance then that element appears black.

Let us consider this black/white occurrence i more detail. First we must remember that we see objects only as a result of the light that they reflect, or in some cases emit. Let us focus on reflected light only. Let us assume that the incident light is characterized by a wave of the form $E(x,y,z,t)$ for all points in space. Further let us assume that the wave can be characterized by a Fourier transform called $E(x,y,z,f)$, where:

$$(2.4)\quad E(x,y,z,t) = \quad E(x,y,z,f)\ \exp(-2\ ft)\ df$$

If we further calculate the amplitude of the wavefront at each frequency, or equivalently look at the spectrum of the signal, we see that for white incident light we have:

$$(2.5)\quad ||E(x,y,z,f)|| = \text{Const}$$

Which is constant for all frequencies or wavelengths.

Now let us assume that the reflected wave is given by $ER(x,y,z,t)$ and that we can show that if the surface has a reflectance at frequency f of, $R(f)$, then;

$$(2.6)\quad ER(x,y,z,t) = \quad E(x,y,z,f)\ R(f)\ \exp(-2\ ft)df$$

Now if we assume that $R(f)$ is independent of frequency, that is:

$$(2.7)\quad R(f) = R$$

We then have;

$$(2.8)\quad ER(x,y,z,t) = R\ E(x,y,z,t)$$

Then, clearly, if the incident ave is white light, we have for:

(2.9) $ER(x,y,z,t) = E(x,y,z,t)$ (R=1 and is white)

or;

(2.10) $ER(x,y,z,t) = 0$ (R=0 and is black)

When we discuss chromatic surface we shall see that this changes since R is now frequency dependent.

When we consider black and white images, we are generally considering that the white light is reflected from the surface, independent of frequency. This means that the surface has no specific reflection characteristic. At most, the surface may absorb the light uniformly, thus resulting in a graying of the light. The grayness in this case is a result of the loss of light intensity. Thus what we see from a surface is the level of intensity of white light. If we let I be the light intensity, then using standard electromagnetic principles, we have:

(2.11) $I = ||E_R||^2$

Using 2.8 we can write;

(2.12) $I = R_2 \; ||E||$

Let us now consider the effects of light on film. Photographic film is simply a gelatin base interposed with silver halide crystals. These crystals are composed of silver halides of the form such as AgBr, eg silver bromide. When light shines upon these crystals, the reaction occur:

(2.13) Br- ---> Br + e

)2.14) Ag+ + e ---> Ag

That is, silver is reduced into the gelatin substance. Upon development (see Jacobson et al) the developer enhances the result of silver generation in the crystals that have been exposed to the light. Thus in a region exposed to light, a large set of silver crystals are produced and in those areas not exposed that are small i number. During development, the remaining silver halides are washed away, leaving only silver in the developed plate, acting as a darkening agent to transmitted light. This result in the standard negative that we have seen in many cases. We should remember that the density of photons is directly proportional to the light intensity so that we can expand the analysis to what we have discussed in the development of (2.11).

Now the resolution of film depends upon how well we can generate a suspension of silver halide crystals, and also how effectively we can develop the film. Such effects as diffusion of the light and scattering in the film plate may lead to the less than crisp photos we may have often seen. In the most effective film resolution that we may have experienced, we look at the standard x-ray film. Here the same film process is use but the incident radiation is at the x-ray frequencies or wavelengths. The x-ray is used because of its ability to penetrate the hard and soft tissues of the body. The net result is a film plate that is exposed in proportion to the transmitted, rather than reflected energy.

The x-ray is a typical example of the continuous gray scale image that is what we commonly refer to as the black and white photo. Continuous gray scale represents the fines gradation of exposed silver crystals that can be obtain on photographic file. The resolution of the typical x-ray film is determined by the density of the silver crystals on the film plate and the ability of the developer to keep the diffusion of the developer process to a minimum.

If we were to digitize the x-ray film, on the basis of human ability to resolve the resolution of the image in terms of line resolution and to resolve the image in terms of shades of darkness, a display of 20" by 20", having 2,000 by 2,000 pixels, or 4 Million pixels per 400 sq in is needed. This equals a resolution of 10,000 pixels per sq inch. In addition the display per pixel must resolve 4096 levels of gray, which is determined to be the maximum level of resolution of shades of darkness of human resolution. Thus a figure of merit for an x-ray is 48 million bits per x-ray, based upon 4 million pixels, at 12 bits per pixel.

This x-ray example is the simplest example that demonstrates the need for combining the concept of the original image of the human body, thee ability to express it in terms of a digitized continuous tone image, and the corresponding use of human responsibility as the measure of comparing the closeness of the digitized image. We show in Figure 2.3 a set of images, that range from a 100 year old photograph, a current x-ray display, and the corresponding digitized display of the x-ray.

This x-ray example has led us into the discussion of gray scale techniques. A gray scale image is typically a digitized version of a black and white image. Remember that the black and white image is in reality a quasi continuous display of the intensity either reflected or transmitted by an achromatic sources, from or through a surface. The gray scale representation is the means of taking that quasi-continuous image and reducing it to a bounded digital representation. We say that the original image is quasi-continuous because of the fact that the silver generated on the film developer plate is microscopic in level and for the most part is as continuous as we can get in nature. In its ultimate limit all such figures are truly discrete.

Figure 2.6 Film Image

a Old Photograph Image

b Dot Matrix Printer Image

c X Ray Gray Scale Image

d Newspaper Halftone Image

e Four Color Magazine Image

We shall use the development of the images in Figure 2.2 as the basis to describe many of the other image representation that are possible. Let us continue with our gray scale representation. There are two elements that are of concern in the digital representation of a gray scale image. They are:

o Pixel Resolution: This issue relate to how many pixel per square inch are necessary to be unresolvable by the human eye. Based upon both psychological and psychological studies, we find that (see the reference by Kunt et al) the retina can resolve images set apart by 1 min or arc. Anything smaller than that and the image falls on the same cell. Thus if we have a viewing distance of about three feet from the screen, we can use the following:

(2.x) $d = r \text{ ang}$

where d is the distance of the pixels on the screen, r is the 3 feet, and ang is the resolutions in radians. Substituting the parameters we find that d approximately equals 0.01 in or 100 per inch. This yields 10,000 pixels per square inch. This is exactly what we displayed earlier.

o Lightness resolution: In a similar fashion, we have found that the retina can respond to about 4000 shades of light intensity and this yields about 12 bit of shades of gray that can be used.

Thus for gray scale coding, the human eye needs about 10,00 pixels per square inch with 12 bits per pixel. This means that the typical, continuous gray scale digitization image requires 120,000 bits per square inch at a 3 foot observation distance.

Let us examine the gray scale quantization a bit further. Let I represent the light intensity at a particular pixel. Let us start with a single bit gray scale quantization. That is at the pixel we use either black or white. In particular, let I0 be the threshold and say:

(2.x) $P = P0$ if $I < I0$

and

(2.x) $P = P1$ if $I >= I0$

Then code this pixel into 0 or 1 according to this level of quantization. The choice of I0 is not arbitrary, as we shall show shortly. Let us expand this to two bits.

(2.x) $P = P0$ if $I < I0$

$P = P1$ if $I0 <= I <= I1$

$P = P2$ if $I1 <= I <= I2$

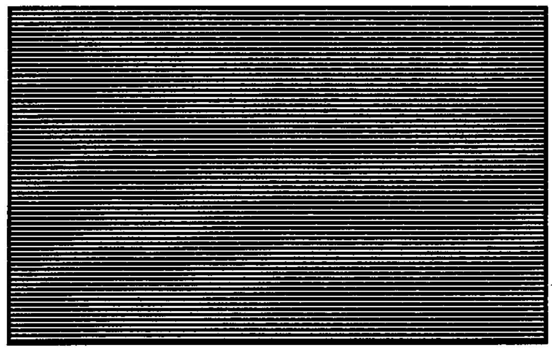
$P = P3$ if $I2 <= I$

We can now code this into two bits. Again the choice of I1 and I2 are not arbitrary. We can expand this into 12 bits, or 4096 levels by selecting I0 through I4095. Again the choice is not arbitrary. Figure 2.5 depicts the optimal gray scale coding map for the encoding of these intensities. We find this through many measurements of human responses and it is also related to the ability of the retina to respond to stimuli as well as the brains ability to distinguish these elements.

Figure 2.7 Film Resolution (35mm,70mm etc.)

Negative.

Resolution
equivalent to
1600 lines/picture
height..



Graininess is due to the
light-sensitive
crystals, of 1/1000th
of a millimetre in diameter,
that form clusters.

Screen image
with 1.85:1 aspect
ratio height
with 11.3 mm
aperture height.

Resolution
equivalent to
700 lines/
picture height.

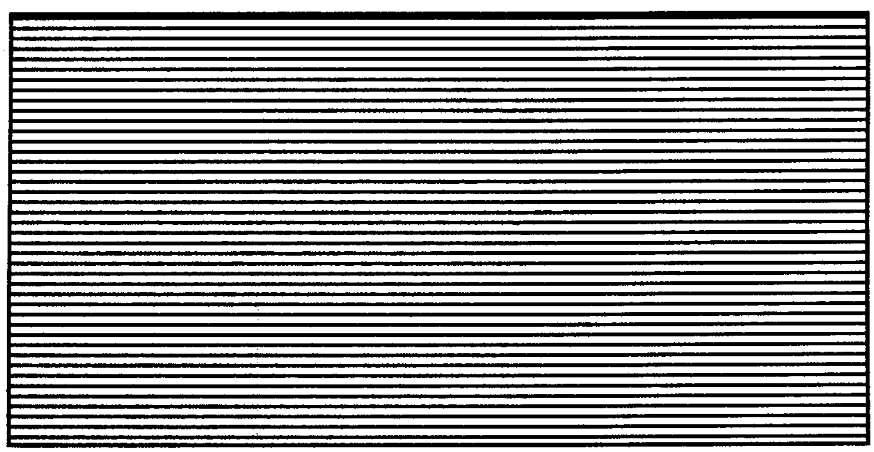


Figure 2.2 : Resolution of 35mm Film.

Figure 2.8 Gray Scale Resolution

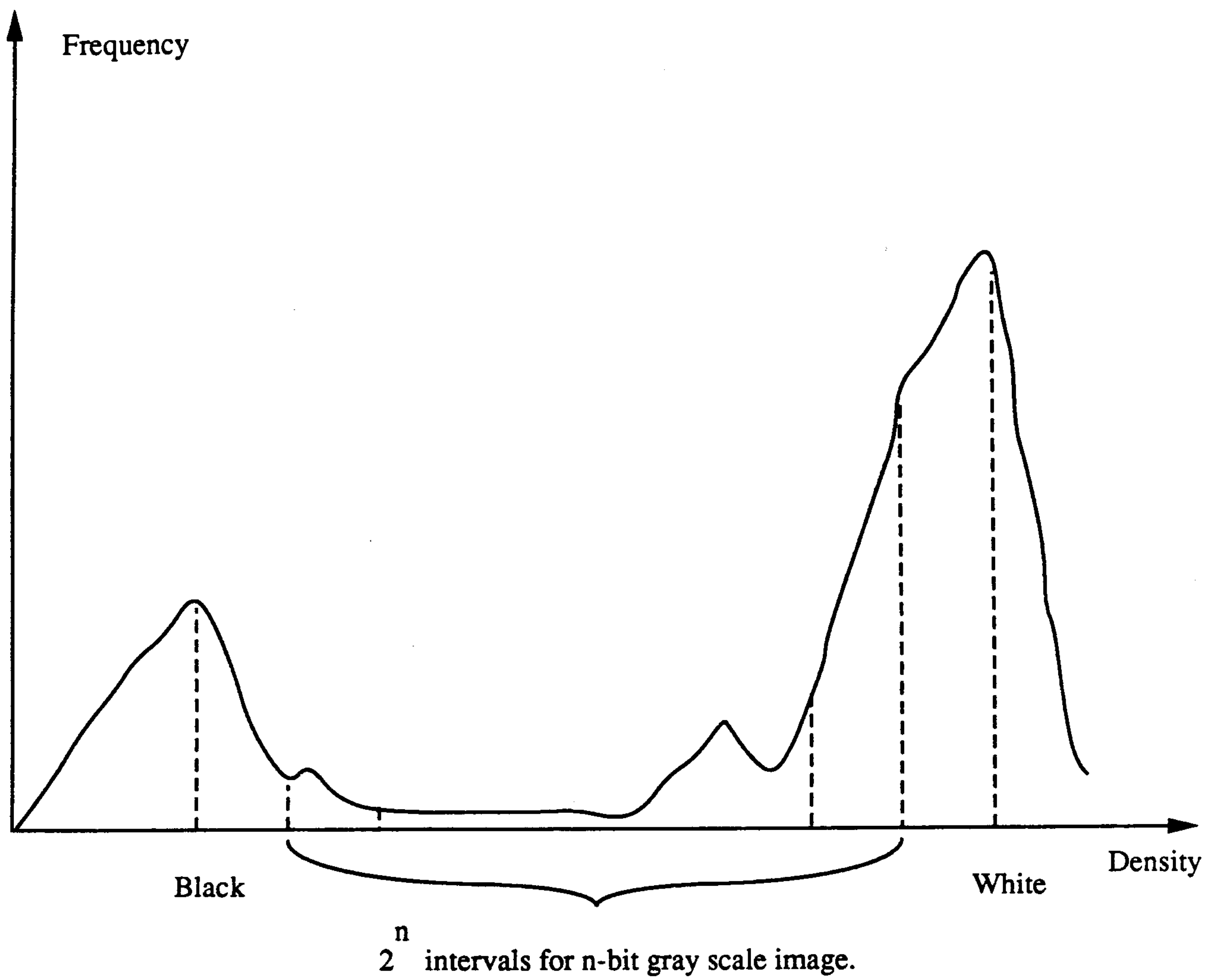


Figure 2.3 : A Gray Scale Resolution.

Now we can view gray scale coding in this form as the ultimate in the achromatic coding or images from a continuous form. At the other extreme is the mapping of images onto dot matrix printers that are common in the PC world. A dot matrix printer can produce about 50 dots per inch or 2500 dots per square inch. Again compare this to the eyes ability to del with 10,000 per square inch (see Durrett p. 223). However, the dot matrix printer prints a dot, black, or no color, white. Thus it is only one bit per pixel. This is simply 2500 bits of information per square inch. This compares to the eyes ability to deal with 120,000 bits per square inch. This is a 50 to 1 reduction in information. Figure 2.2 displays a dot matrix image that shows this resolution.

We can expand this capability with a technique called digital halftoning. Quite simply, halftoning is a technique that combines the limited capability of having only one bit per pixel into sets of such pixels that, to the eye, look like more than just one bit, they looks like gray scale. It is based upon the fact that collections of bits have reflectivity parameters that appear to the eye as gray scale. The work by Ulichney shows how this gray scale factor can be obtained. In addition, there are many gray scale algorithms that permit the optimization of the visual response. These are presented in both Ulichney and the work of Foley and VanDam. Let us use a simple example to show how halftoning is developed.

Consider the images shown in Figure 2.x. Here we have a set of black or white dots that fill one of three possible matrices (eg Figure .x a,b,c) In Figure 2.x a we have a 2x2 matrix that has five possible entries. We have a set with no dots, one with one, two, three and four. Thus with 2 dots we have 2^2 or 4 possible entries. Each of these entries have less and less reflectance or in turn have a perceived shade of grayness. Let us assume that we are taking a gray scale image with 100 dots per inch and compressing it down to a halftone image with only 50 dots per inch. Let us further assume that we wish to have the halftone image approximate the gray scale image. We shall proceed as follows:

o Use the halftone basis set of the 2x2 matrix and note that since halftone uses only 50 dots per inch, that two dots will equal 4 dots of the gray scale image. Specifically if we define:

D GS = the gray scale image density in dots per inch

D HT = the halftone image density in dots per inch

N HT = the size of the half tone cell in dots

G HT = the number of gray scale levels in a halftone set

then we can show that we can compress the gray scale into the halftone.

o The gray scale compression can be given as follows. The number of gray scale dots compressed is N GS which equals:

N GS = N HT * (D GS/ D HT)

o But in the N GS cells there are L GS gray scale levels per cell which equals L GSB, gray scale bits. Remember that this is at most 12 bits or 4096 levels. Thus there are a total of:

TB GS = L GSB * (N GS * N GS)

TB GS gray scale bits in the gray scale cells. For example in the case of 4x4 gray scale and 12 bits per cell we have 16x12 or 192 bits per gray scale cell.

o However, in the halftone cell, we have 2x2 or 4 cells with only one bit per cell. This yields an equivalent

TB HT = L HTB * (N HT * N HT)

In this case there are 4 bits. Thus the halftone has reduced the bits from 192 to 4, or a bit reduction of 48 times. We obviously loose a significant amount in this process, but the resulting image is often quite acceptable (see Figure 2.2).

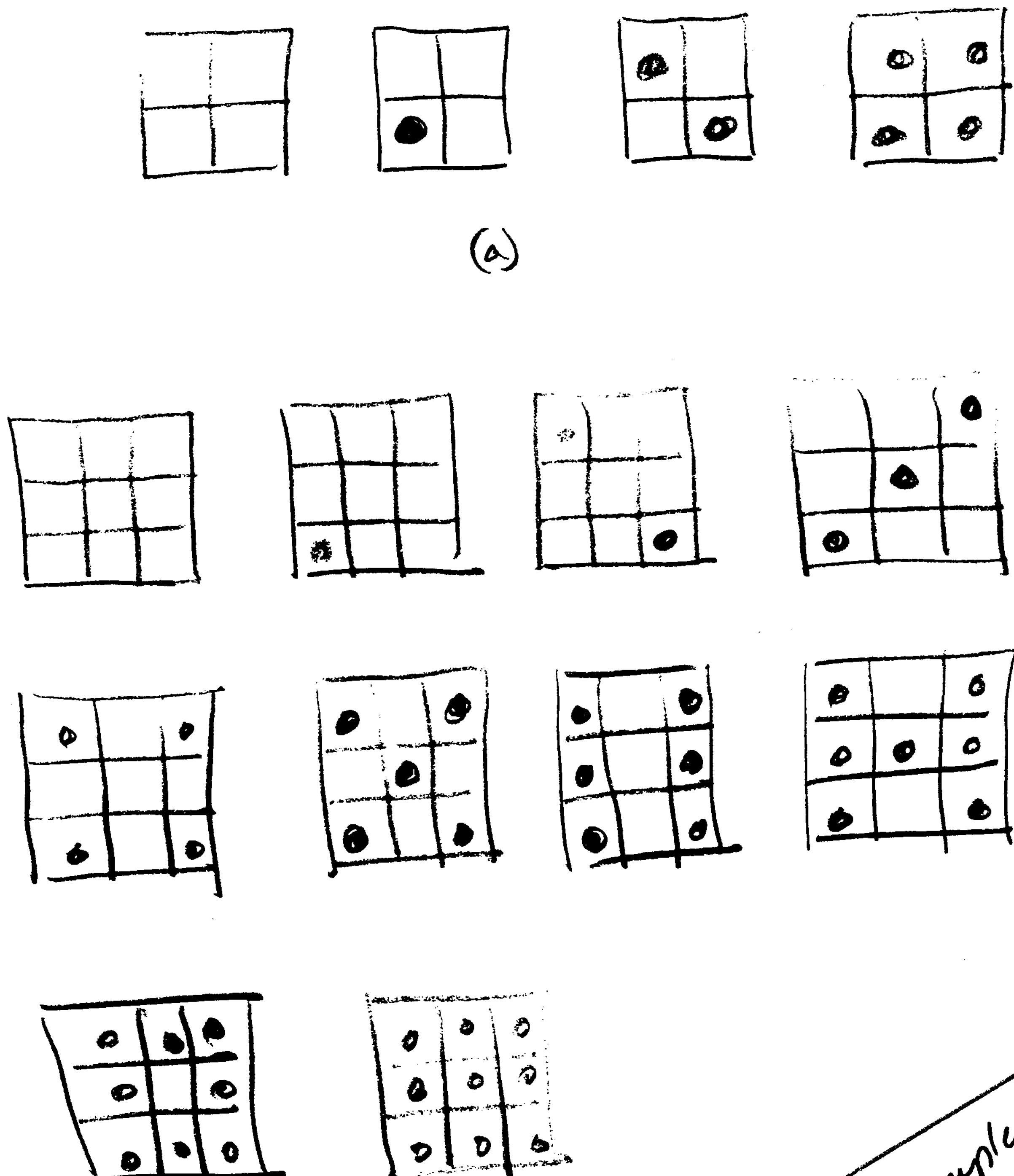
o To select the proper halftone bit pattern, we must look at the several sets of gray scale dots or cells and average the gray scale level, and then further quantize it to fit the gray scale levels that have been allowed in the halftone quantization.

This process of halftone production is a well accepted practice in the field of newspaper print and in many other printing systems. It is the most obvious choice in use of dot matrix printers.

There are several problems that arise with the halftone imaging however. They typically result from the use of a set of periodic basis functions to generate the pseudo-gray scale images. This can be eliminated by the use of entering randomness in the selection of the dot location and in the assurance of no long run sequences of the same pattern even if the average gray scale density remains relatively constant.

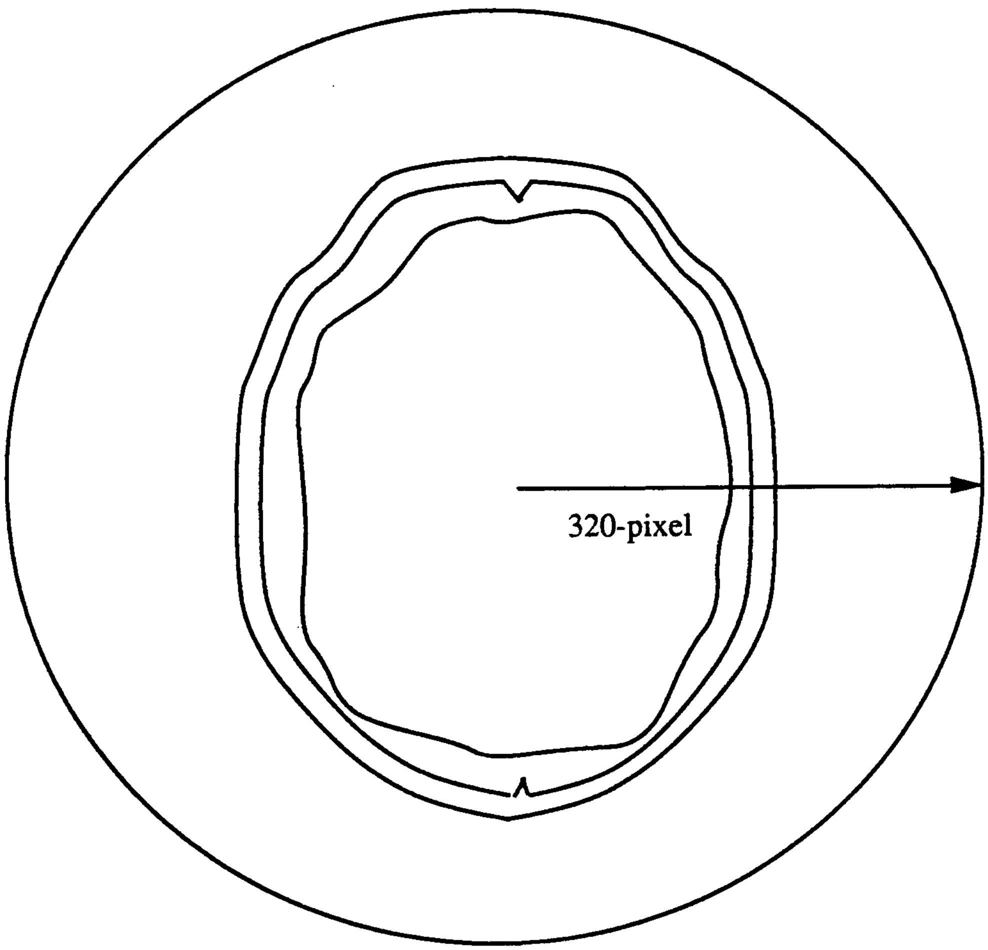
Figure 2.10 Halftone Imaging

Figure 2.9 Halftone Image Basis Sets (a) N=2,(b) N=3,(c) N=4



More Examples: Use Problem Sets.

Figure 2.11 Halftone to Gray Scale Comparison a : Computed Tomography Image of the Brain.



b : Resolution of a Computed Tomography Slice.

2 bytes per pixel,
average entropy ≈ 8 bits per pixel

Figure 2.5 : A Gray Scale Medical Digitization.

2.2.1.2 Chromatic Images

The ability of the eye to discriminate colors is one of the factors that discriminate the human from the other animal species. The achromatic discrimination abilities relies upon the ability of the eye to be sensitive to the intensity of the radiation of the surface of the retina. The color discrimination capability is the ability of the same neurons to respond in a differential fashion to photons of differing wavelength.

In this section we shall develop the concept of chromatic images, and show how those image are represented and formed and how we can characterize those images for the ultimate purpose of sizing the source and storing the data. The normal visible range of the eye is from 450 nm to 650 nm. Some humans may respond to slightly lower and higher wavelengths. The typical sets of standard colors are :

- o Pure Blue 470 nm
- o Pure Green 505 nm
- o Pure Yellow 575 nm
- o Red > 610 nm

The concept of pure green is really a psychometric factor. It states that when this color, defined in terms of a narrow linewidth source of radiation, is shown to a large group of people, then they agree in large numbers that the wavelength shown is the pure representation of yellow to them. In contrast, there does not seem to be a uniquely agreeable wavelength for red.

We can define the colors that we have seen in terms of the source of the light if the light is transmitted directly into the eye or of the reflective nature of the surface for light that is reflected into the eye. Let us consider first the issue of transmitted light and then reflective light.

If we use a narrow laser source, we can define the transmitted light in terms of the power density a a function of wavelength. Define P(l) as this spectrum and we have:

$$P(\quad) = Po\ f(\quad)$$

where Po is the peak intensity and f( ) is the spectral characteristic. Light is said to be saturated depending upon how broad a bandwidth the source is. Thus a narrow band light source is said to be more saturated.

We can in a similar fashion look at reflective surfaces and consider the effects of white light being reflected from the surface and entering the eye. The material on the surface will have differing reflective characteristics depending on the wavelength. Let Po be the intensity of the incident radiation, and we shall assume that it is white light. Let R(L,x,y,z) be the reflectivity of the surface as a function of the wavelength and the position. Then we can define the light incident on the eye in the standard fashion.

We can now consider other ways of generating different colors. There are basic types that we have depicted in Figure 2.x. These types are subtractive and additive.

Figure 2.12 Color Mixtures

(a) Subtractive

(b) Additive

The subtractive techniques takes white light as the source in a transmissive fashion. It then passes the light though filters that take out narrow bands of light and leave a combination of all other colors. Consider the example in Figure 2.x a. Here we have shown white light on a yellow filter that passes right that has had yellow removed and then we pass it through a filter that has a capability to remove cyan. The resulting color is green as perceived by the observer. It should be noted that this perception depends highly on the narrowness of the filters and the ability of the filter to be centered on the desired wavelength.

Consider now the additive form of color mixing. This is the more common form and the one that we shall be using at length. In this case we take three primary colors, red, green and blue and start with equal intensities of each. We then vary the intensity of the separate colors to create the total color. The resulting total additive color is:

$$IT = I(R)+I(B)+I(G)$$

The resulting colors can be a wide mixture of all colors perceived by the human eye.

This effect can be theoretically perceived by looking at the three separate spectra shown in Figure 2.x. Here we have shown somewhat broadened spectra. We then combine two of the spectra and generate a third combined spectrum that peaks somewhere in between the two. It is this effect that allows for peaking at any point and thus having the psychometric effect of a continuum of colors.

Figure 2.13 Additive Spectra for Continuous Color Generation

The recognition of this additive effect of emitted spectra and the generation of continuous color has led to various characterization of color mixtures. The simplest is the chromaticity chart that starts with the additive equation of the sources and normalizes the overall intensity. That is we start with the equation:

$$T = 1 + R+B+G$$

so that we can define :

$$B = 1-R-G$$

Now we can plot the amount of red and green on a two dimensional chart, knowing that the amount of blue is merely subtractive. This is plotted in Figure 2.x and is called the chromaticity chart.

Figure 2.14 Chromaticity Chart

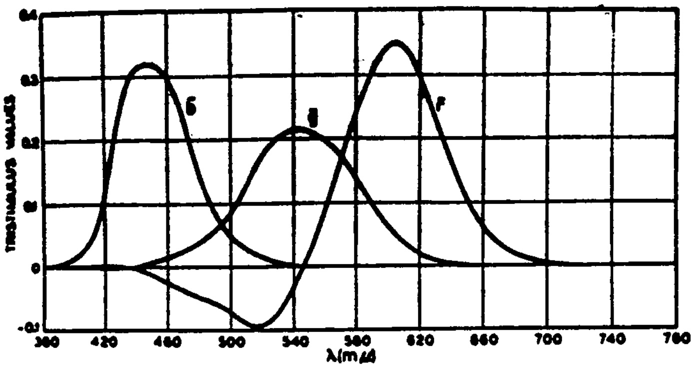


Fig. 4. Tristimulus values $\bar{r}$, $\bar{g}$ and $\bar{b}$ of spectral stimuli of equal radiance for the Standard Observer (from Wintringham [13]).[3]

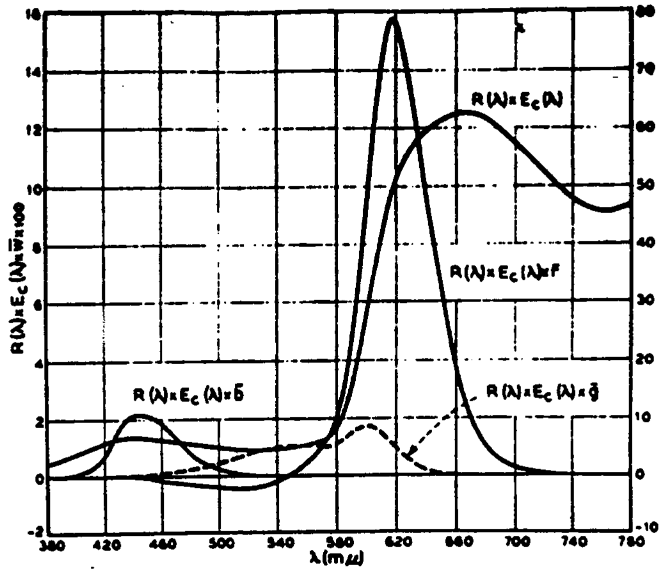


Fig. 6. Products of the tristimulus curves $\bar{r}$, $\bar{g}$ and $\bar{b}$ with the reflectance of a color sample, $R(\lambda)$, irradiated by Illuminant C, $E_c(\lambda)$ (from Wintringham [13]).[3]

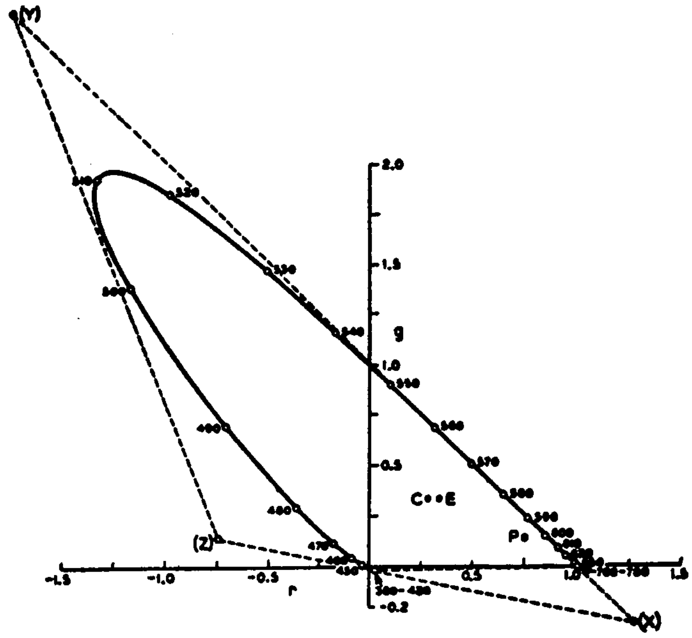


Fig. 5. The rg chromaticity diagram for the Standard Observer (from Wintringham [13]). The wavelengths (in nm) of the spectral colors appear on the horseshoe shaped locus. Point E represents equal-energy white, C represents Illuminant C which is a standard bluish-white source, P represents a specific color sample irradiated by Illuminant C and (X), (Y) and (Z) are the standard CIE nonphysical primaries discussed in Section II. D.

that is illuminated by a light of a specific spectral distribution. The spectral distribution of the reflected light may, of course, be thought of as being composed of an infinite series of spectral colors. To determine how much of each primary is needed in the mixture, a product is formed of each of the tristimulus values with the spectral reflectance of the object as shown in Fig. 6. The areas under each curve, as obtained by integration, are the desired tristimulus values R, G and B for the sample. Chromaticity coordinates can be calculated using (3) and the point "P" is plotted in Fig. 5

In 1931 such calculations were commonly performed on desk calculators, and the negative lobes of the functions of Fig. 4 introduce negative product terms in which the negative sign is error prone with repetitive summing and differencing operations. It would be much better if there were no negative lobes, and it would be convenient if the quantities were zero over as large a range as possible. Calculation of the luminance of a color would be made much easier if the luminosity coefficients of two of the primaries were equal to zero. The luminance of a color would then be equal to the number of units of the other primary used in the match.

It was these considerations, among others, that led the CIE to propose a new set of primaries. The spectral locus is totally contained within the triangle formed by these new primaries denoted by (X), (Y) and (Z) (as can be seen in Fig. 5) implying that all spectral colors can be matched with a positive quantity of each primary. The use of such "nonphysical" primaries should be no cause for concern. For measurement purposes any real set can be used and the results can be transformed by a 3 × 3 matrix to the nonphysical set.

Luminosity information is obtained from the tristimulus value of the (Y) primary—the luminosity coefficients of the other two primaries are equal to zero. The resulting tristimulus values $\bar{x}$, $\bar{y}$ and $\bar{z}$ are shown in Fig. 7. Note the all-positive nature of the functions, and that $\bar{y}$ is identical to the V_λ curve of Fig. 2. The xy chromaticity diagram for the 1931 CIE Standard Observer is shown in Fig. 8. The equal-energy white (E) has the coordinates (1/3, 1/3) because it is the reference white for this system. All the color mixture properties that we have previously described for the rg diagram are valid for this diagram, however, the equations for color mixture are especially simple for this case. If we are given the chromaticities x_1, y_1, x_2; y_2 and their luminances L_1 and L_2 the chromaticity of the mixture is simply

$$x_3 = \frac{x_1(L_1/y_1) + x_2(L_2/y_2)}{(L_1/y_1) + (L_2/y_2)}$$

$$y_3 = \frac{L_1 + L_2}{(L_1/y_1) + (L_2/y_2)} \qquad (4)$$

and the luminance is

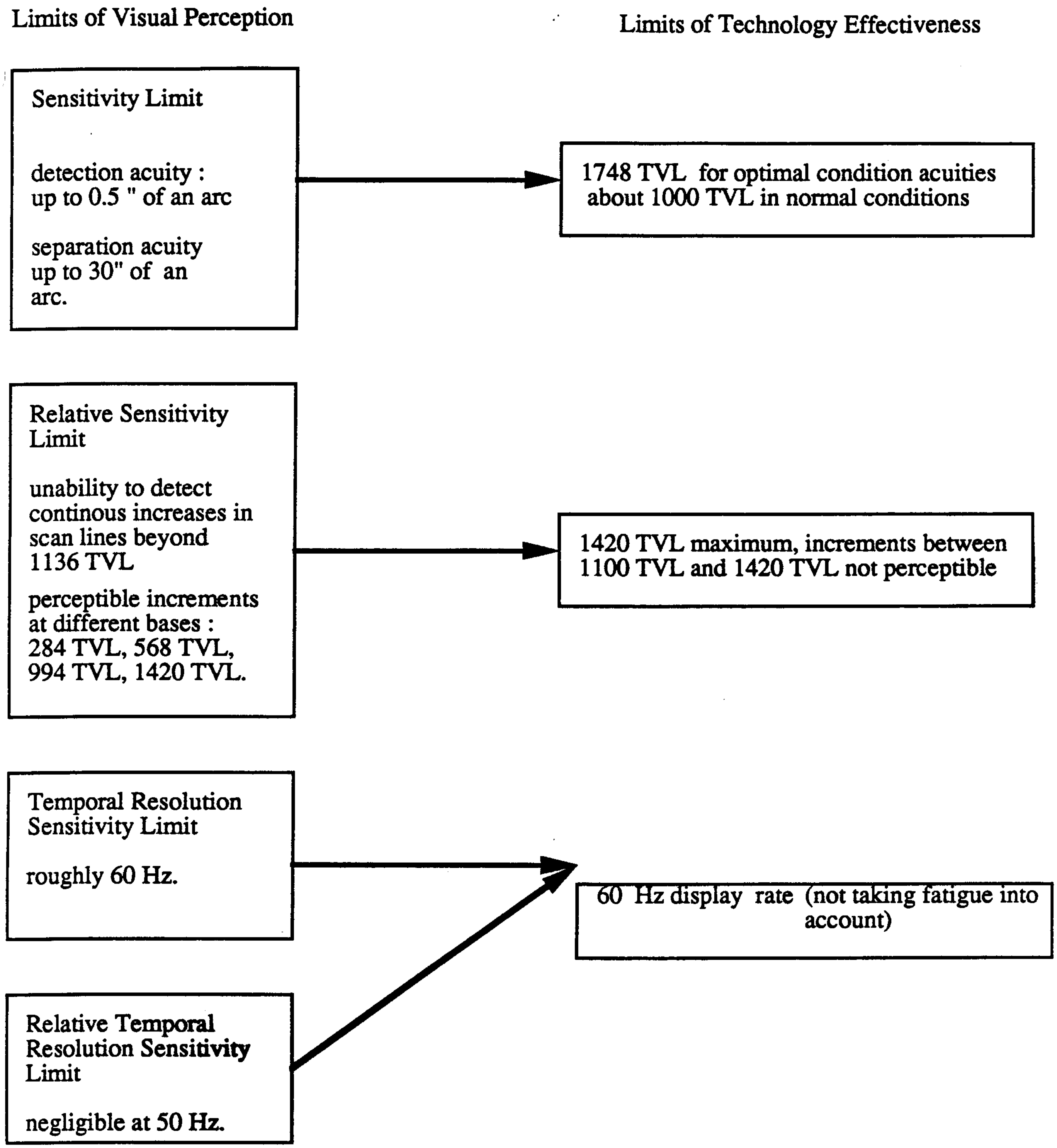


Figure 5 : Limits of Visual perception and Corresponding Limits of Technology Effectiveness.
(From 'Limitations of Current Psychophysical Approaches to Display Evaluation'
P.J. Hearty in
'High Definition Television Colloquium '85'
Minister of Supplies and Services, Canada, 1985).

We can then generate any desired color by combining the amount of red or green on this normalized basis. Note that at R=G=0 we have pure blue. In addition we can note the extension of this chart beyond the region for the sets of colors generating them. This is a measure of the psychometric factors generating these colors.

The implications for the use of this psychometric definition of color and the human response falls into many categories. Two of them are most important in the areas of multimedia communications. The first is the use of color in the RGB combinations for the generation of multiple colors on a CRT screen. The basic CRT screen is in actuality a set of overlays of Red, Green and Blue phosphors. Using these combinations we can generate any set of colors on a CRT screen. Consider the types of CRT displays as shown in Figure 2.x (Merrifiedl p.70). The RGB phosphors are located in a spaced form on the surface of the screen. Three different electron guns are displayed behind the screen and are focused through a set of masks. The Figure shows four types of masks that are used in practice. These masks are called shadow masks and ensure that the guns of the different colors do not impinge on the wrong phosphor.

The resulting image is a collection of the RGB phosphors that are illuminated by an intensity in proportion to the color that is desired at the specific location.The granularity of the shadow mask is a measure of the pixels per inch of the display. For example, the distance between the Red phosphors on the horizontal or vertical axis may vary from 0.6 to 0.2 mm. The discrete pixel is the combination of a set of pixels. If we connect an RGB combination in a triangular fashion, then in Figure 2.x we can see that with 0.3 mm spacing, we see that the pixel of color, an RGB triangular combination is on the order of 0.2 mm. As we discussed in the resolving power of the eye, this gives significant resolution at close distances.

Specifically, with 2,000 by 2,000 monitors, we can have 6,000 by 6,000 individual elements of color at the RGB level yielding the overall combination of 4 million pixels. With a 60 cm by 60 cm display, we have 3000 RGB cells at 0.2 mm spacing. In the current state of the art we have 0.1 mm spacing of these cells.

Figure 2.15 CRT RGB Masking Techniques (Ref Merrifield)

Shadow-Mask CRT Technology

The shadow-mask color CRT assembly consists of three closely spaced electron guns, a shadow-mask, and a three color phosphor screen. Focused electron beans emitted from each primary gun pass through apertures in the metal shadow-mask and impinge upon phosphor dots for each corresponding color. Figure 3-9a illustrated a *delta gun* configuration of a shadow-mask CRT. The three electron guns are arranged in an equilateral triangle, or delta. Each shadow-mask aperture allows the three electron gun beams to project onto an inverted delta or triad of phosphor dots. The angle of incidence of an electron beam as it passes through a shadow-mask aperture determines the color of phosphor dot it excites. Electron beams of a particular gun are blocked by the shadowing of the mask from impinging upon the other two color phosphor dots of each triad. A shadow-mask CRT has a very simple mechanism for selecting color. The three independent guns in the shadow-mask design provide independent control of the luminance of the red, green, and blue phosphors. In this manner, it is possible to reproduce any color within the chromaticity triangle formed by the primary colors.

Several configurations of gun alignments, mask structures, and phosphor arrangements are available. Figure 3-9b illustrates an in-line gun configuration projecting through mask apertures onto a delta-type phosphor dot faceplate. The in-line gun shadow-mask

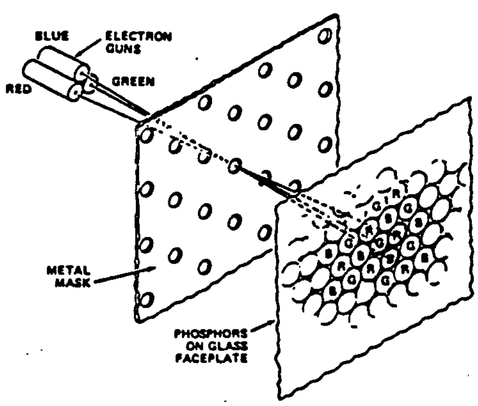


Figure 3-9a Delta-gun/Delta mask color CRT.

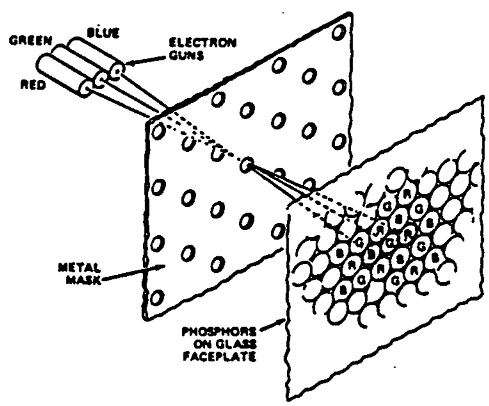


Figure 3-9b In-line gun/Delta mask color CRT.

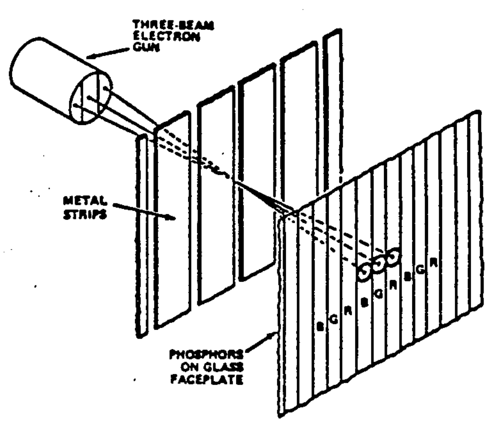


Figure 3-9c In-line gun/Slotted mask color CRT.

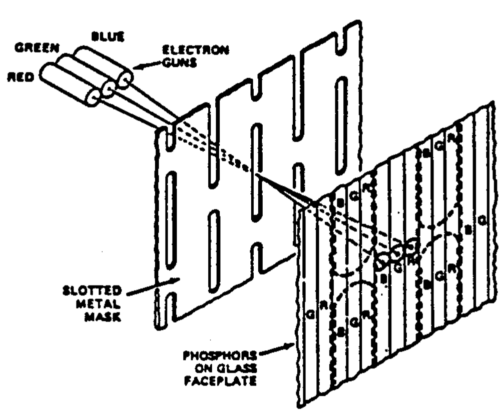
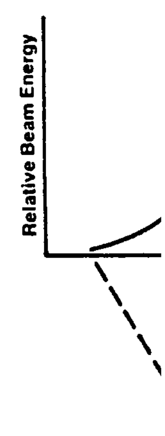


Figure 3-9d Single lens in-line gun/Metal-strip mask color CRT.

CRT has th
mask and pl
the in-line g
through a sh
cause of the
Resolution c
to their smal
beam at the
also availabl
vertical pho
a higher lum
The granu
distance bet
ranging fron
are conside
be confused
several mas
electron opt



Shadow Ma

Figure 3-10 S

Chapter 4

2.2.2 Means of Storage

The ways of storing still images has evolved significantly over the past decade. The method most used ten years ago was that of paper tape and today there is extensive use of laser disk storage technology. In this section, we will develop some of the current and some of the potentially significant means of storing and retreading still images.

As part of the storage process, we first want to consider the means of compressing the image data and as part of that understanding the ultimate information in the image itself. This latter issue relates to the minimum number of bits that are necessary for the representation and storage of the image. Let us begin with the

Figure 2.16 Disk Storage Systems (WORM)

2.2.3 Means of Reconstitution

The issue of reconstituting still images is one of obtaining either the best video display techniques or in using many of the existing and improving methods of hard copy display. We often think of reinstating the image on a video display and do not recognize that the use of hard copy display terminals is also a significant means of reconstituting the image. There are dramatic advances being made in the areas of both displays and hard copy devices and these will be reflected in the way the systems will evolve with the capability to provide significant editing and composition interactivity on displays.

(1) Information Definition

(2) Information Content of Image

(3) Compression Techniques for Still Images

Figure 2.17 Gray Scale Medical Imaging System

Figure 2.18 Halftone Dot Matrix Printers

Serial Printer Technology	Speed	Cost Range	Print Quality	Graphics Capability
Dot Matrix	up to 800 cps	$199 - $5,600	Draft to Near Letter	up to 380 dpi
Daisywheel	up to 90 cps	$495- $3,350	Letter only	rules only
Ink Jet	up to 15 pages / mn	$495- $14,495	Draft to Letter	up to 480 dpi
Ribbon Resistive	up to 132 cps	$1,399- $1,699	Draft to Letter	up to 240 dpi

Legend : - cps : characters per second
- dpi : dots per inch.

Table 2 : Different Serial Printer Technologies and Their Characteristics. (From 'Datapro Reports on Microcomputers' McGraw-Hill, 1989).

Figure 2.19 Color Hard-Copy Devices (Ref Durrett, Color & Cmptr)

Input Device	Characteristics
CDP9000 (Calera)	recognizes characters, character types and positions, separates characters from graphics, recognizes up to 270 cps
Datacopy Scanner Model 830 (Xerox Imaging Systems)	scans continuous tones and line art, resolution can be enhanced up to 450 dpi, up to eight bits per pixel, six preset halftone patterns, one programmable pattern, 16 levels of contrast
ScanJet Plus (Hewlett-Packard)	scans continuous tone and line art, resolution can be enhanced up to 1,500 dpi, continuous tones converted to halftones or preserves up to 256 shades of gray
3119 PageScanner (IBM)	scans continuous tones and line art, resolution of 240 dpi, 128 levels of gray
Discover 7320 Model 30 (Kurzweil)	optical character reader, 4M of memory, regognizes landscape text and six foreign languages
MSF-400G (Microtek)	scans continuous tone and line art, resolution of 400 dpi, eight bits per pixel, 14 levels of contrast, 14 levels of brightness, 12 half-tone patterns, 256 shades of gray

Table 3 : Some Scanners and their Characteristics.
(From 'Datapro Reports on Microcomputers', McGraw-Hill, 1989).

Figure 2.20 Color Single Frame Displays

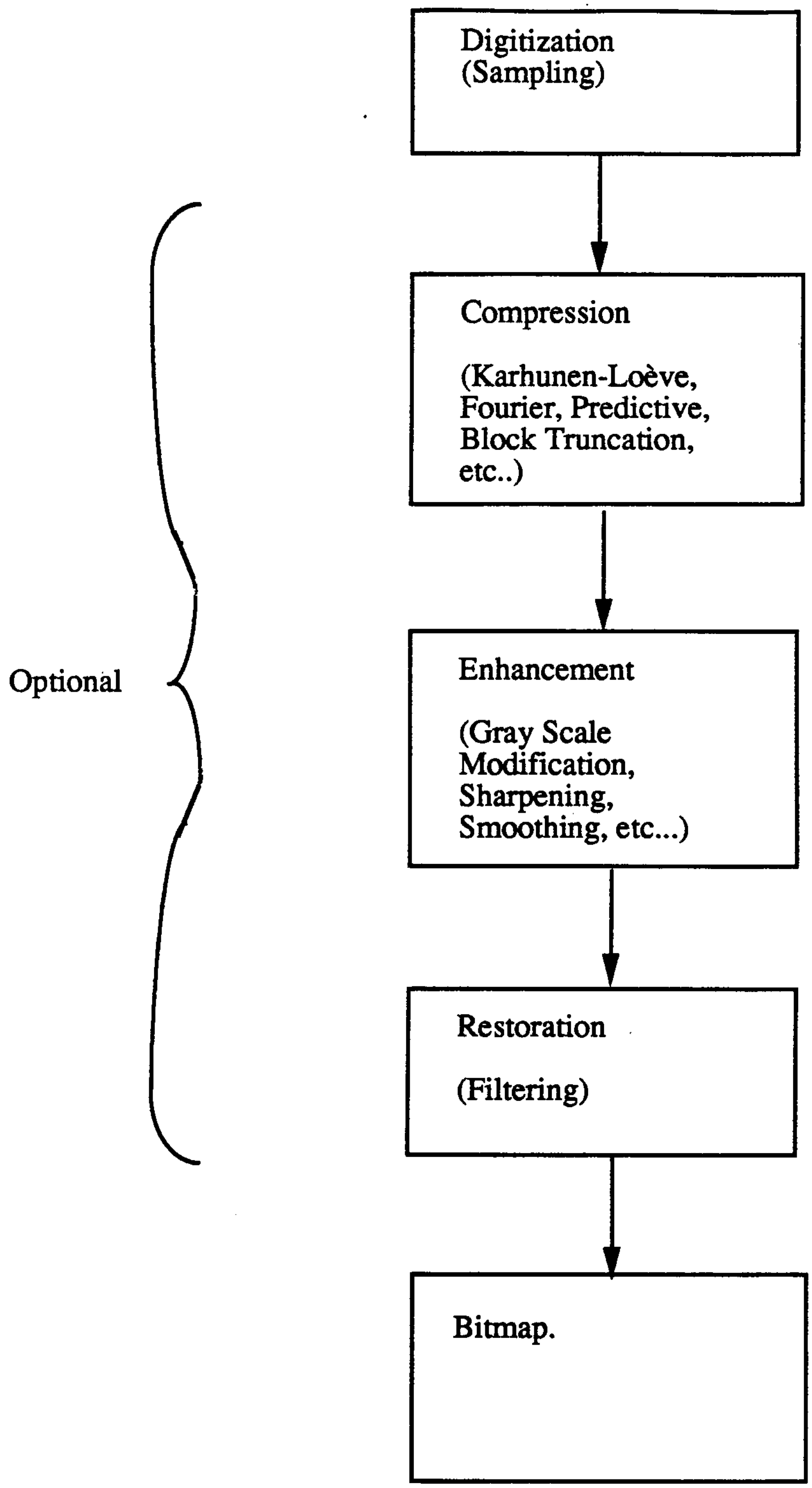


Figure 2.16.a : Sampling and Reconstitution - Analog to Digital.

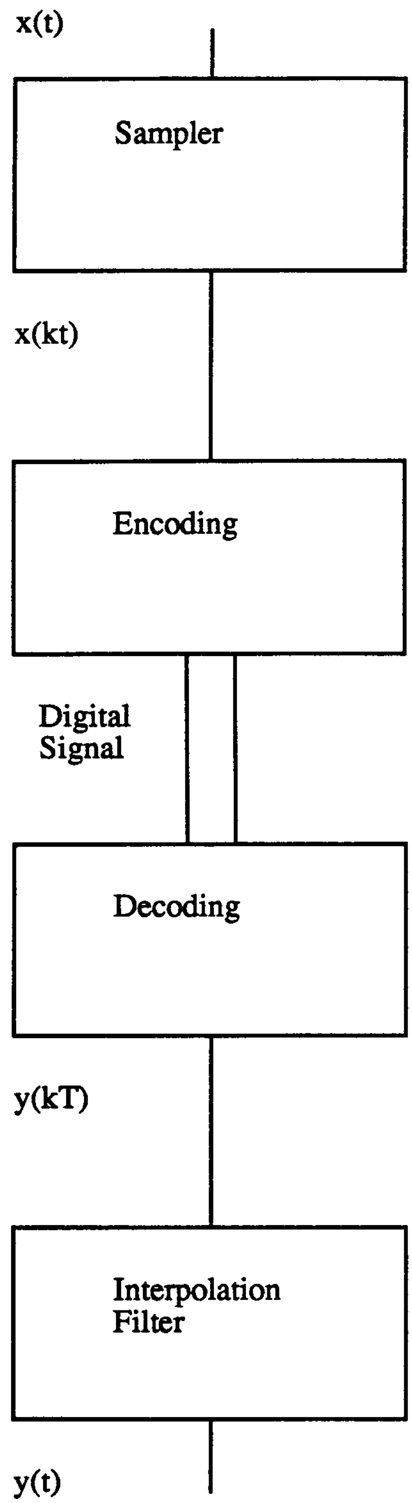
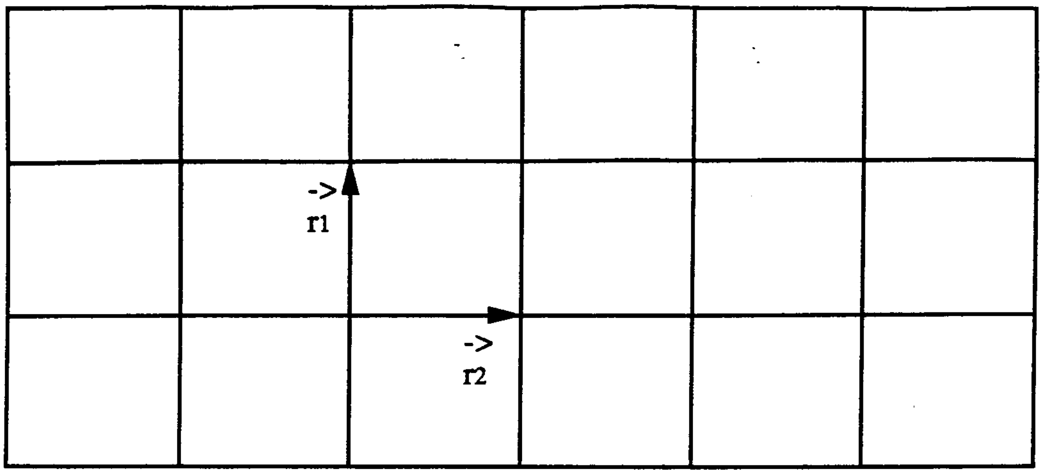


Figure 2.16.b : Sampling and Reconstitution - Analog to Analog.



Picture plane

$$F(\vec{w}) = \frac{G(\vec{w})}{Q} \sum_p \sum_q F(\vec{w} - \vec{w}_{pq})$$

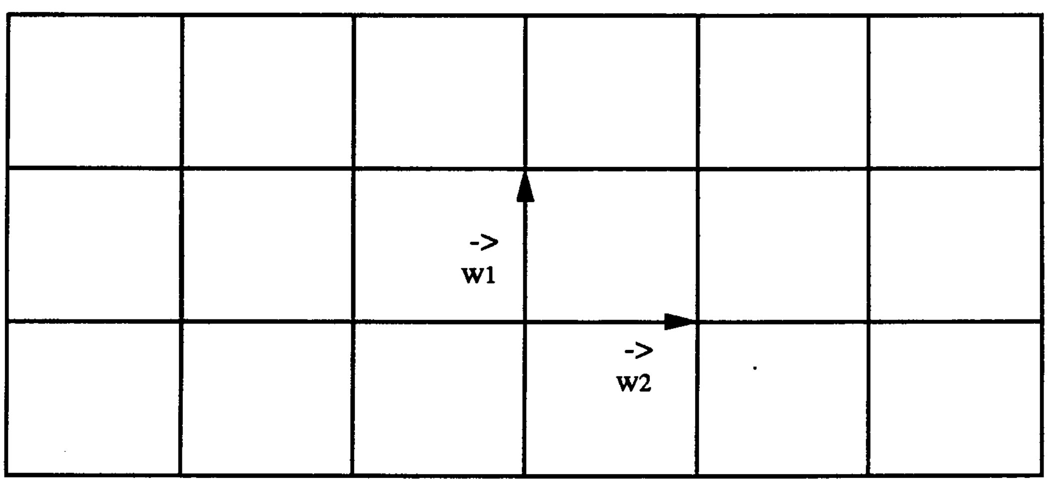
Q = area of the parallelogram formed by $\vec{r}_1$ and $\vec{r}_2$

$G(\vec{w})$ = Fourier Transform of the interpolation function $g(\vec{r})$

$\vec{r}$: position vector in the Picture Plane

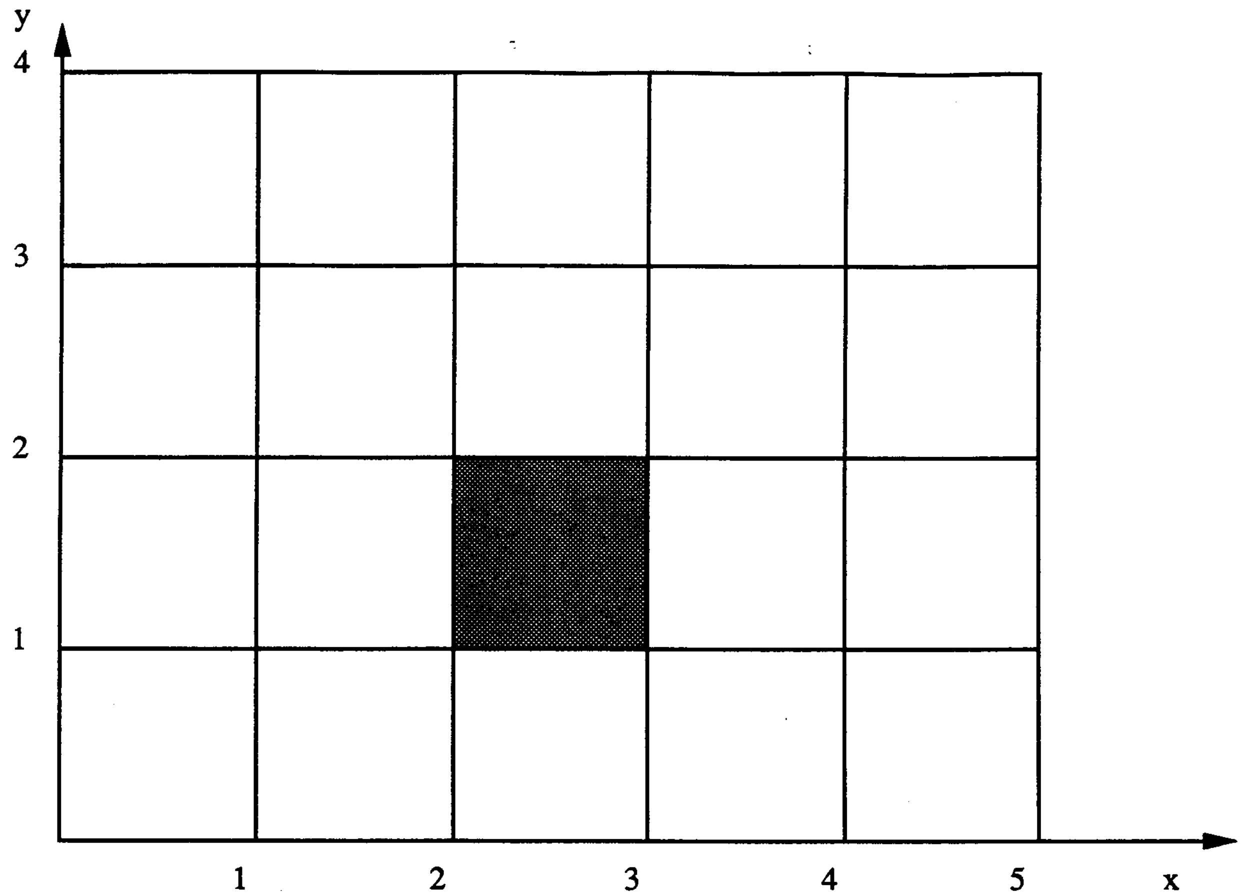
$\vec{w}$: position vector in the Spatial Frequency Plane.

$f(\vec{r})$ may be reconstituted everywhere from its sample values over the lattice of points $(m\vec{r}_1 + n\vec{r}_2)$ if $\vec{r}_1$ and $\vec{r}_2$ are small enough that $F(\vec{w})$ does not overlap with its images on the periodic lattice of points $(p\vec{w}_1 + q\vec{w}_2)$.



Spatial Frequency
Plane.

Figure 2.14.a : Image Mapping to Digital - Sampling Using an Array of Points.



Function $\varnothing_{2,1}(x,y)$ for M = 5, N = 6.

The function f(x,y) in the plane P is approximated by a sum of the form

$$\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn} \varnothing_{mn}(x,y)$$

where the functions $\varnothing_{mn}$ are an orthonormal and
where the constants a_{mn} that minimize the mean-square error are

$$a_{mn} = \int\int f(x,y)\, \varnothing^*_{mn}(x,y)\, dx\, dy.$$

For Standard Sampling

$$\varnothing_{mn}(x,y) = \sqrt{MN / AB} \quad \text{for } mA/M \leq x < (m+1)A/M \text{ and } nB/N \leq y < (n+1)B/N$$

$$= 0 \quad \text{otherwise}$$

where A and B are the sides of the rectangle within which the pictures are defined

$$a_{mn} = \sqrt{MN/AB} \int_{nB/N}^{(n+1)B/N} \int_{mA/M}^{(m+1)A/M} f(x,y)\, dx\, dy$$

Figure 2.14.b : Image Mapping to Digital - Sampling Using Orthonormal Functions (Standard Sampling).

2.3 Video Images

As we have discussed with the set of still images, the key factor in developing a means to display images is finding an acceptable tradeoff between the ability of the human eyes to respond and the limited processing and storage capability of the computer. In the world of video images, there again is the resolving capability of the human eye, combined with the issue of the field of view of the image and the update rate of the images to meet the eyes ability to comprehend motion. Thus the selection of the size of the standard television set in terms of its aspect ration was based upon significant testing of the viewers ability to view information in that form. The update rate of Television is 30 frames a second, a rate that matches the lower limit of the human response curve for recognizing image motion.

2.3.1 Means of Characterization

The video image is more than just the sequencing of a set of still images. There are several psychological and physiological factors that relate to the scan rates and interlacing of images to provide and adequate video image. In this section, we will focus on the scanning input to the video display and shall also discuss the scanned output of the display.

The video displays follow techniques that extend those of the CRT displays that we developed in Section 2.2. Specifically, we scan the screen in frames, in the US and other countries using NTSC video, this is 525 lines of scans per frame and 30 frames per second. A frame is composed of two fields, the first field being 262.5 lines and the second field being a same number of lines but these lines spaced between the first field lines. This is shown in Figure 2.x. The two fields compose a single frame.

Figure 2.21 Video Scanning Techniques (NTSC, SECAM, PAL)

The signal that is generated at the transmitter is a combination of the red, blue and green values of the image that is scanned at the original location. This is not the signal that is transmitted. What is transmitted is a set of three video signals that is a linear transformation of the RGB signal set. This composite transformation set is generated so as to make the received signal backward compatible with the standard black and white sets that were in existence in the United States at the time of the introduction of color TV. In addition the signals are required to fit into the same bandwidth as the black and white signals.

If we let R, G, and B be the red green and blue signals, then in NTSC video we generate a signal set that is composed of the following three signals;

$$Y=0.3R+0.59G+0.11B$$

$$I=0.6R-0.28G-0.32B$$

$$Q=0.21R-0.52G+0.31B$$

These signals are called the brightness or luminance signal, the inphase signal and the quadrature signal. It is found that with this transformation and the scan rates selected for capture and ensuing reproduction of the signal, that Y occupies 4.2 MHz of bandwidth and if processed by itself provides an adequate black and white signal. I occupies 1.5 MHz of bandwidth and carries the orange/cyan mix of colors in its signal. The Q signal is 0.6 MHz in bandwidth and carries the green/purple colors of the signal.

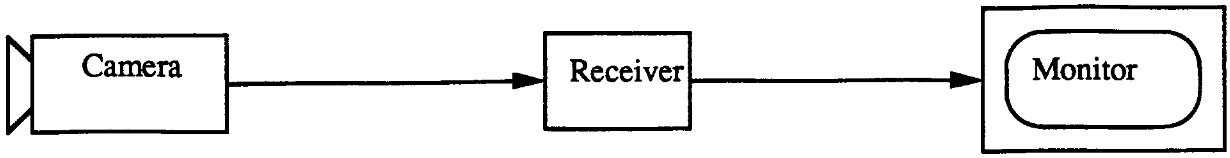
It is actually possible to display these signals by themselves, that is the I or Q signals and see what the resulting image appears like.

The composite video signal of Y,I and Q are carried in a complex fashion. The Y signal occupies the lower end of the 6 MHz bandwidth spectrum and is not encumbered by any other signal. The I and Q signals are carried at a frequency that is 3.58MHz up from the carrier. The I signal is sent using a sine modulation and the Q signal using a cosine modulation. Coherent demodulation is used at the receiver so that the two signals may be separated. At the very upper limit of the band, or the carrier plus 4.5 MHz is located a small FM voice carrier that is used to include the voice signal. All of the video signals are carried in an AM format.

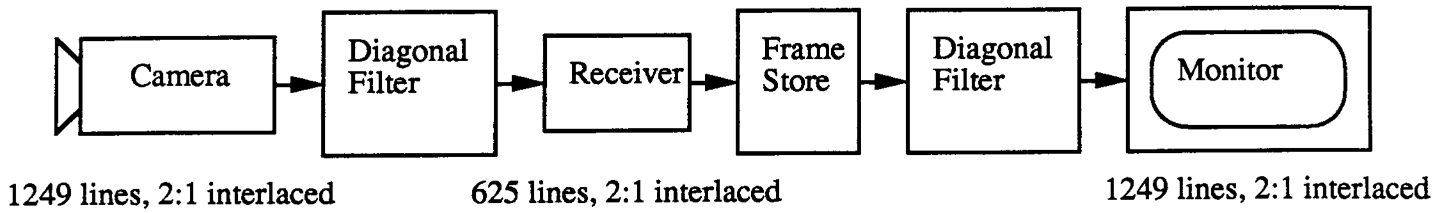
At the receiver, the signals are recovered and the Y, I and Q signals are placed through a transformation to recover the R,G,a and B signals again. These are then used to excite the three phosphors on the television screen.

The High Definition TV alternatives are presented in Figure 2.x We see that these standards require twice the scan rate and that these scan rates may be of various forms.

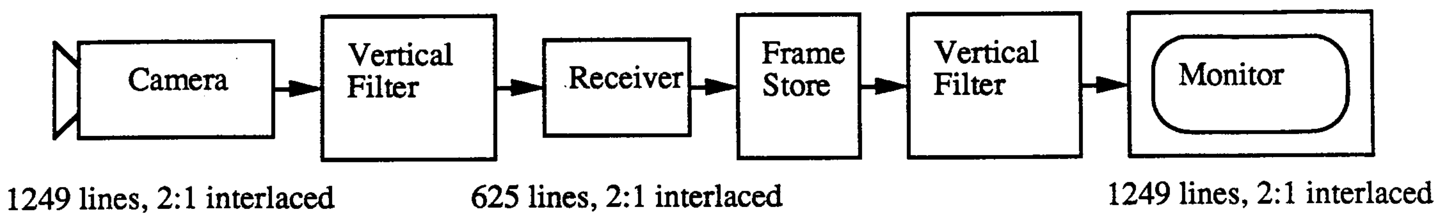
Figure 2.22 HDTV Scanning Alternatives



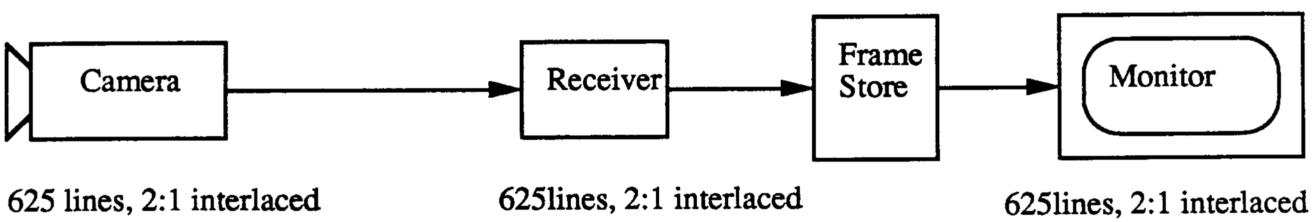
1 : HDTV System with 20 MHz bandwidth, 1249 lines, 2:1 interlaed, 50 fields/s, 4:3 aspect ratio



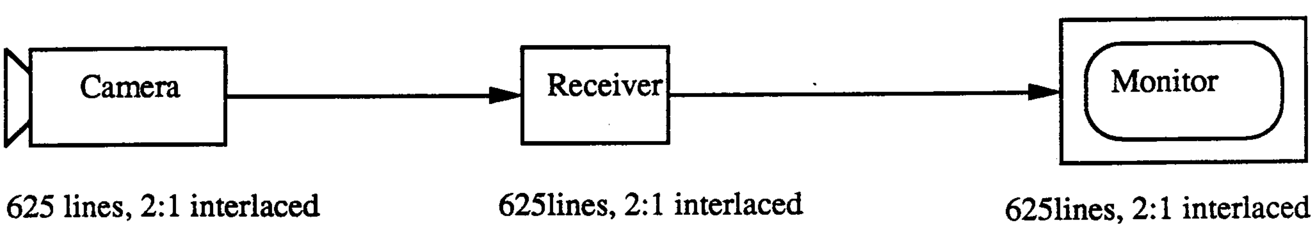
2 : Diagonal pre- and post-filtering with progressive scan reproduction.



3 : Vertical pre- and post-filtering with progressive scan reproduction



4 : Progressive scan reproduction of standard TV signal.



5 : Standard TV (5 Mhz, 625 lines, 2:1 interlaced, 50 fields/s).

Figure 2.7.a : HDTV Scan Alternatives - TV, EDTV and HDTV Alternatives.

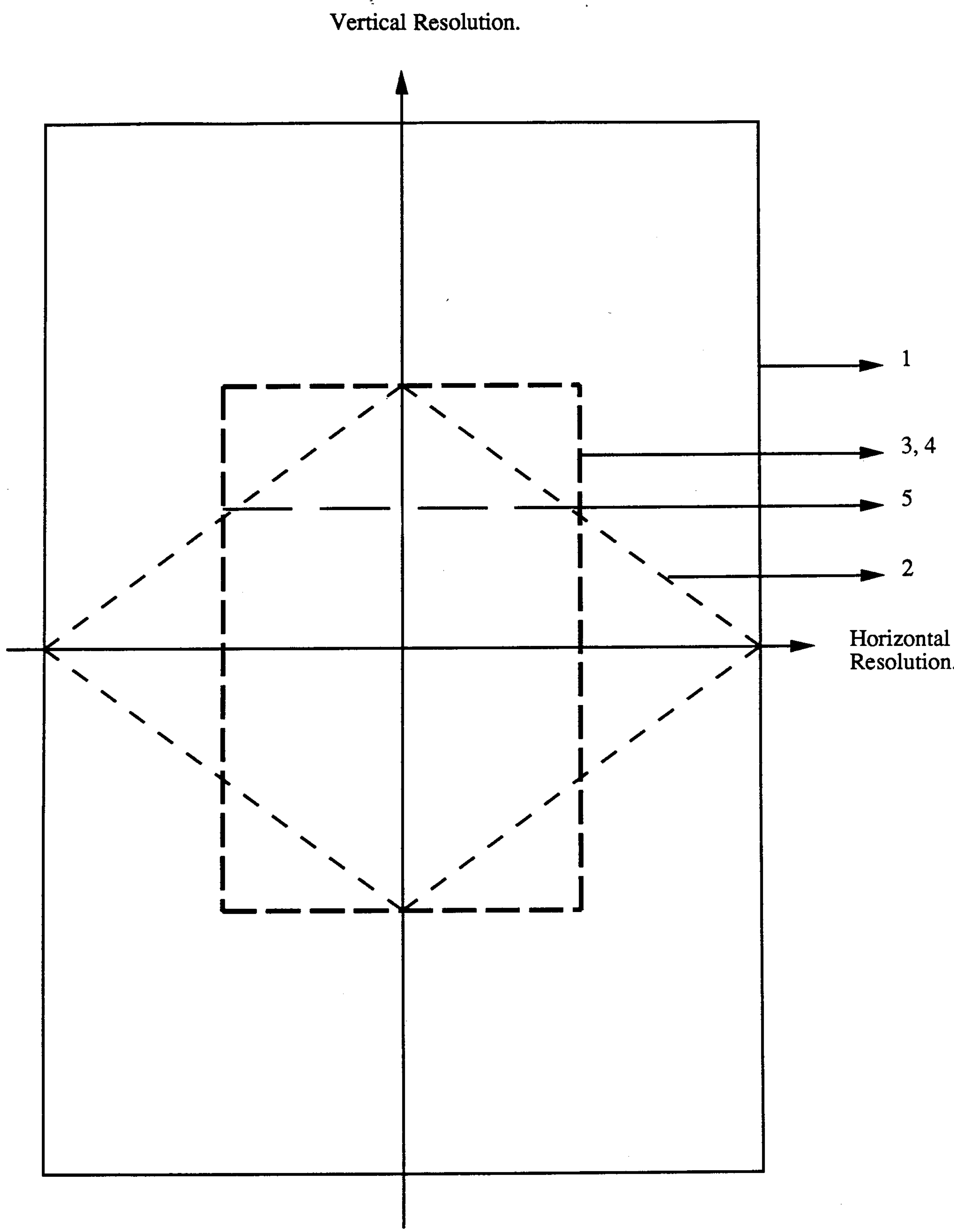


Figure 2.7.b : HDTV Scan Alternatives - Comparison of Resolutions for the Schemes in a/.

2.3.2 Means of Storage

Storage of video information is a significantly greater challenge than that of the still image. The still image storage is for a single or set of single images, whereas the storage requirements for video are based upon a continuum of many such frames at the scan rate of the device. In this section, we shall first characterize and size the storage requirement and then provide a set of alternative storage systems.

Figure 2.23 High Density Video Storage Systems

2.3.3 Means of Reconstitution

The means of reconstituting a video image is usually performed on a video display. In this section we hall concentrate on the display architecture and show how we can implement many of the schemes that we have discussed in Section 2.3.1.

(1) Nyquist Theorem

(2) Bandwidth Example

(3) Modulation Effects Bits/Sec / Hz

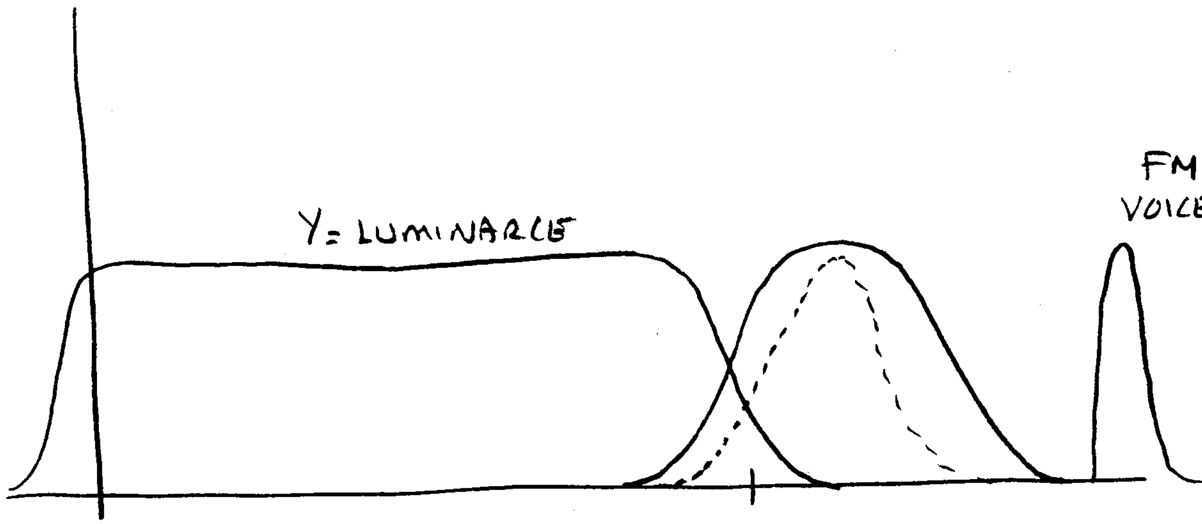
Figure 2.23 Video Displays

Figure 2.24 Video Sampling and Digitization

IBM Graphics Standards	Characteristics
MDA (Monochrome Display Adaptor)	text only, 80 columns x 25 lines, operates at 15.75 kHz
CGA (Color Graphics Adapter)	text and graphics, resolution of 320x200 pixels, four simultaneous colors, digital signal at 18.5 kHz
EGA (Enhanced Graphics Adapter)	text and graphics, resoltion of 640x350 pixels, sixteen simultaneous colors, digital signal at 21.85kHz
PGA (Professional Graphics Adapter)	text and graphics, resolution of 640x480 pixels, 256 simulatneous colors, analog signals
VGA (Video Graphics Array)	text and graphics, resolution of 640x480 pixels (1,024x768 with a Display Adapter 8514/A). 256 colors (32 intensity levels of eight colors : red, green, blue, cyan, magenta, white, black and brown) or 64 shades of gray, color gradation, analog signal.

Table 1 : IBM Graphics Standards and Their Characteristics.
(From 'Datapro Reports on Microcomputers'
McGraw-Hill, 1989).

Figure 2.25 Video Bandwidth



2.4 Voice

The human voice is a complex information transmission medium that is created with the interworkings of the human brain, overall nervous system and the vocal chords of the human body. Combined with the creation of speech is its reception by the human ear which has its own characteristics.

2.4.1 Means of Characterization

The voice is the second of the sense that we find in a multimedia system and its characterization can take on several dimensions. First, we can look at the signal itself, that is the conversion of the sound pressure wave into and electrical wave, and develop the digitization of that signal. From that digitization we can determine the bandwidth necessary for adequate transmission and size the storage requirements. A second approach is to recognize the speech using many of the speech recognition systems and then convert the speech into a new form, the written word. With the developments in speech recognition it is now possible to do this type of characterization for many applications.

Let us first develop an understanding of the generation and reception of speech in the human. Figure 2.x depicts the generation system of the human which is a combination of the neck and head in conjunction with the functioning of the lungs in the expelling of air. The basis elements of the speech generation path are:

o Oral cavity including the tongue and mouth, within the confines of the lips and down to the velum at the rear of the mouth.

o Nasal Cavity from the nostrils at the front to the pharynx at the ear.

o Oral pharynx from the velum down to the epiglottis,

o Nasal Pahynx at the rear of the nasal cavity and above the oral pharynx.

o Laryngeal pharynx region from the velum down to the vocal chords.

Each of these areas are resonant regions that are used for the generation of particular types of sounds. The vocal chords are used in particular in the generation of speech by passing the air from the lungs through the chords and by exerting various modulations o the chord openings by the local chord muscles. This air can then be made to resonate at different modes in the various cavities and the resulting sounds are speech. If we were to block any one of the resonant cavities, all other elements working properly, we would seriously affect the sound. This is a common effect when a person has a cold.

Figure 2.26 Human Speech Generation System(Head and Neck, Ref Parsons p. 63)

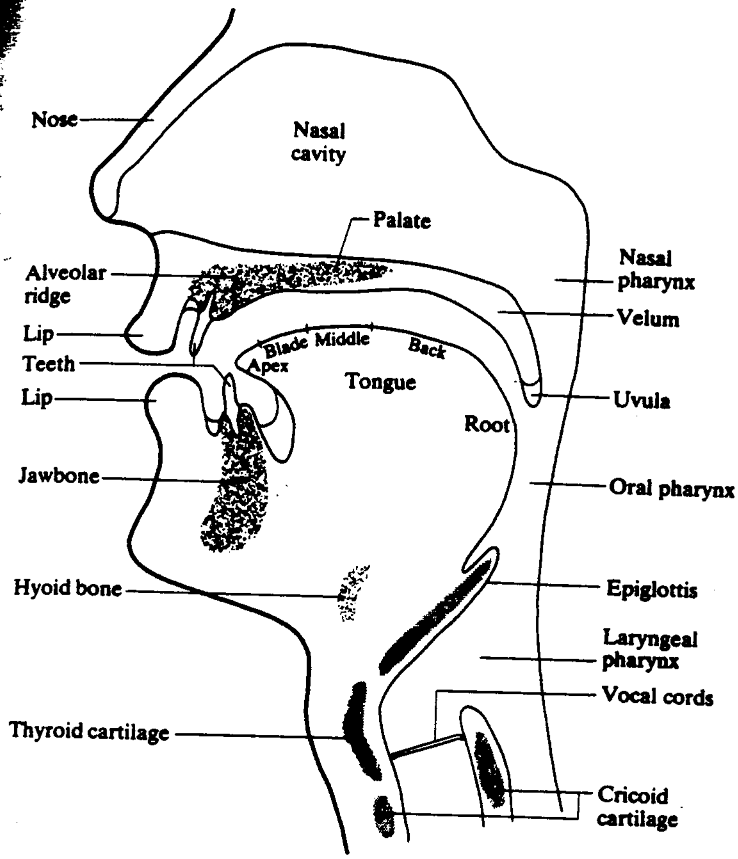


Figure 3-3 Sagittal section of the human head, showing the principal organs of speech.

The roof of the mouth can be divided into two principal regions. In front, the roof is formed by a bone called the palate which separates the mouth from the nasal cavities and supports the upper teeth. At the back of the palate, the roof is formed of muscle and connective tissue; this structure is called the velum, or soft palate. (Writers who call the velum the soft palate refer to the palate as the hard palate to distinguish the two.) The uvula is a small fleshy appendage at the rear of the velum. The velum can be lifted by a muscle and pressed against the back wall of the pharynx to seal the nasal passages off from the rest of the vocal tract. At the front of the palate there is a ridge, formed by the thickening of the bone where the front teeth are inserted; this is called the alveolar ridge.

The tongue is a large system of muscles connected in front to the lower jaw and in back to bones in the throat and head. Its biological functions include tasting and manipulation of food during mastication. For articulatory purposes it is convenient to divide the tongue into regions. In the absence of distinct landmarks, it is difficult to define these regions precisely, but they are approximately

Figure 2.27 depicts the human ear and the organ the receives the speech produced either by another person or even themselves. The ear has three major parts; the outer ear, the middle ear and the inner ear.

The outer ear extends from the pinna or outer ear surface to the eardrum through the meatus. This is itself a resonant cavity for the propagation of the sound entering the human ear. The middle ear is compose of the ear drum, the bone structure called the ossicles and the entry into the Eustachian tube. The ossicles contains three bones called the hammer, anvil and stirrup. These bones are used for the transmission and conduction of the sound responses into the nerve endings that are key to the transduction of sound into the overall nervous system.

The inner ear contains the cochlea which converts the mechanical vibrations into electrical impulses in the hearing part of the central nervous system.

The human ear can hear sounds from the range of 16 Hz to 16 KHz with some exceptions up to 20KHz. The sound pressure limits range from 0 dB or 0.0002 dynes/cm2 to 130 dB. At the high level the human ear generates pain and is damaged. At the lower level there is no perceptible response.

Figure 2.27 Human Ear and Speech Reception (Ref Parsons p.67) ɩns will be discussed in greater

ɩther point, the air flow past the
ɩd noise whose frequency spec-
he constriction. Sounds so pro-
ɩn can occur with or without

ɩff at any point while the talker
ɩd upon release (i.e., when the
ɩr. The combination of a short
ɩteristic sound. If the release is
ɩ, [p], [t]); if gradual and tur-
e.g., tch or German pf) and is

ɩn the vocal cords, vibrations
ɩvula (French or Prussian r),

ttal output. Linguists think
m into vowels and conson-
er from a physiological
ɩsed on the sound; (2) from
ɔ to the signal emanating
ɩ of *articulatory phonetics*:
any given speech sound.
ɩics: what the measurable
d how acoustical features

ɩnetics in detail later, but
ɩe sound is modulated by
to change the quality of
ɩons on the voice.
ɩe operation of *filtering*.
he vocal tract, like any
tion of its shape. These
single most important
vowels and some of the
lly important informa-
ɩpes of modulation are
wideband noise bursts

3-3 HEARING AND PERCEPTION

By hearing we mean the process by which sound is received and converted into nerve impulses; by perception we mean, approximately, the postprocessing within the brain by which the sounds heard are interpreted and given meaning.

Hearing

We will start with the anatomy of the ear. The ear is divided into three parts: the outer ear, the middle ear, and the inner ear (Fig. 3-8).

The outer ear consists of the *pinna* (the visible, convoluted cartilage), the external canal (*external auditory meatus*), and the eardrum (*tympanic membrane*). The pinna protects the opening; its convoluted shape is thought to provide some directional cues (Schroeder, 1975). The external auditory meatus is a nearly uniform tube approximately 2.7 cm long by 0.7 cm across through which the sound passes to reach the eardrum. Like all tubes, it has a number of resonant frequencies, of which only one, at approximately 3 kHz, falls in the frequency range of speech. The tympanic membrane is a stiff, conical structure at the end of the meatus. It vibrates in response to the sound and is the first link in a chain of structures which transmit the sound to the neural transducers in the inner ear.

The middle ear is an air-filled cavity separated from the outer ear by the tympanic membrane and connected to the inner ear by two apertures called the *oval* and *round windows*. The middle ear is also connected to the outside world by way of the *eustachian tube*, which permits equalization of air pressure between the middle ear and the surrounding atmosphere.

The middle ear contains three tiny bones or *ossicles* which provide the acoustical coupling between the tympanic membrane and the oval window. These bones are called the *malleus* (hammer), *incus* (anvil), and *stapes* (stirrup). The malleus is attached to the tympanic membrane, the stapes to the oval window, and the incus connects the two. The function of the ossicles is twofold: (1) impedance transformation and (2) amplitude limiting.

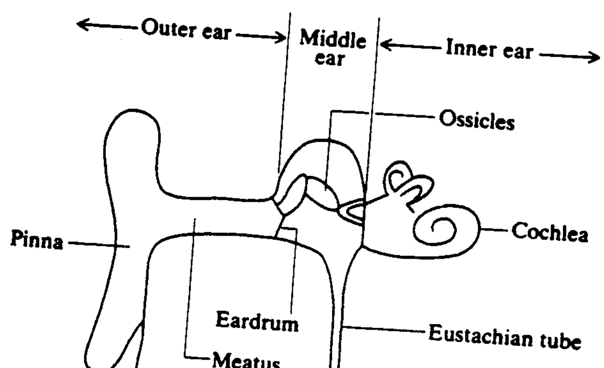


Figure 3-8 Sectional view of the human ear.

The signal generated by the human voice system generates a pressure wave of the form:

p(x,y,z,t)

which is spatially and temporally dependent. The wave is received by the ear and is processed into recognizable speech. We frequently can take this pressure wave and receive it by an electrical tansducer like a microphone and convert it into an electrical signal. The resulting signal is a representation of the temporal characteristics of the speech. It does not provide full spatial characteristics and in fact involves the impact of the directionality of the receiving transducer.

The typical speech temporal profile is depicted in Figure 2.x. As we have already noted, speech is constrained in the 16 Hz to 16Khz range so that the speech signal:

s(t)

is composed of only those specific frequencies.

Figure 2.28 Voice Signal Profile

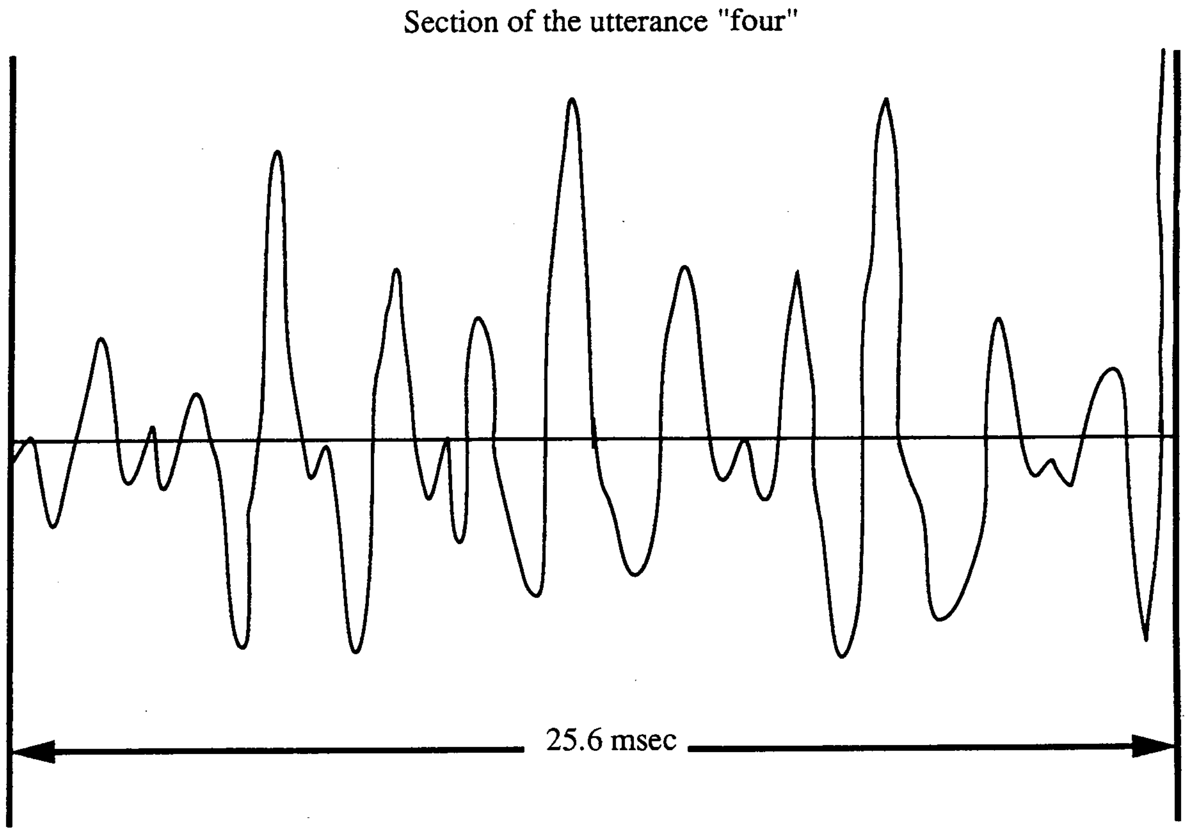


Figure 2.9: Voice Signal Profile

If we define s(t) as the speech signal, we say that this signal is a random process since it changes in time in a random fashion. We can determine the correlation function of this process by taking its average as it is mixed against a shifted version of the signal. We define R(T) as the correlation function of the signal. Specifically if we let E[ ] be the averaging operator on the random process, then we have:

$$R(T) = E[s(t)\ s(t+T)]$$

where we have assumed that the process is stationary, that is R is independent of t.

The spectrum of the signal is defined as the following:

$$S(f) = \quad R(T)\ \exp(-j2\quad fT)\ dT$$

The spectrum for a typical voice signal is sown in Figure 2.x

We can now use the Nyquist sampling theorem that states that if we sample a signal at twice the maximum frequency rate, then we can totally reconstruct the signal. Thus if a signal is from 0 to 10KHz, we must sample at 20,000 per second. For voice this means for full fidelity we must sample at 32,000 per second. Generally this is not the case and we frequently limit the upper voice signal in telephony rates to 4 KHz and the sample rate is 8,000 per second.

The sampled signal now can be given by:

s(kT0) where T0 equals:

$T0 = 1/8,000$

and s(kT0) is the sample value. We can now quantize the samples to create a digital sample. If we quantize s into two levels all we need is on bit (eg 0 or 1). If we quantize into four levels we can do that with two bits. If we continue, we can quantize into 256 levels and do this with 8 bits. This 8 bit quantization has been shown to give minimum human distortion in terms of the human response.

Thus typical speech is sampled at 8,000 times per second and at 8 bits per sample. This means that voice is captured at the rate of 64,000 bits per second. High fidelity speech may be at the 32,000 sample rate ad may even use 10 bit quantization. Thus for high fidelity speech we have 320,000 bits per second.

Figure 2.29 Voice Spectrum

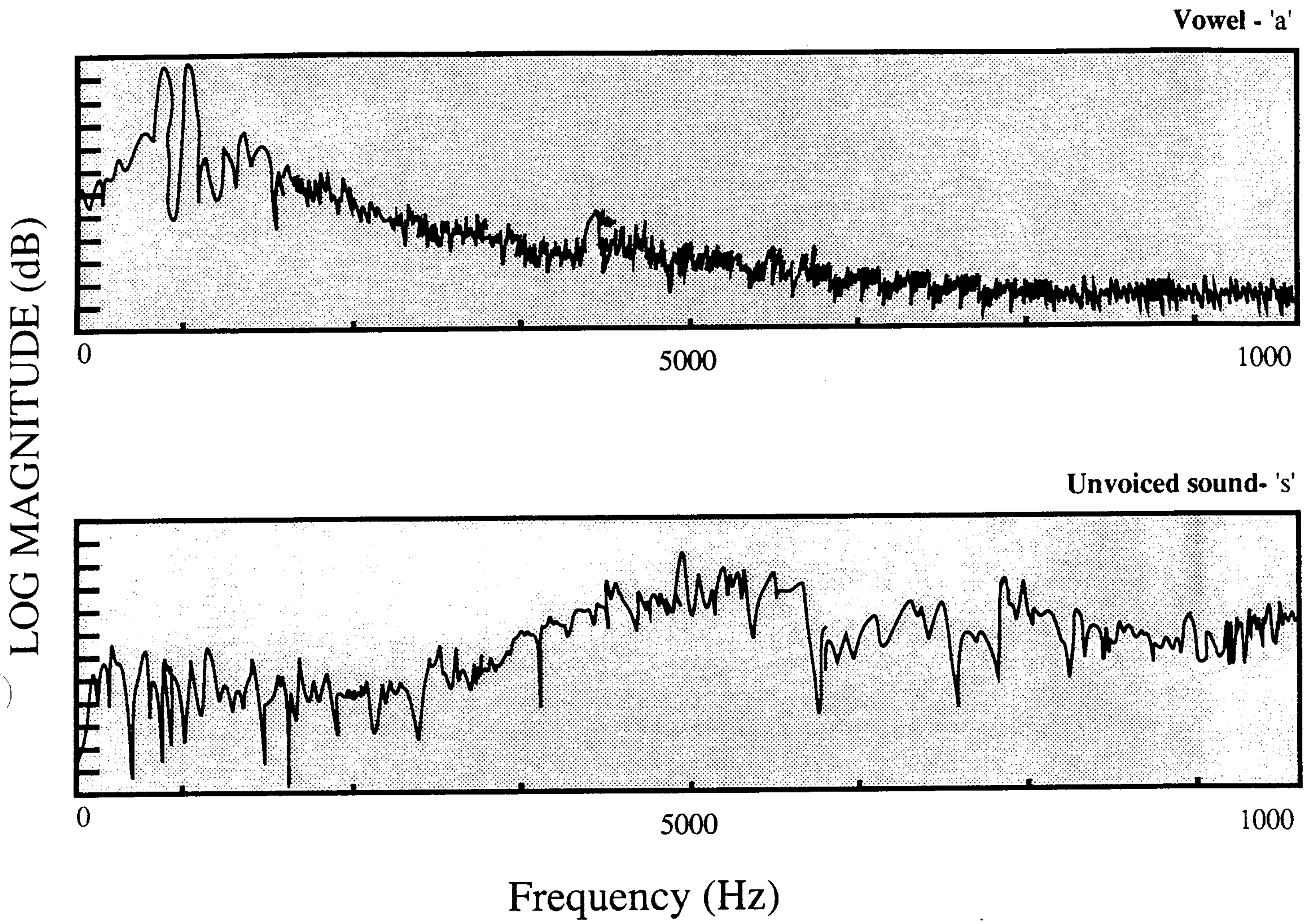


Figure 2.10 Voice Spectrum (Ref —)

Limits of Human Hearing

Limits of Technology Effectiveness

Limits of Human Hearing
Audibility Limit between 16 Hz and 20,000 Hz greatest sensitivity between 1000 Hz and 4000Hz.
Change in Frequency Detection Limit 0.2 - 0.4 %.
Perception of Tone Limit in Time 0.5ms minimum duration.
Resolution of sound between 200 Hz and 2000 Hz.

Table ~~Figure~~ 6 : Limits of Human Hearing.
(From 'McGraw-Hill Encyclopedia of
Science and Technology' , vol
McGraw-Hill, 1987).

We show the details of the voice sampling process in Figure 2.x. Let us review this in further detail.

o First we have the general signal s(t) which is the voice signal.

o Second, we take and sample the signal at the sample points. The sample pints are twice the maximum frequency of the spectral density. The corresponding sample time intervals are defined as T sec. For 4 Khz bandwidth this is 1/8,000 sec or 125 micro second sampling.

o Third, enumerate the sequence of samples;

$$\{s(kT)\} : \quad k = 1, \ldots\ldots, N$$

as the sampled signal.

o Fourth, using a quantization scheme, select the number of quantization levels, using an a priori choice of through an optimizing process, but choose the level to be consisted with a binary format. That is the quantizing levels should be some factor of 2 to a power. Let us assume that there are 256 levels or 8 bits of quantization.

o Fifth, choose the quantizing levels either a priori or through an optimizing scheme. The levels are defined as follows. Let Lk be the kth level and let the total level be;

$$\{Lk\} \quad : k=1,\ldots,2N$$

where N is the quantization bits.

Then we have the quantization mapping of :

s(kT) is in Ln if   Ln< s(kT) < Ln+1

o Sixth, then map s(kT) into bit pattern Bn which is the binary level of Ln.

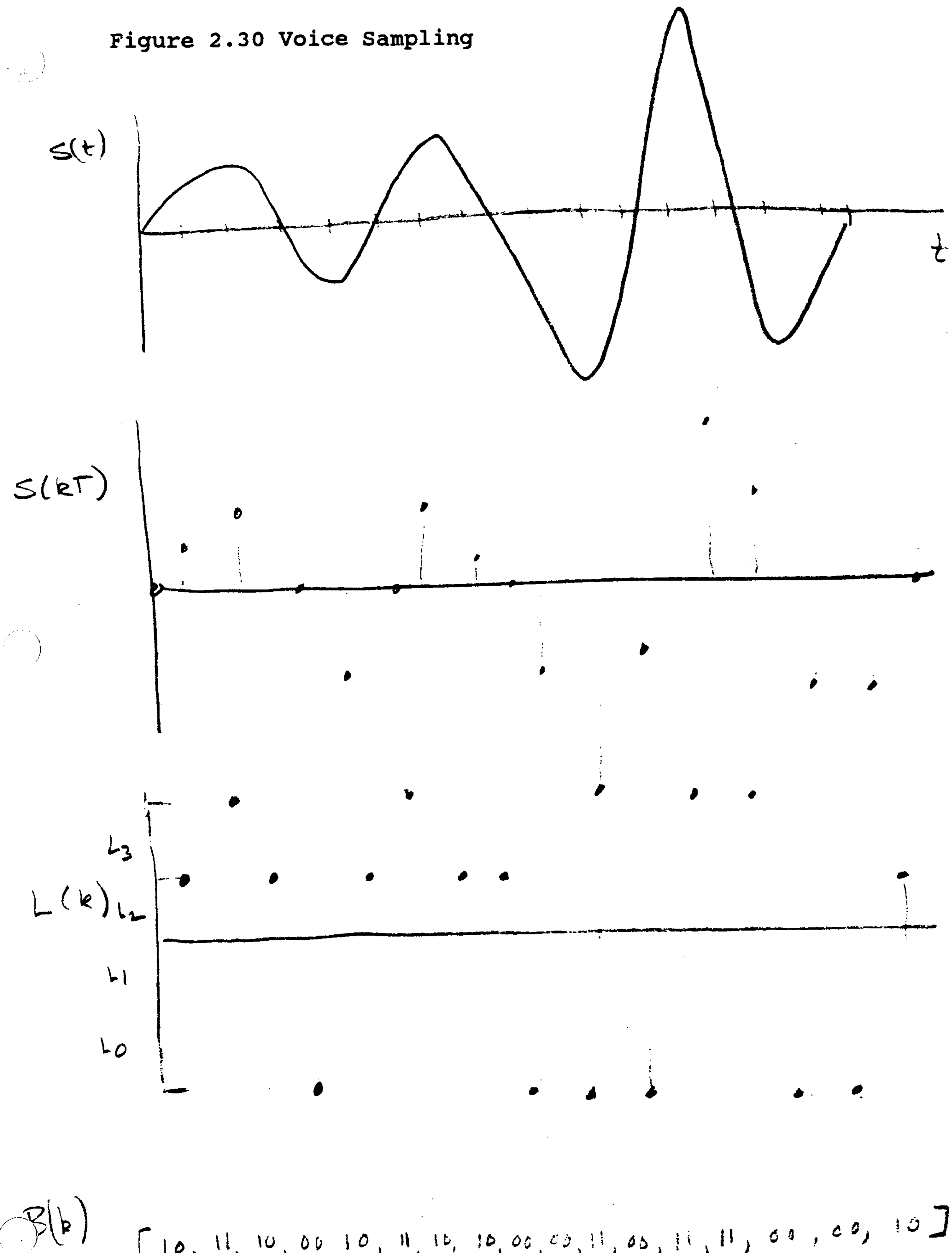
o Seventh, create a data stream of the form:

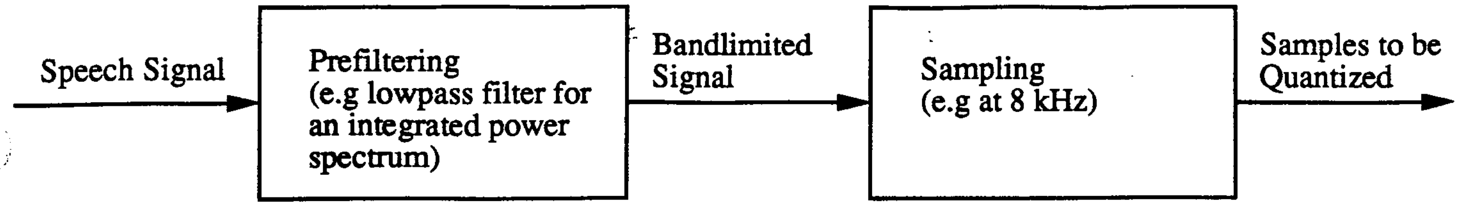
[B1,B2,.......,Bn,Bn+1,.....]

where Bi is the specific bit stream:

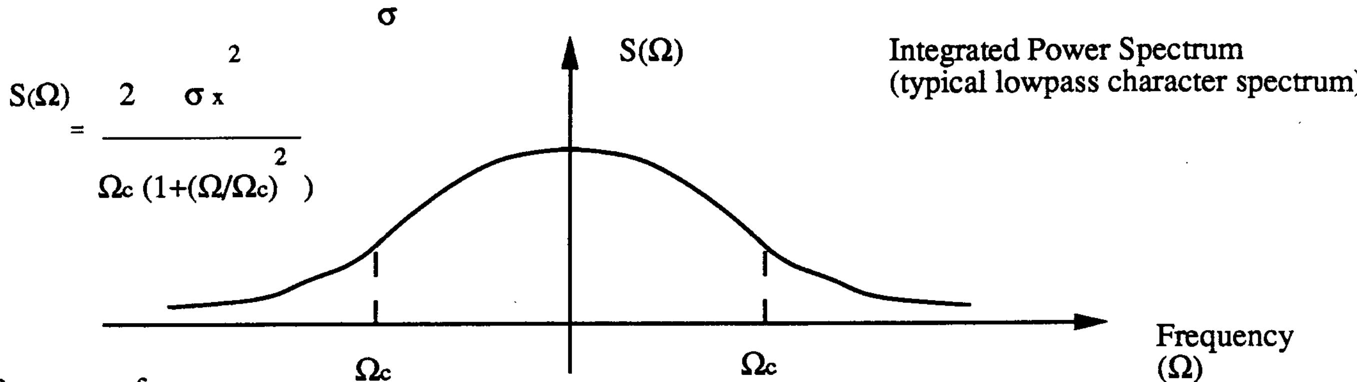
Bi = [0,1,0,1,1,...,0]

Figure 2.30 Voice Sampling

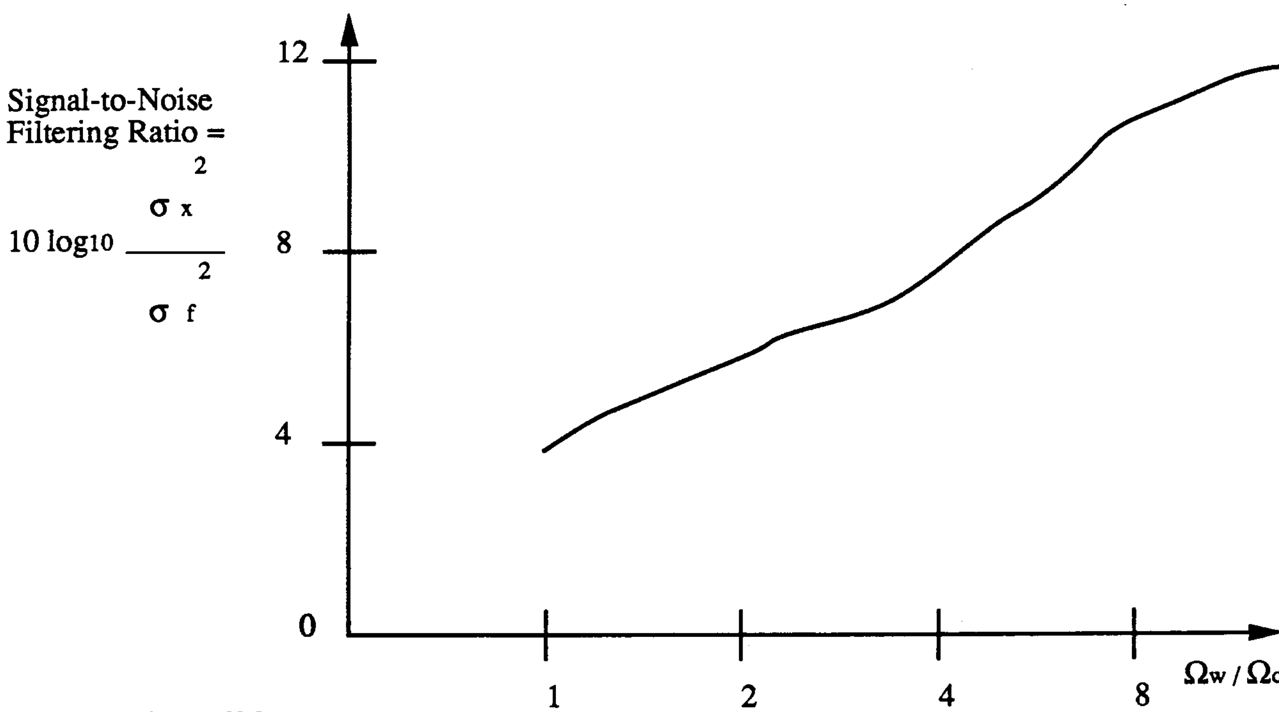




a : Voice Sampling Scheme.



Ω_c : corner frequency.



Ω_w : nominal cutoff frequency

σ_f^2 : prefiltering error variance.

b : Effect of Prfiltering Speech for an Integrated Power Spectrum.

2.19 : Voice Sampling.

2.4.2 Means of Storage

Voice storage systems are generally well developed and the major factors in such systems are the ease of access and the ability to store large volumes of speech while still retaining the quality of the original.

Figure 2.31 Voice Storage Systems (Ref Voicetek)

2.4.3 Means of Reconstitution

Speech regeneration can be done in one of two ways. As we discussed earlier in the section, we can take the digitized version of the speech and regenerate it directly by reversing the process. A second approach of speech generation from the written word, this is called speech synthesis. The development of speech synthesis systems has been significant over the past several years and these now provide interesting alternatives for use in multimedia systems.

2.5 Graphics

Graphics is distinct from the other three representations. In still image, video and voice the information medium is naturally created and is generated from some form of human activity. In the graphics world, the information is created by a computer to represent some form of reality but is not a priori created by the natural world. Graphics development has progressed significantly over the past few years. The original systems were those that were developed to use a minimum of storage and a minimum of processing. In addition they were designed for vector oriented displays and not for fully bit mapped displays. In this section we shall discuss three of the graphics standards and show how they have evolved through what is now their present state.

The first standard that we shall discuss is the GKS, Graphics Kernel System, standard that was originated in the 1970, 's. It is the oldest standard and many of its characteristics reflect the limitations on the technology at the time it was developed. The standard uses the basic vector graphics approach and although it is designed for use on many machines, it does not press the capabilities of the processors to allow for maximum flexibility. Thus it has only two dimensional characteristics and requires significant detailed definition for a fully flexible design.

The second standard is the PHIGS, Programmer Hierarchical Interactive Graphics Standard, that is much more flexible and assumes an generally more capable processor. It however still relies on the vector graphics approach to display.

The third standard, although a defacto standard, is that of PostScript. This system was developed in conjunction with the work on the Apple MacKintosh system and takes full advantage of the bit mapped displays. It functions on smaller machines, utilizing the power that is available in the smaller processors, smaller in size but significantly greater in processing capability. In addition is recognizes the availability of the bit mapped display and the inherent advantages of the existing MAC display software and the introduction of laser printers.

2.5.1 Means of Characterization

Graphics characterization has evolved over the past ten years with significant growth in capability. The graphics software packages are now extensively used in engineering, architecture, and in many artistic applications to design both still and full motion depictions of real and imagined events. Unlike stills, video and voice, the relationship to the external reality for this media characterization is less strong. For example, we can think of a designer who is developing a new auto body and in so doing is using a graphics design package to help do so. The graphics package starts with the ideas that the designer has and follows the designers lead in developing the body design of the auto. In this case, the output is the reality and not the input.

Many standards have been developed for the development of graphics displays. These standards rang fro such as GKS, the Graphics Kernel System, to PHIGS (the Programmer's Hierarchical Interactive Graphics Standard), the CGM (Computer Graphics Metafile).

Graphics systems take a set of of desired images and reduce them to a defined set of primitive elements that in sum make up the resulting image. Let us consider a simple example. We may desire to develop the graphical representation of a computer terminal. We can first start with a set of rectangles that can represent the overall structure of the terminal, the key board and the display device. We can then add to these basis elements sets of rectangles, circles or ellipses that make up the keys, displays, logos etc that make up the final terminal. We can then apply further and further detail to the device to make up a more complex and detailed display. We may even use the capability of creating complex curved surface by using spline functions that are basically higher order polynomials defined on a limited set of R N.

The net result of a graphics system is a picture representation that can fall into one of two major categories, vector graphics or bit mapped. In the bit mapped systems of generation, the images are created by a fully random selection of bits and color associated with the pixels. That is we can imagine that the end user may be capable of entering in to the system the image on a bit by bit basis and doing so by selecting a pixel and then choosing the colors or intensities for the pixel. This would mean that for a fully bit mapped system,we would have to have N by N entries with the B bits chosen per entry. Thus there would be N*N*B steps involved. For the 2K monitors, this would entail almost 100 million steps per screen. Such an endeavor is beyond the capabilities of most humans.

The alternative is to select a system that has a set of basis functions such as circles, rectangles, ellipses, etc and that in addition is aided by a set of primitive commands that allows for the placing, orientation and coloring of the basis functions. The net result is a vector oriented system that does not require a bit by bit painting of the display device. In addition this system of basis functions and primitives allow more readily for the animation of the graphics images.

Graphics standards have several objectives in the development of their design. Typical sets of objectives that will be found in the three systems that we are to discuss are as follows:

o Machine Portability

o Application Program Independence

o Display Independence

o Hierarchical Designs

o Ease of Compilation

We shall see that some of these goals are not fully achieved in all of the approaches but that they are generally thought of in the implementations.

2.5.1.1 GKS

GKS, the Graphics Kernel System, is a graphics design tool that has the status of being the first graphics standard. It is composed of six general elements:

o Primitives
o Attributes
o Attribute Bundles
o Segments
o Viewing
o Input

Let us describe each of these in turn.

o Primitives: The primitives or output primitives, or output functions are those functions that are defined in GKS that are used for the generic characterization of the basic sets of elements that are to be drawn from in GKS. The primitives consist of the following set:

o Polyline: This allows for the drawing of a set of connected lines from any point to any other set of points.

o Polymarker: These are markers that are placed on the figures to specify particular locations on two dimensional surfaces.

o Text: This allows for the display of arbitrary strings of characters.

o Fill Area: This allows for the defining and filling of arbitrary boundaries of polygonal figures.

o Cell Array: This displays a grid of rectangular elements.

o Generalized Drawing Primitive: This primitive allows for the generation of arbitrary shapes of figures.

o Attributes: Attributes are specific characteristics than can be attributed to any of the primitives. The attributes are specified in the form:

SET FILL AREA COLOR INDEX

SET POLYMARKER COLORINDEX

SET TEXT ALLIGNMENT

For example the attribute SET LINEWIDH SCALE FACTOR affects the polyline line width and allows for arbitrary scaling.

o Attribute Bundles: These are collections of attributes that allow for the defining of specific primitive attributes out of the main source code. For example we can use the expression:

```
SET POLYLINE INDEX(1)
```

to set the polyline to the characteristics of index 1. Index one may be machine dependent and the applications programmer may not have to worry about the portability of the code on a machine dependent basis. Thus GKS allows for the assignment of attribute and primitive bundling through the index function.

o Segments: Collections of primitives and attributes, and even attribute bundles create segments, which are separate descriptive parts of the total graphics picture. For example if we are drawing a house, we can create a segment that is the roof of the house, composed of many polylines, and their attributes and many of the other primitives as necessary. The net result is the segment of the graphics representation.

o Viewing or Transformation Functions: This element of GKS allows for the definition and description of the viewing space of the GKS display. Typical viewing functions are:

```
SET VIEWPORT
SET WINDOW
SET WORKSTATION WINDOW
```

These functions are used to specify such functions as the location of the window coordinates on the screen.

o Input: This function provides for the operator input to the program and allows for the entry of such elements that can provide location, select numbers and text for input, and select or pick specific segments of graphics for display or inclusion. The input functions are in effect a command language for the operation of the GKS kernel.

The overall architecture of the GKS system is shown in Figure 2.x. It has three main layers. They are:

o Primitive Attributes that are used to characterize the primitives and create the individual segments of the GKS image.

o Segment Attributes that are used for the detail specification of the segments.

o Workstation Attributes that provide the details associated with the specific display device. These are typically given in terms of specific device drivers.

Figure 2.32 GKS Overall Architecture (Ref Bono et al. IEEE CGA 7/82 p. 15)

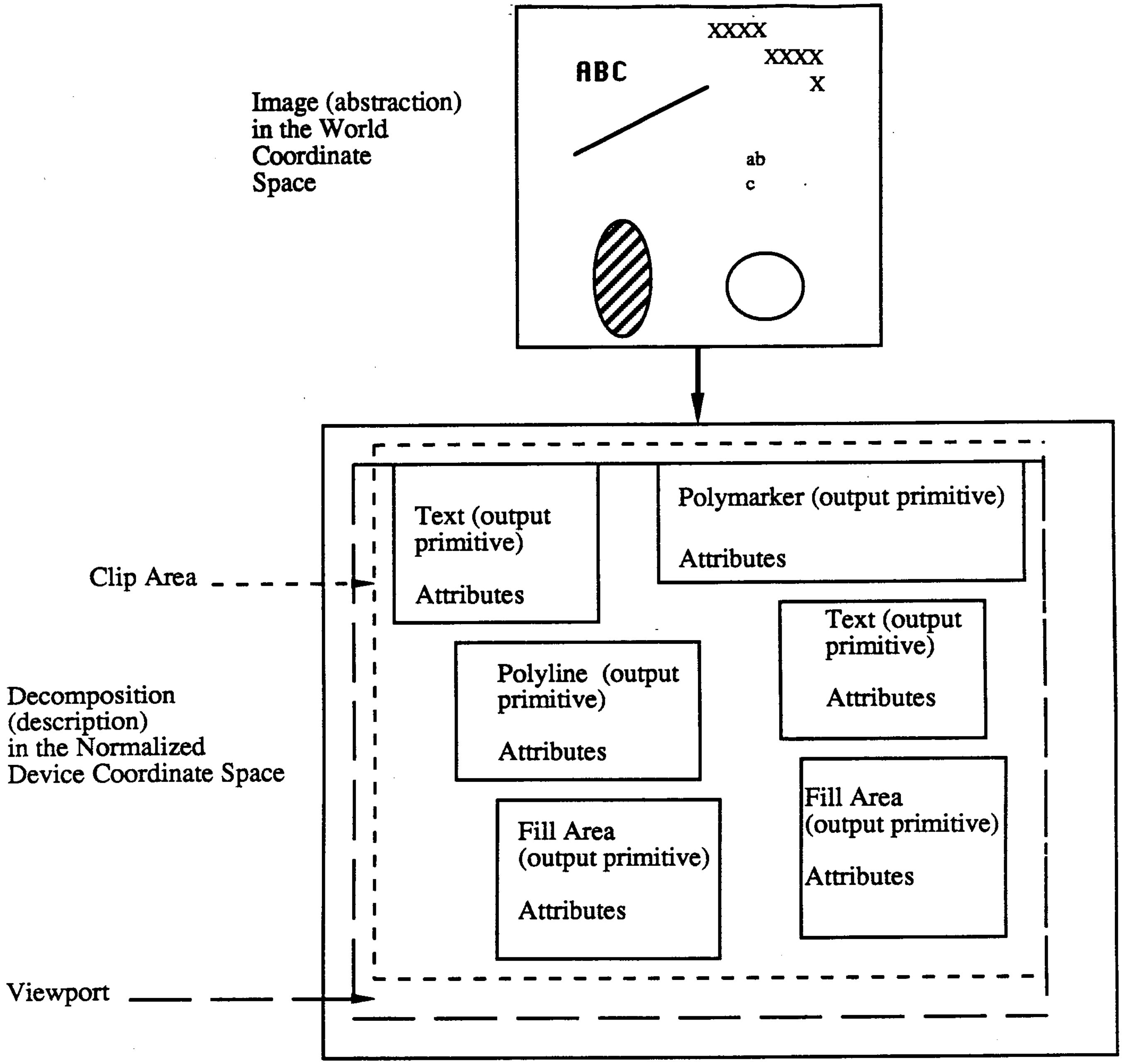


Figure 2.12.a : Graphics Standards - From an Image to a Display Using GKS.

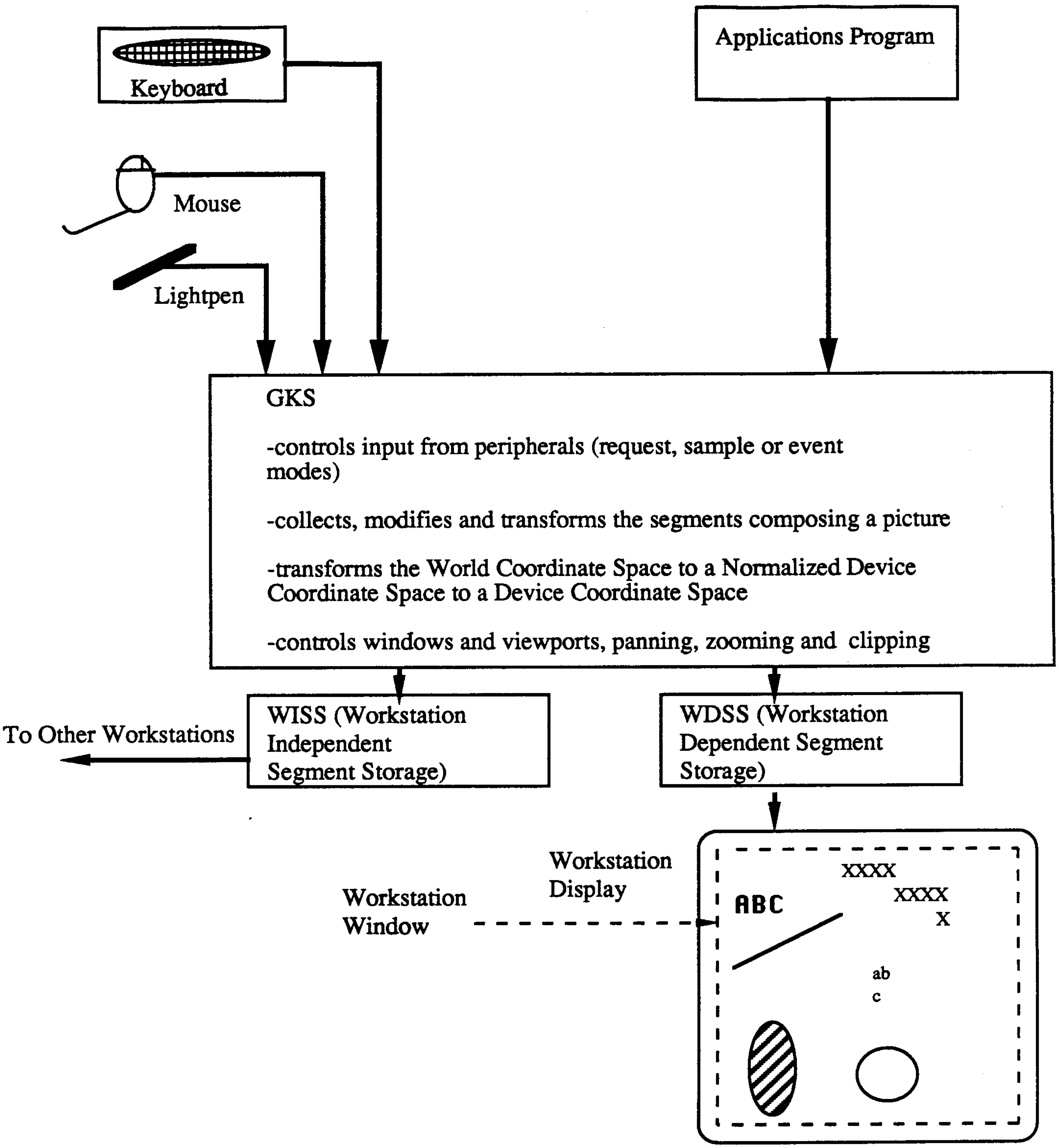


Figure 2.12.b : Graphics Standards - GKS System.

some part of which is mapped onto an area of the screen. The situation is complicated in GKS by g several workstations active at the same time. uld all workstations, then, be forced to use the same ewport? An application might require one display to give an overall view of the picture being displayed, while other looks at the detail of the picture.

KS achieves this flexibility through three different two-dimensional Cartesian coordinate systems and two distinct window/viewport mappings. The applications programmer defines his output in terms of a world coordinate system mapped onto some part of the normalized device coordinate plane. The set of active workstations can then take separate views of the NDC space and map these onto workstation-dependent parts of the display, expressed in device coordinates.

[redacted]

```
SET WINDOW(XMIN,XMAX,YMIN,YMAX)
DRAW PICTUREA
SET WINDOW(X2MIN,X2MAX,Y2MIN,Y2MAX)
DRAW PICTUREB
```

In this hypothetical package, PICTUREA is drawn with the first coordinate system, whereas PICTUREB is drawn with the second coordinate system. The user effectively sees a display made up of two parts with different coordinate systems. The user's view of the system is that both coordinate systems must be known to the system as pictures. However, in reality only the second coordinate system is known in a conventional package. When the user needs to point to a particular position in either coordinate system, the system cannot deliver the position in the correct coordinate system.

To ensure that the user's view of the system is correct, GKS allows the definition of multiple window/viewports, all existing simultaneously. The GKS equivalent of the above program would look like this:

```
DEF WINDOW (1,XMIN,XMAX,YMIN,YMAX)
DEF WINDOW (2,X2MIN,X2MAX,Y2MIN,Y2MAX)
SELECT WINDOW (1)
DRAW PICTUREA
SELECT WINDOW (2)
DRAW PICTUREB
```

Note that the form of the program in GKS has a tendency to define all the coordinate systems at the start of execution and then select the particular transformation when required. The other program form has transformation definitions scattered throughout.

[redacted]KS input is defined in terms of a set of logical devices. [redacted]w:

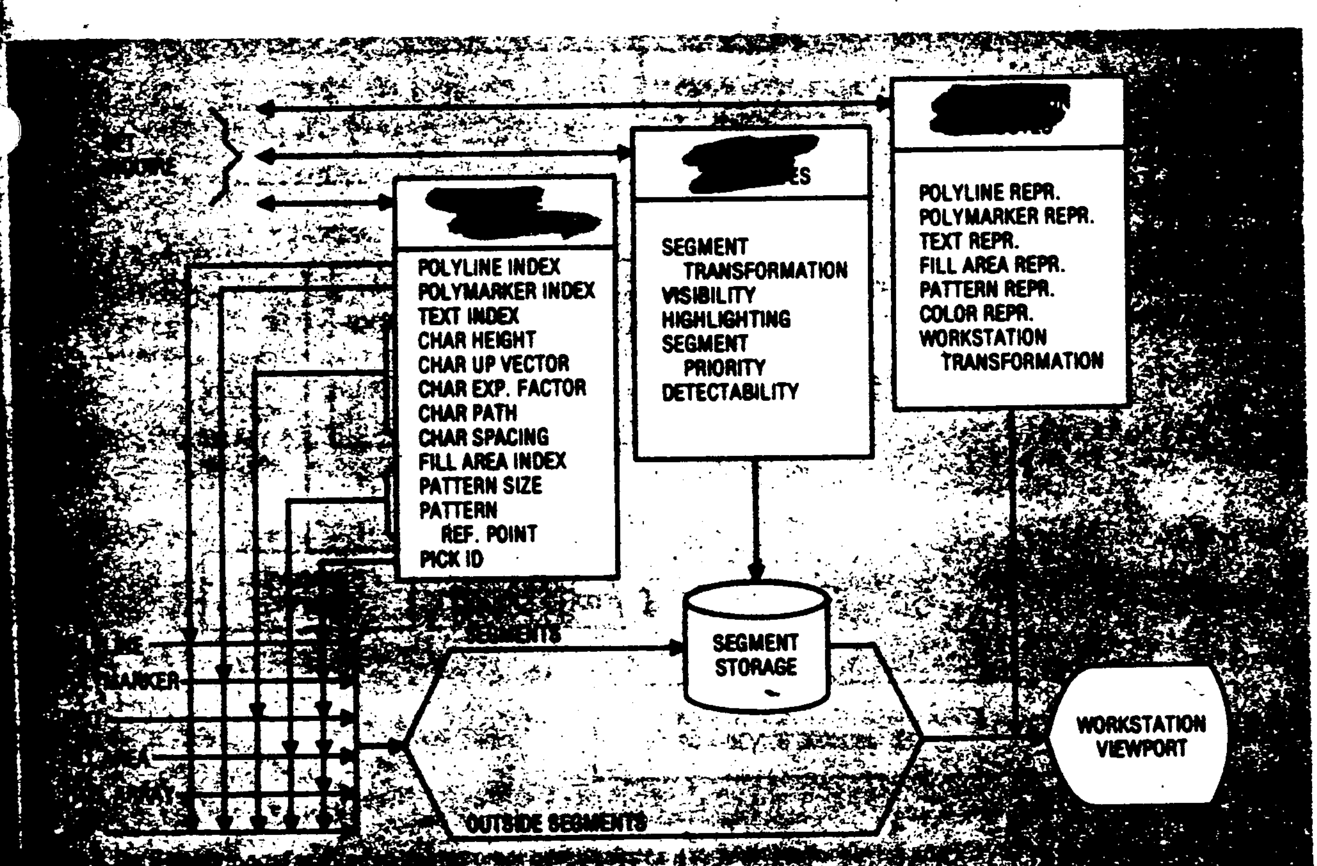


Figure 4. Binding of attributes to primitives and segments in GKS.

Segments. It is possible to generate output primitives so that they are displayed on all active workstations. However, in an interactive environment the complete picture frequently needs to be split into a number of objects or segments that can be manipulated independently. You may wish to highlight a particular part of the picture or remove it for some reason. In working with a refresh display, a user must often move parts of a picture around. This is achieved via a segment transformation matrix, which may be altered after the segment is defined.

Segments are stored on only those workstations that are active when the segment is defined. This is adequate for most purposes, but occasionally you need to see a segment on a workstation that was not activated when the segment was created. For example, the user may be defining a picture made up of segments on a refresh display and then at some stage may wish to copy the current display to a p ter. GKS allows this through a *device-independent* s *ment storage*, which can keep copies of segments as th are formed and incorporate any segment transformatio that occur at the workstation. When a copy is require the segments can be sent from device-independent s ment storage to a specified workstation. In the more com plex implementation levels of GKS, a segment can also b inserted into another segment. Figure 3 illustrates th behavior of segments in GKS, and Figure 4 shows how at tributes are bound to primitives and segments as they ar passed to the workstation.

Viewing. The typical graphical package has a sin window/viewport transformation that allows the a plication programmer to define his own coordin

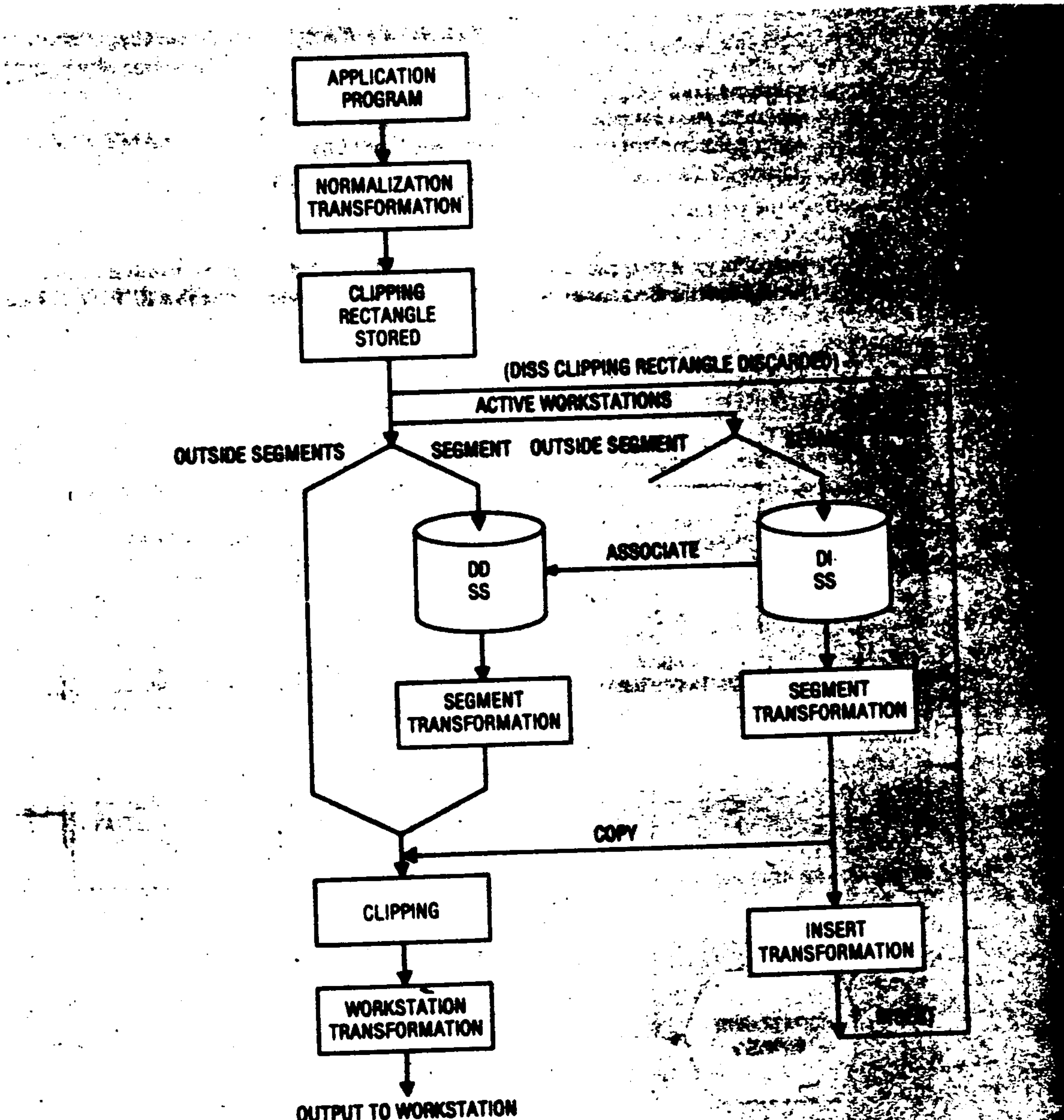


Figure 3. Data flow chart for graphic output in GKS. Only one workstation is shown together with the de independent segment storage. Also shown are the actions of associate, copy, and insert associated with the de independent segment storage.

A typical GKS image generation is shown in Figure 2.3x. It shows how the sets of primitives scan be combined to generate a set of segments. These segment in turn are used to generate the overall image. GKS descriptions are generally machine dependent and are structured by the collection of segments of the image. GKS allows for device independence by allocating the device specific portions the device drivers that are separate from the source code.

We can now take any GKS image and address the issue as to how we characterize the image in terms of data storage. Specifically, let us consider the image of a house that we have shown in Figure 2.31. The hose is composed of the following segments:

- o Roof
- o Chimney
- o Body of House
- o 10 Windows
- o 10 Window Shutters
- o Door
- o Light

The attributes associate with the roof are those of color, texture, scale and several other factors. Similar attributes are attributable to the other primitive in the segments.

Now we can look at the image from two points of view. First is we look at the 1,000 by 1,000 monitor at 12 bits per pixel, we have

12 million bits for the display. If on the other hand we look at the segment characterization, we can say the following:

- o 7 Separate Segments
- o 10 primitives per segment
- o 20 Attributes per segment
- o 20 Characters per segment
- o 8 bits per character.

This yields 244K bits for the storage of this image using GKS and no coding for either primitive or attribute expressions. This is a 50:1 image compression based just upon the fact that we know how the image is constructed. This is typical of may of the image systems that are found in the graphics area.

Figure 2.33 Graphics Generation of GKS

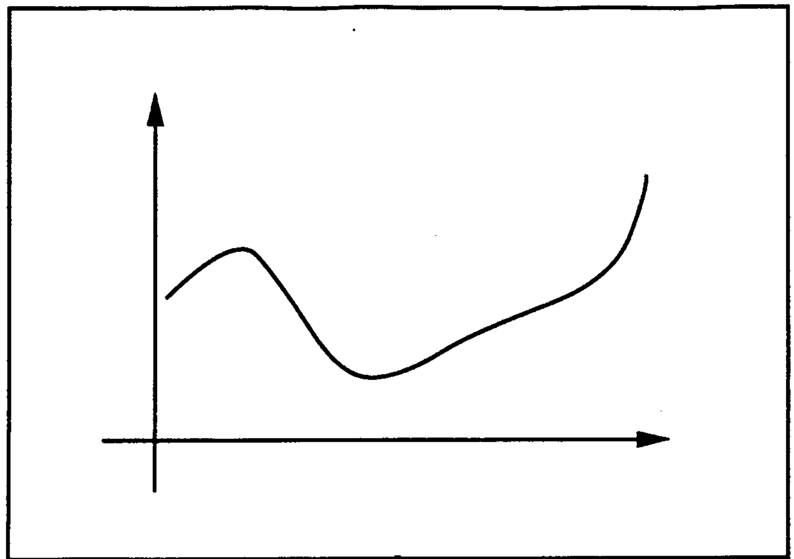
2.5.1.2 PHIGS

PHIGS is a second generation graphics standard. PHIGS is dramatically different from GKS in many dimensions, the two most important being the hierarchical nature of PHIGS and the second being the ability to dynamically center attributes as the application program is being run.

The overall architecture of the PHIGS approach is shown in Figure 2.x. Here we first show the basic application program which the PHIGS terms are inbedded. This interfaces with the PHIGS package that includes the graphics system to actually generate the graphic images and its associated graphics data elements.. These are then interfaced with the Input device drivers and the graphics display drivers that finally interface with the operator. As with the GKS approach, there is a segmentation of the device dependent elements and the the image dependent elements.

Figure 2.33 PHIGS Architecture (Ref Shuey et al p.51 IEEE CGA,August 1986)

Image (abstraction)



Decomposition
(description)
into hierarchically
arranged elements

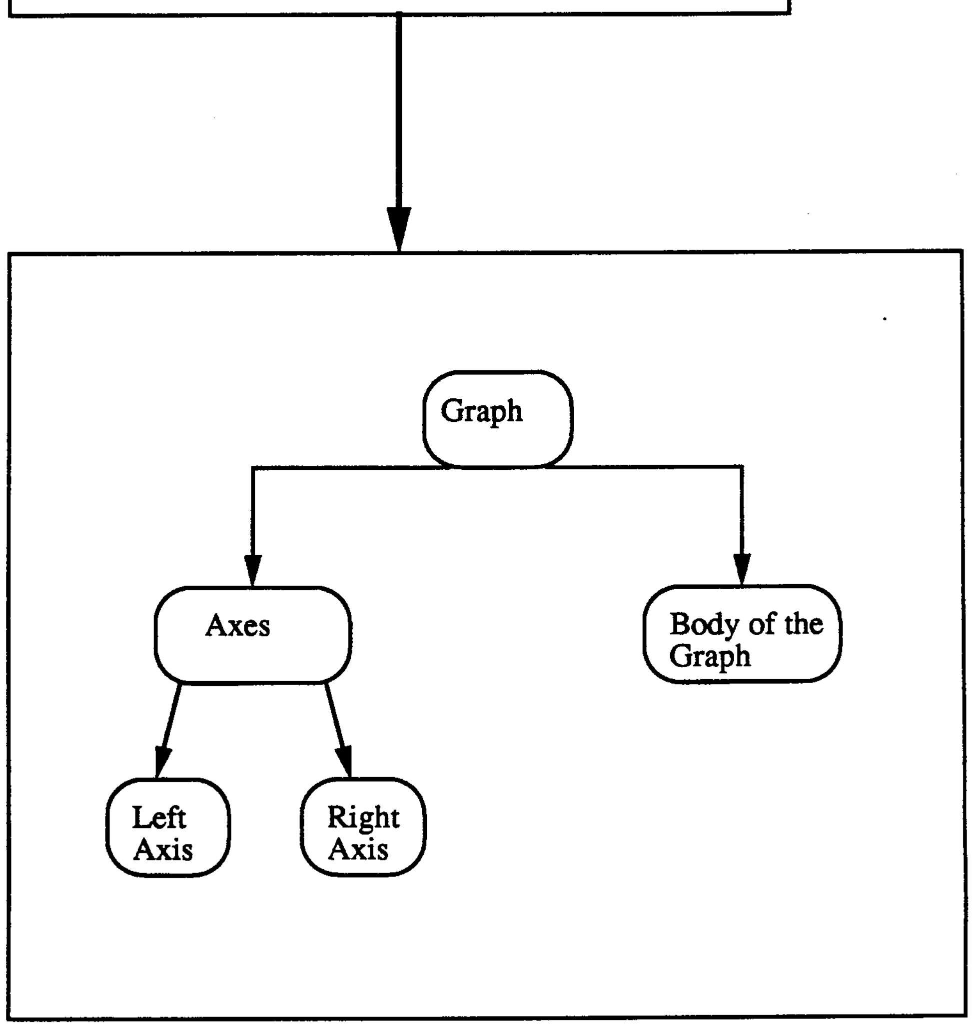


Figure 2.12.c : Graphics Standards - From an Image to a Description using PHIGS.

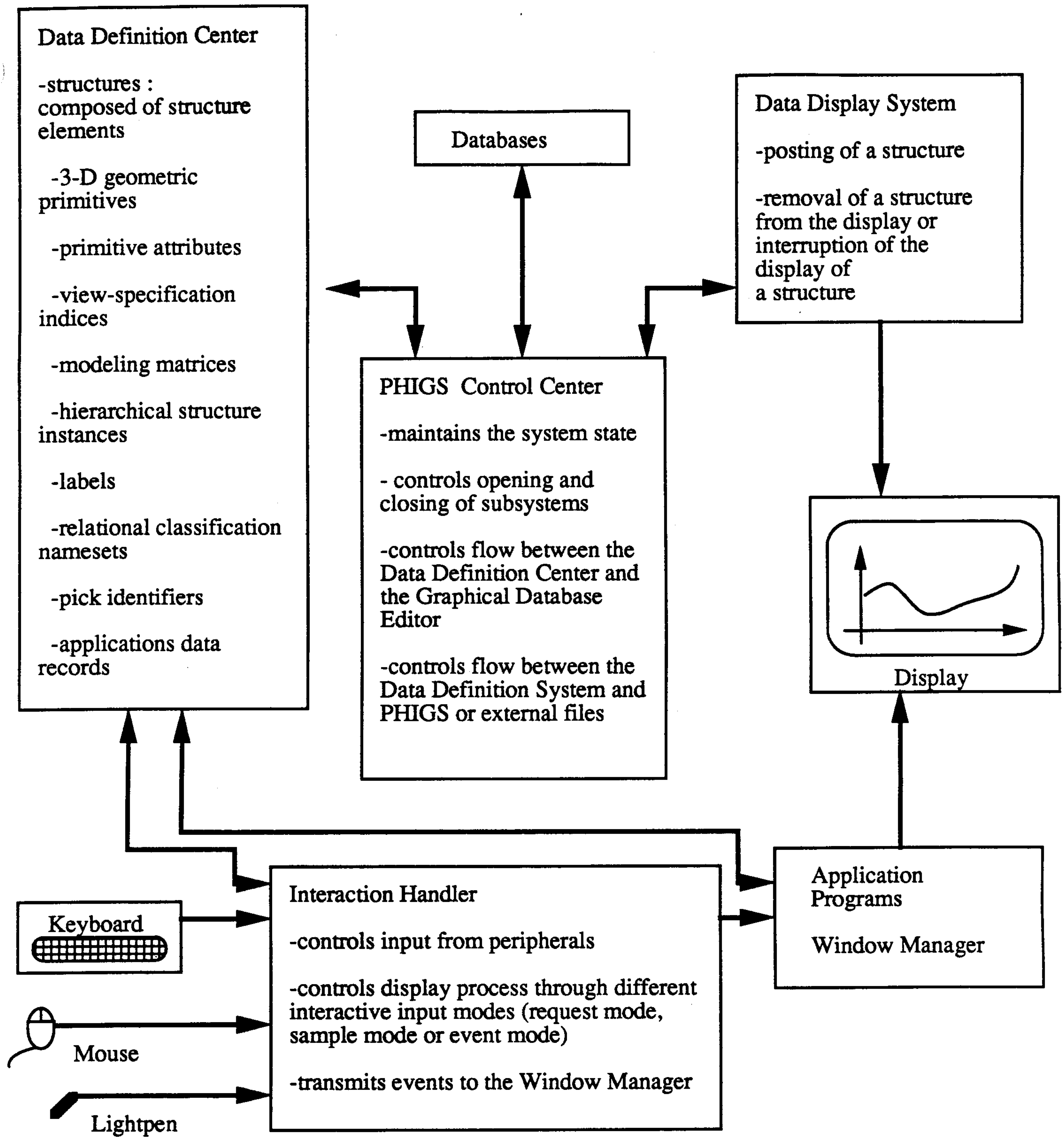


Figure 2.12.d : Graphics Standards - PHIGS System.

control and monitoring, and scientific modeling and simulation are a few application areas that have a common need for a graphics system like PHIGS. All these applications have large databases organized in a variety of ways; all attempt to interact with data in response to real-time feedback and/or operator interaction; and all attempt to reflect changes to their models in some form of graphical rendering.

Figure 1 depicts a typical system. Such a system maintains two sets of data: the application model, maintained by the application in a form suitable for manipulation by the application, and the graphics data, maintained by the graphics system in a form suitable for graphics rendering and manipulation.

The key features of PHIGS are its ability to efficiently describe the application model in the graphics system and to rapidly update the graphics model and corresponding rendering as the application model changes. The organization, content, and modifiability of the graphics data are the key aspects that make PHIGS a model manipulator's graphical toolset.

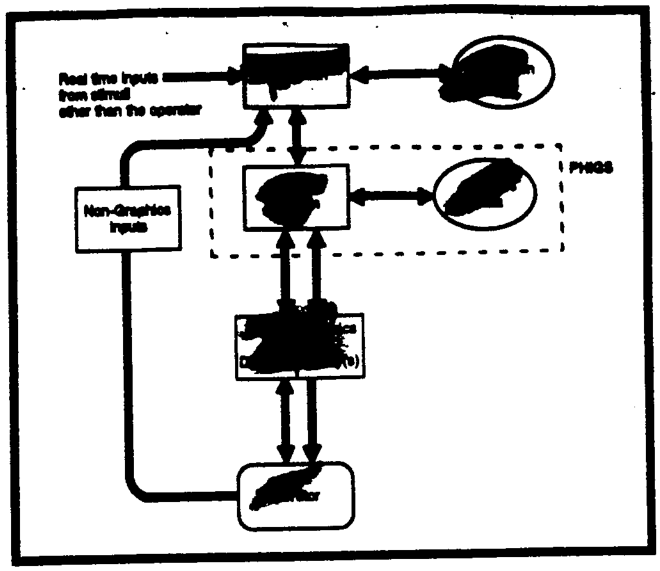


Figure 1. A typical applications system with graphics.

Model organization

Application models are organized along widely varied criteria. As a result, the topology of application data varies. For example, a model of the anatomy of an animal may be organized by physical connectivity such as muscles and bones, by major systems such as circulatory, nervous, and respiratory, or by more abstract terms such as chemical compounds or functions. However, regardless of the application or the organization almost all application models are multilevel (that is, not flat).

The organization of graphics data in PHIGS is also multilevel; it is hierarchical. A hierarchical data organization is common in many applications, and can easily be mapped onto from other data organizations. Indeed, PHIGS's graphics data structure places few restrictions on how graphics data must be organized, giving application programmers maximum flexibility. For example, branches of the hierarchy in PHIGS have no inherent geometric meaning, although many applications do associate geometric meaning to branches of application data (for example, the connections between segments of a robot arm—the joints—are generally represented by branches of a hierarchy with transformations associated with each branch).

Model content

The graphics data structures of PHIGS—the basic organizational building blocks of graphics models—are called *structures*. For the most part, structures contain the standard graphical elements found in other standard programmer interfaces, such as GKS, or in other standard device interfaces, such as CGI. We will discuss in detail the graphical elements that are unique to PHIGS later in this article.

The power of PHIGS is derived not from the primitives and attributes it provides, but from how these elements are organized and manipulated. PHIGS could have provided a very powerful framework for a graphics system based on an entirely different set of primitives and attributes. Instead of using primitives and attributes that are planar- and vector-oriented, PHIGS might have augmented or replaced those primitives with solids or volumes with attributes such as light models and shading. However, a common base of knowledge and practice has not yet evolved for these primitives and attributes, so it is outside the purview of the standard to define these types of primitives and attributes. PHIGS uses graphics data organization and manipulation techniques that are becoming widely accepted and used.

Model modification

In addition to having a multilevel, flexible data organization, PHIGS allows modification of the graphics data at any time. The editability of the graphics data is such an integral part of PHIGS that creation and modification of graphics data are one and the same. Any portion of the graphics model can be modified to reflect changes in the application model and, as a result, the graphical rendering also reflects the change. It is this facility that makes interactive, dynamic graphics possible.

For model rendering, PHIGS defines all geometric primitive elements and geometric aspects of the graphics model in what is known as a *modeling coordinate system*. Figure 2 depicts the relationship of the modeling coordinate system to the rest of the PHIGS transformation pipeline. Note that elements of the graphics model can pass through multiple modeling transformations in succession, each mapping the data back into the modeling coordinate space for further transformation, if necessary. In this way structures representing a geometric entity are defined. A particular geometric entity can be defined in a coordinate space suitable for its definition, and it can be combined with other geometric entities by applying the appropriate transformations. Since modeling transformations are also elements of the graphics data structure, and since every element of the graphics data structure can be modified at any time, complex and interesting effects can be achieved. Manipulation of a robot arm or a

PHIGS has the two basic elements of the primitive and the attribute. The primitive carries with it the same function of form and shape that we had in GKS. Thus the primitive of a line or circle can be found in the PHIGS format. The attribute is similar to that of GKS in that is associates with a primitive a specific set of characteristics such as shading and width.

The difference between PHIGS and GKS is that in GKS we bind the attribute and the primitive together once and for all. In PHIGS, we can associate a set of attributes and primitives together as needed and this association can change with time. This can be shown in Figure 2.x The binding of the two elements is an output process not a definition process.

This allows for a significantly more flexible design and it allows for the ability to perform animation in a three dimensional context which is not possible with the GKS type of design. Like GKS however it is a structural language and builds from the vector graphics paradigm.

PHIGS is a hierarchical structure. Specifically it allows for the development of STRUCTURES that are bindable primitives and attributes and these structures can be composed upward or decomposed downward. For example in GKS the HOUSE was defined in terms of the roof, the chimney the windows, etc. In PHIGS, we can take the house and take any one elements and further decompose it. Thus the roof can be decomposed into the shingles, the nails, the edges etc. The shingles can be further decomposed into the tar, the paper, and the gravel. We could continue this process down to whatever detail is necessary. This approach in PHIGS allows the graphics designer to provides greater detail in the specific image.

- GKS provides a limited modeling transformation, called a normalization transformation, that is applied once to a given set of data. [redacted]odels are created and manipulated. GKS does provide a composition space at the image level, after the primitives have been normalized, a strategy that is consistent with its role as an image-manipulation system. PHIGS is a model-manipulation system.

It is important to note that PHIGS and GKS differ only in the areas outlined above, the organization and manipulation of graphics data. In many fundamental ways—primitive attributes, workstations, graphics input, error processing, deferral and control—PHIGS and GKS are either identical, or PHIGS is a superset of GKS. Where differences exist between PHIGS and GKS, they exist in order to allow PHIGS to meet its goal of being an effective model-manipulator's graphical toolset.

It is important to build on a common base of graphics knowledge. PHIGS and GKS share a common set of graphics concepts and terminology, and together address the needs of the graphics marketplace by supplying consistent graphics standards.

Key functional areas of PHIGS

As we have described, graphics data in PHIGS is organized in structures. This organization, controlled by the application program, allows the programmer to arrange the data in a way best suited to the intended task.

Structures contain *structure elements*, which represent output primitives, attribute selections, viewing selections, modeling transformations, labels, name set elements, application-specific data, or structure references. The application program gives structures a unique identifier for later reference.

Structure hierarchy

A structure hierarchy is created through the use of the EXECUTE structure element. During display traversal, this element causes another structure to be executed within the context of the original structure, resulting in a hierarchical organization, like that diagrammed in Figure 6. In this example, structure 1 contains EXECUTE structure elements referring to structures 2 and 3. Structure 1 is called the parent of structures 2 and 3; structures 2 and 3 are called the children of structure 1. There is no limit to the number of structures that may execute a given structure, nor is there a limit to the depth of the hierarchy.

For a structure hierarchy to be traversed for display, a structure must be posted to a workstation. (In our example, structure 1 would be posted to a workstation.) Structure traversal begins with posted structures. However, PHIGS does allow for the creation of structures that are neither executed as a part of another structure nor posted to a workstation. PHIGS stores those structures to be used later in EXECUTE structure elements or as posted structures, for application use later. A common example is a symbol library composed of a collection of structures.

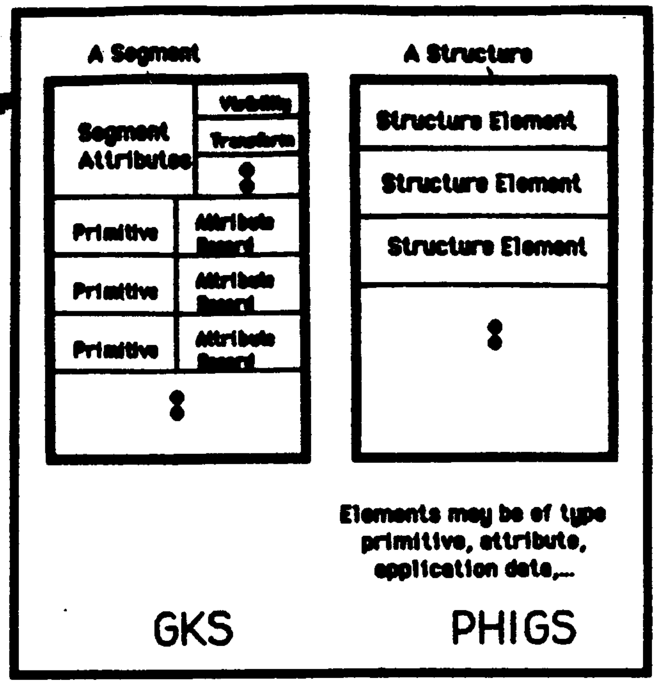


Figure 4. A comparison of segment content in GKS and PHIGS.

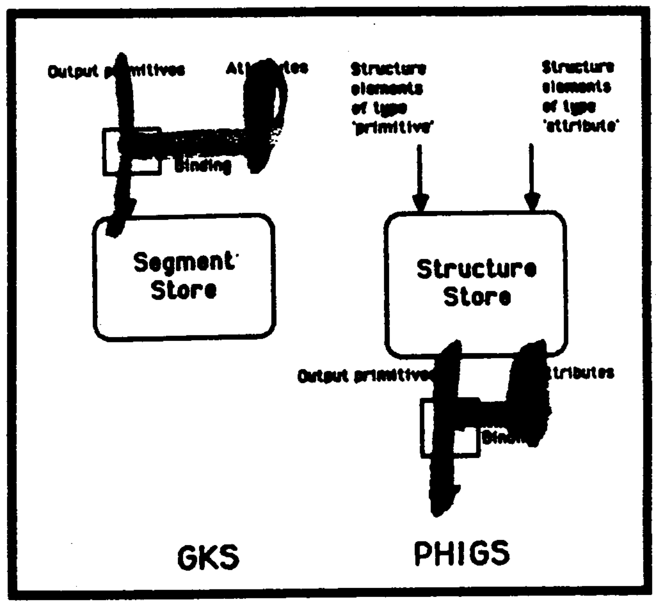


Figure 5. A comparison of attribute binding in GKS and PHIGS.

Traversal-time binding

Like other graphics systems, PHIGS uses attributes to define the appearance of an output primitive. The attributes defined in PHIGS are the same as those defined in other graphics standards, but in PHIGS they are bound at structure traversal time, not when the structures are created. Since attribute specification is modal, how attributes are used in the structure hierarchy is important. When an EXECUTE structure element is encountered

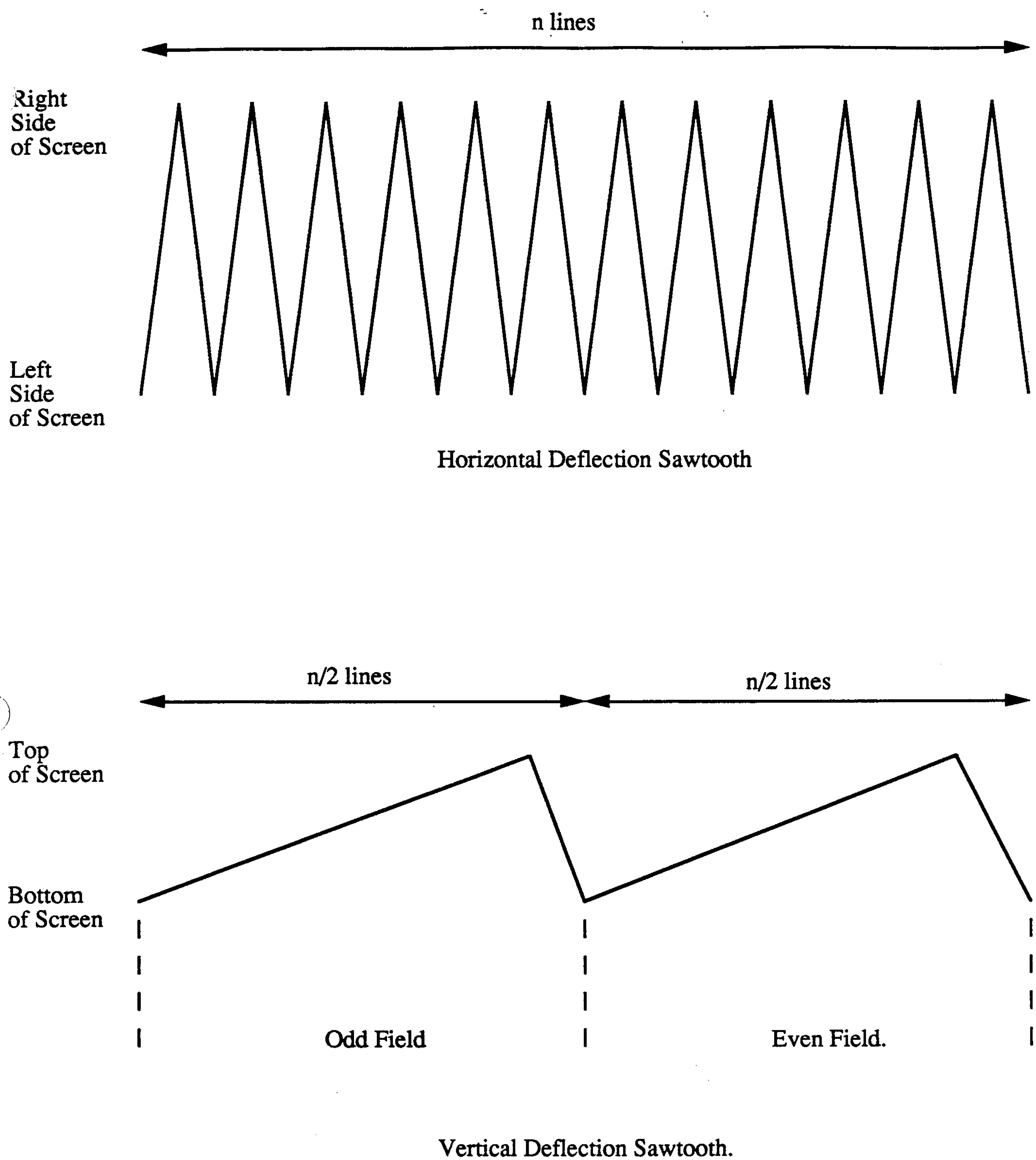


Figure 2.6 : Video Scan Scheme.

Figure 2.34 PHIGS Structure Store (Ref Shuey et al p.53)

In the GKS environment we had the segment and in the PHIGS environment we have the structure. One of the major differences is that a GKS segment is a defined and immutable element whereas the PHIGS structure is highly flexible and thus amenable to redefinition.

The further capability of binding attributes on the output side allows for the flexibility to create highly real graphics.

Figure 2.3x shows further detail on the PHIGS structure. As with the GKS model, we can develop methods for sizing PHIGS graphics image data elements by sizing the lines of code and then proceeding to consider the compression capabilities that are possible within the code. We consider several of these in the problems.

Figure 2.35 Graphics Generation of PHIGS

2.5.1.3 PostScript

PostScript was the development of John Warnock while at Xerox PARC and is focused on the truly bitmapped display devices such as high resolution terminals displays and laser printers. The PostScript approach is one that mimics the paradigm of the printer placing ink on the page to obtain the effect that is sough by the graphic artist. It is a much more flexible development language and implementation system than many of the other alternatives. It, in addition, is directed at the transfer of the developed graphics to paper, whereas the PHIGS approach is optimized for the continuing display on electronic media.

The basis design object in PostScript are the:

o Text Elements that provide the large selection of standard and customizable text fonts.

o Geometric elements that provide for the definition of circles, lines and rectangles.

o Sampled Image elements that permit the direct importation ad manipulation of external images.

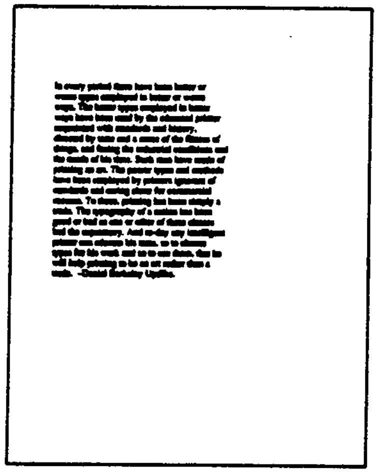
These are all combined in the overall construct of the imaging model. This is the equivalent to the segment of GKS or the structure of PHIGS. The imaging model is composed of three elements;

o Current Page: This is the abstraction of the working space on which the graphics layout is made. The current page is the blank space that is filled out by the designer.

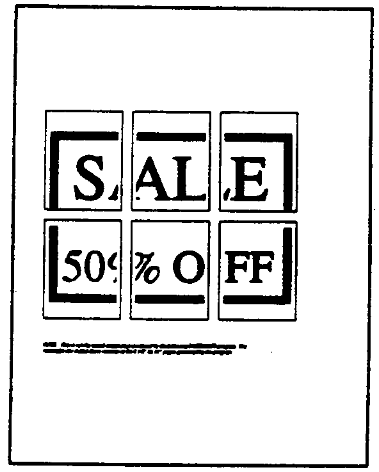
o Current Path: This is an abstraction that may be independent of the current page and it is the collection of the basic graphics objects.

o Clipping Path: This is the active part of the current page that may be drawn upon.

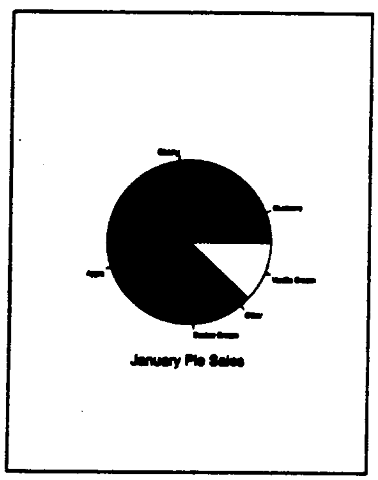
Figure 2.36 Graphics Generation of PostScript



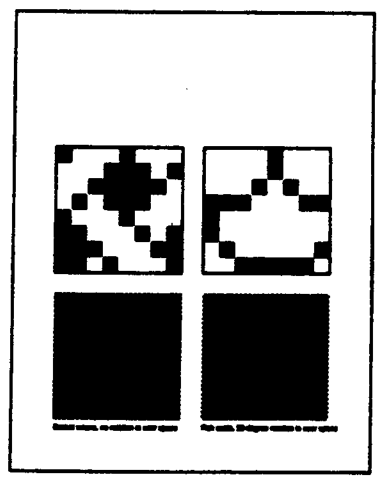
A Simple Line Breaking Algorithm



Making a Poster



Drawing a Pie Chart



Filling an Area with a Pattern

***NOTE:** This is not the actual output page produced by the following POSTSCRIPT program. The rectangles are scaled down versions of the 8 1/2" by 11" pages generated by the program.*

13 / Making a Poster

This program demonstrates how to print a picture larger than a sheet of paper (8.5" by 11") on several sheets of paper that can be pasted together later.

"printposter" takes a large picture (larger than 8.5" by 11") and prints it on several pages according to the number of rows and columns specified. Imagine superimposing a grid composed of the specified number of rows and columns on the large image. Then each rectangle in the grid represents an 8.5" by 11" page to be printed. "printposter" takes three arguments: a procedure representing the large picture, the number of columns and the number of rows.

```
/printposter
 { /rows exch def
   /columns exch def
   /bigpictureproc exch def
```

Set up a clipping region for the page we will print on. Since most printers cannot print to the very edge of the paper, we will explicitly set up the clipping boundary so that it lies within the printing boundaries of the printer and we will compensate for this when we print the large image so that all parts of the image are actually printed.

```
   newpath
    leftmargin botmargin moveto
    0 pageheight rlineto
    pagewidth 0 rlineto
    0 pageheight neg rlineto
   closepath clip
```

Readjust the origin on the page so that it coincides with the origin of the clipping boundary.

```
   leftmargin botmargin translate
```

For each row of pages...

For each page within that row...

```
   0 1 rows 1 sub
    { /rowcount exch def
      0 1 columns 1 sub
       { /colcount exch def
         gsave
```

Translate the large picture so that the desired section will be imaged on the printed page. We must translate the large picture in the negative direction so that the lower left corner of the section to be printed always coincides with the origin.

```
          pagewidth colcount mul neg
            pageheight rowcount mul neg
            translate
```

Execute the large picture, clipping to this page.
Since the showpage operator has the side effect of executing the initgraphics operator (which would reset the clipping region), we bracket it by the gsave and grestore operators.

```
          bigpictureproc
          gsave showpage grestore

         grestore
       } for
    } for
 } def
```

(continued)

```
/inch {72 mul} def
```

```
/leftmargin .5 inch def
/botmargin .25 inch def
/pagewidth 7.5 inch def
/pageheight 10 inch def
```

These are the dimensions of the clipping boundary.

```
/salesign
 { gsave
```

This procedure draws a large sign with a border. The sign is 22.5 inches wide and 19.5 inches high which fits comfortably on 6 8.5 inch by 11 inch pages (the final result will be 2 rows of pages high and 3 columns of pages wide).

```
    /Times-Roman findfont 500 scalefont setfont
    2.5 inch 11 inch moveto
    (SALE) show
    /Times-Roman findfont 350 scalefont setfont
    1.45 inch 4 inch moveto
    .5 setgray (50%) show
    0 setgray ( OFF) show
    newpath
```

```
     .5 inch 18 inch moveto
     22 inch 18 inch lineto
     22 inch 2 inch lineto
     .5 inch 2 inch lineto
    closepath
```

Specify the path for the border.

```
    gsave
     .75 inch setlinewidth stroke
    grestore
```

First paint the border with a thick black stroke.

```
    10 setlinewidth 1 setgray stroke
```

Then paint a thin white stroke down the center of the border.

```
  grestore
 } def
```

```
{salesign} 3 2 printposter
```

Print the large picture on a total of 6 pages. The image is three columns of pages wide and 2 rows of pages high.

2.5.2 Means of Storage

The storage of the graphics display systems depends upon whether they are in a bit mapped form or in a vector scan mode. We shall briefly discuss the vector form but shall concentrate on the bit mapped version. For bit mapped displays, there are many standards for their storage. Several of the example that we shall discuss are the DDIF, GIF, HALO and Figaro formats.

Figure 2.37 Bit Mapped Storage Standards (a) DDIF, (b) GIF

2.5.3 Means of Reconstitution

The means for reconstituting the graphics displays are similar to those for fixed images. We shall briefly discuss the alternatives.

Figure 2.38 Graphics Display Terminal

2.6 Conclusions

This chapter provides the reader with a method for the characterization and modeling of various multimedia information sources. The source characterization problem is one of the key problems in the development and analysis of multimedia systems. The source of a single user must be composed of a set of the fundamental sources that that user may interact with. Those fundamental sources are those that we have developed in this section, and possible others that may be developed as new technologies are developed.

The most significant fact that the reader should obtain fom this chapter is the recognition that in a multimedia environment, we are always striving to duplicate or record a reality that exists external to the machine. The techniques for doing so have been developed both within the context of the computer world and within the world of print medium. We too often forget the importance of the print medium as a means of transferring information from one location to another and from one person to another. We also are seeing that with the introduction of such applications as digital printing systems, we are seeing the melding together of the print and electronic industries from both the operations and applications viewpoint.

Chapter 2 Multimedia Characteristics.

2.1 Types of Media.

2.1.1 Still Image.

'The 35mm Handbook'
M. Freeman
Ziff David Books, 1980

'Content-Driven Transmission of Grey-Scale Images'
H.M. Dreizen in
IEEE Transactions on Communications, vol. COM-35, no.3, March 1987

'A Painting Algorithm of Monochromatic Images Using Color-Sequences'
T. Agui, M. Nakajima, H. Kitamura
translated from Denshi Tsushin Gakkai Ronbushi, vol 67-D, No.3, March 1984
Scripta Publishing Company, 1984

'Anti-Aliasing for Broadcast Compatible Television'
W. Bender
SID 84 Digest

'Locally Optimal Run-Length Compression Applied to CT Images'
M.L. Rhodes, J.F. Quinn, J. Silvester in
IEEE Transactions on Medical Imaging, vol. MI-4, no. 2, June 1985

'CRT Typeface Design and Evaluation'
W. Bender, R.A. Crespo, P.J. Kennedy, R. Oakley
MIT Industrial Liaison Report S0288-069, 1988

'Multimedia Communications - Reading and Writing the Electronic Book'
N. Yankelovich, N. Meyrowitz, A. van Dam in
'Computer', IEEE, October 1985

'Chapter 1/Section 7 : Introduction/
A Generalized Image Processing System' in
'Fundamentals of Electronic Imaging Systems-
Some Aspects of Image Processing'
W.F. Schreiber
Springer Series in Information Sciences, Springer-Verlag, 1986

2.1.2 Video Images.

'Color and Black & White Television Theory and Servicing'
A.A. Liff, second edition,
Prentice-Hall, Inc., Englewood Cliffs, 1985

'A Scanning Scheme for a New HDTV Standard'
B. Wendland in
'HDTV '85'
Minister of Supply and Services Canada, 1986

'Psychovisual Bases for Television Picture Enhancement'
E. Bourguignat in
'Image Coding'
edited by M. Kunt, T.S. Huang
Proceedings of theSPIE-International Society for Optical Engineering, volume 594, 1985

'Chapter 1/ Section 8 : Introduction/
A Simple Television System' in
'Fundamentals of Electronic Imaging Systems-
Some Aspects of Image Processing'
W.F. Schreiber
Springer Series in Information Sciences, Springer-Verlag, 1986

2.1.4 Graphics.

'PHIGS : Programmer's Hierarchical Interactive Graphics Standards - A Giant Step toward a Universal Graphics Standard'
M. Plaehn in
Byte, November 1987

2.2 Image Construct Paradigms

2.2.1 Resolution.

'Digital Picture Processing'
Second Edition, vol. 1
A. Rosenfeld, A.C. Kak
Computer Science and Applied Mathematics, Academic Press, 1982

'Digital Coding of Wavforms -
Principles and Applications to Speech and Video'
N.S. Jayant, P. Noll
Prentice-Hall Signal Processing Series, 1984

'Image data Compression Based on Interpolative DPCM by Area Decomposition'
A. Machizawa, M. Tanaka in
tranalated from Denshi Tsushin Gakkai Ronbushi, vol 69-D, no.3, March 1986
Systems and Computers in Japan, vol. 17, no.12, 1986

'Data Compression of Gray-Scale Images by Level Plane Coding'
Y. Sato, K. Takahashi, M. Ohta
tranlsted from Denshi Tsushin Gakkai Ronbushi, vol 69-B, no.8, August 1986,
Electronics and Communications in Japan, Part 1, Vol. 71, no.1, 1988

'A Painting Algorithm of Monochromatic Images Using Color-Sequences'
T. Agui, M. Nakajima, H. Kitamura in
translated from Denshi Tsushin Gakkai Ronbunshi, vol67-D, No. 3, March 1984
Scripta Publishing Co., 1984

'Anti-Aliasing for Broadcast Compatible Television'
W. Bender in
SID 84 Digest, 1984

'CRT Typeface Design and Evaluation'
W. Bender, R.A. Crespo, P.J. Kennedy, R. Oakley
MIT Industrial Liaison Report S0288-069, 1988

'Direct Comparison of 35mm Film and High Definition Television'
A. Kaiser, D.F. Morss III in
'High Definition Television Colloquium '85', vol. 2,
Minister of Supplies and Services, Canada, 1986

'Segmentation of Color Images Using Edge-Preserving Filters'
M. Pietkäinen, D. Harwood in
'Advances in Image Processing and Pattern Recognition'
edited by V. Cappellini, R. Marconi
North-Holland, 1986

'Image Acquisition and Presentation'
A.N. Venetsanopoulos in
'Advances in Image Processing and Pattern Recognition'
edited by V. Cappellini, R. Marconi
North-Holland, 1986

'Surface Reconstruction from Planar Cross Sections for 3-D Integrated Representation of Medical Images'
D. Vandermeulen, P. Suetens, J. Gybels, A. Oosterlinck in
'Advances in Image Processing - Proceedings'
edited by A. Oosterlinck, A.G. Tescher
SPIE - The International Society for Optical Engineering, 1987

'Pattern Representation'
P.J.W. ten Hagen, G.G. Trienekens in
'Data Structures for Raster Graphics'
edited by L.R.A. Kessener, F.J. Peters and M.L.P. van Lierop
Eurographic Seminars, Springer-Verlag, 1986

'Using Linear Quadtrees to Store Vector Data'
H. Samet, C.F. Shaffer in
'Data Structures for Raster Graphics'
edited by L.R.A. Kessener, F.J. Peters and M.L.P. van Lierop
Eurographic Seminars, Springer-Verlag, 1986

'Problems with Raster Graphics Algorithms'
W.R. Franklin in
'Data Structures for Raster Graphics'
edited by L.R.A. Kessener, F.J. Peters and M.L.P. van Lierop
Eurographic Seminars, Springer-Verlag, 1986

2.2.2 Information Content.

'Locally Optimal Run-Length Compression Applied to CRT Images'
M. L. Rhodes, J.F. Quinn, J. Sylvester in
IEEE Transactions on Medical Imaging, vol. MI-4, no.2, June 1985.

'A Modified Block Truncation Coding Technique for Image Compression'
E. Walach, J. Bruck, D. Chevion, E. Karnin, D. Ramm in

'Advances in Image Processing and Pattern Recognition'
edited by V. Cappellini, R. Marconi
North-Holland, 1986

'Intermediate Data Structures for Display Algorithms'
M.L.P. van Lierop in
'Data Structures for Raster Graphics'
edited by L.R.A. Kessener, F.J. Peters and M.L.P. van Lierop
Eurographic Seminars, Springer-Verlag, 1986

2.2.3 Decomposition.

'Digital Picture Processing'
Second Edition, vol. 1
A. Rosenfeld, A.C. Kak
Computer Science and Applied Mathematics, Academic Press, 1982

'Digital Coding of Waveforms -
Principles and Applications to Speech and Video'
N.S. Jayant, P. Noll
Prentice-Hall Signal Processing Series, 1984

'Adapative Gray Scale Mapping to Reduce Registration Noise in Difference Images'
T.F. Knoll, E.J. Delp in
Computer Vision, Graphics and Image Processing 33, 129-137, 1986.

'Content-Driven Progressive Transmission of Grey-Scale Images'
H.M. Dreizen in
IEEE transactions on Communications, vol. COM-35, no.3, March 1987.

'A Computer Vision System for Interpretation of Paper-Based Maps'
J. Alemany, R. Kasturi in
'Applications of Digital Image Processing X - Proceedings'
edited by A.G. Tescher
SPIE - The International Society for Optical Engineering, 1987

'A Hybrid Coding Scheme for Still Images :
Adaptive Vector Quantization with Correction Information in DCT Domain'
G. Tu, G. Verbeek, J. Rommelaere, A. Oosterlinck in
'Applications of Digital Image Processing X - Proceedings'
edited by A.G. Tescher
SPIE - The International Society for Optical Engineering, 1987

'Distributed Arithmetic Implementation of the D.C.T. for Real Time Photvideotex on ISDN'
A. Leger, J.P. Duhamel, J.L. Sicre, G. Madec, J.M. Knoepfli in
'Advances in Image Processing - Proceedings'
edited by A. Oosterlinck, A.G. Tescher
SPIE - The International Society for Optical Engineering, 1987

'A Model for Raster Graphics Language Primitives'
W.J.M. Teunissen, J. van den Bos in
'Data Structures for Raster Graphics'
edited by L.R.A. Kessener, F.J. Peters and M.L.P. van Lierop

Eurographic Seminars, Springer-Verlag, 1986

'Data Structures for Ray Tracing'
F.W. Jansen in
'Data Structures for Raster Graphics'
edited by L.R.A. Kessener, F.J. Peters and M.L.P. van Lierop
Eurographic Seminars, Springer-Verlag, 1986

'Intermediate Data Structures for Display Algorithms'
M.L.P. van Lierop in
'Data Structures for Raster Graphics'
edited by L.R.A. Kessener, F.J. Peters and M.L.P. van Lierop
Eurographic Seminars, Springer-Verlag, 1986

2.3 Video Construct Paradigms.

2.3.1 Resolution.

'Color Display System for High-Definition Television'
K. Morita, T. Kataoka in
'High Definition Television Colloquium '85', vol. 2,
Minister of Supplies and Services, Canada, 1986

'Limitations to Current Psychophysical Approaches to Display Evaluation'
P.J. Hearty in
'High Definition Television Colloquium '85', vol. 2,
Minister of Supplies and Services, Canada, 1986

'An Image Knowledge Based Video Codec for Low Bitrates'
R.H.J.M. Plompen, J.G.P. Groenveld, F. Booman, D.E. Boekee in
'Advances in Image Processing - Proceedings'
edited by A. Oosterlinck, A.G. Tescher
SPIE - The International Society for Optical Engineering, 1987

2.3.2 Information Content.

'Transmission Systems for Communications'
fifth edition
Bell Telephone Laboratories Incorporated, 1982

'Getting the Picture -
A Guide to CATV and the New Electronic Media'
S.B. Weinstein
IEEE Press, 1986

'Digital Coding of Waveforms -
Principles and Applications to Speech and Video'
N.S. Jayant, P. Noll
Prentice-Hall Signal Processing Series, 1984

'The Challenge of VLSI Technology to Video Communications'
T.R. Hsing, M.L.Liou in
'Applications of Digital Image Processing X - Proceedings'
edited by A.G. Tescher

SPIE - The International Society for Optical Engineering, 1987

'Applications of Digital Image Processing X - Proceedings'
edited by A.G. Tescher
SPIE - The International Society for Optical Engineering, 1987

'Improved Hybrid Coders with 2-D Signal Processing for Moving Pictures'
J. Speidel, P. Vogel in
'Advances in Image Processing - Proceedings'
edited by A. Oosterlinck, A.G. Tescher
SPIE - The International Society for Optical Engineering, 1987

2.3.3 Decomposition.

'The Application of Human Visual System Models to Digital Color Image Compression'
C.F. Hall in
'Image Coding'
edited by M. Kunt, T.S. Huang
Proceedings of the SPIE - The International Society for Optical Engineering, volume 594, 1985

'An Integrated Approach to Video Standard Conversion and Pre-Processing for Videoconference Codec'
L. Chiariglione, M. Guglielmo, F. Tommasi in
'Image Coding'
edited by M. Kunt, T.S. Huang
Proceedings of the SPIE - The International Society for Optical Engineering, volume 594, 1985

2.4 Voice Construct Paradigms.

2.4.1 Resolution.

'Digital Coding of Waveforms -
Principles and Applications to Speech and Video'
N.S. Jayant, P. Noll
Prentice-Hall Signal Processing Series, 1984

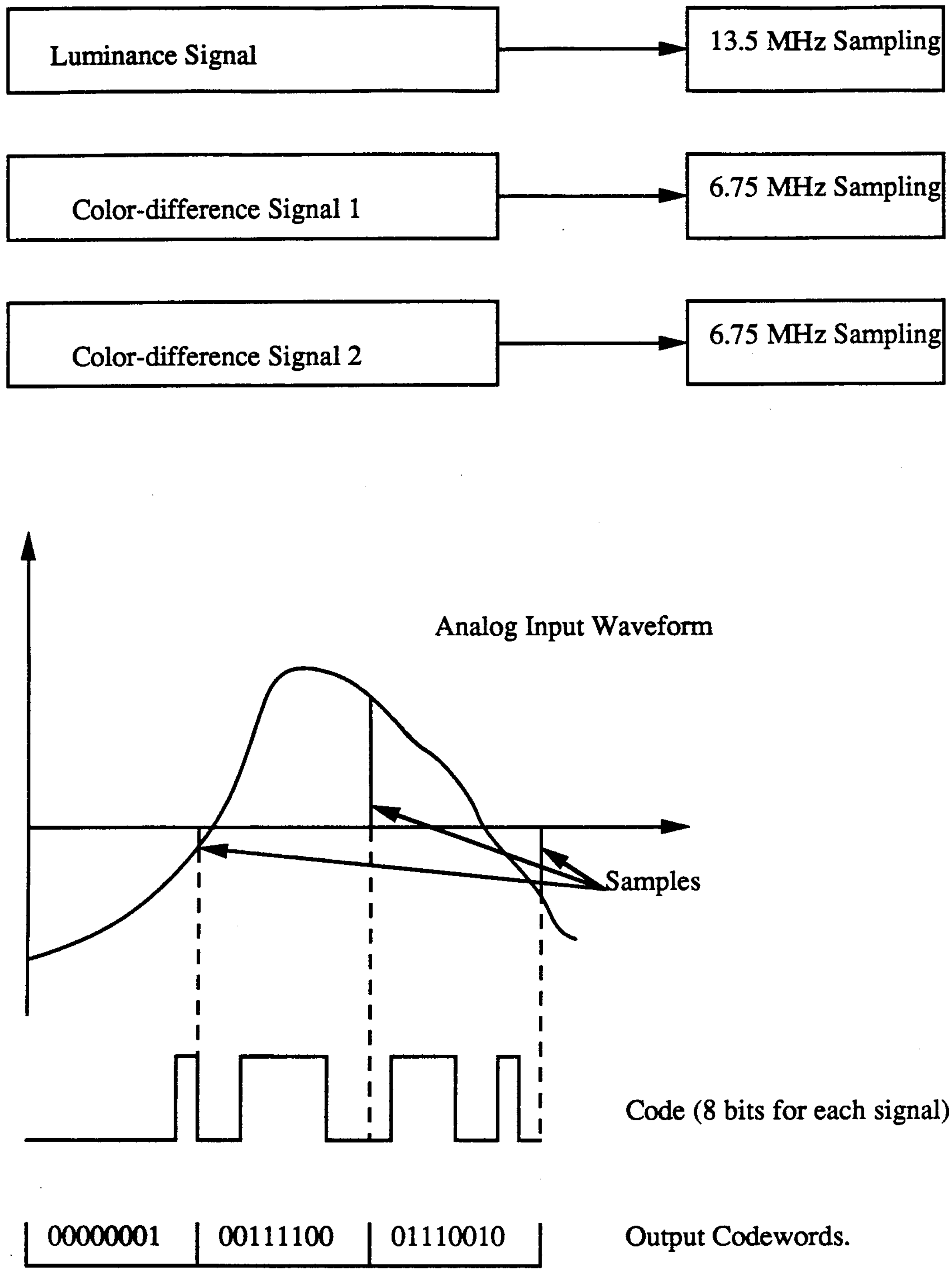
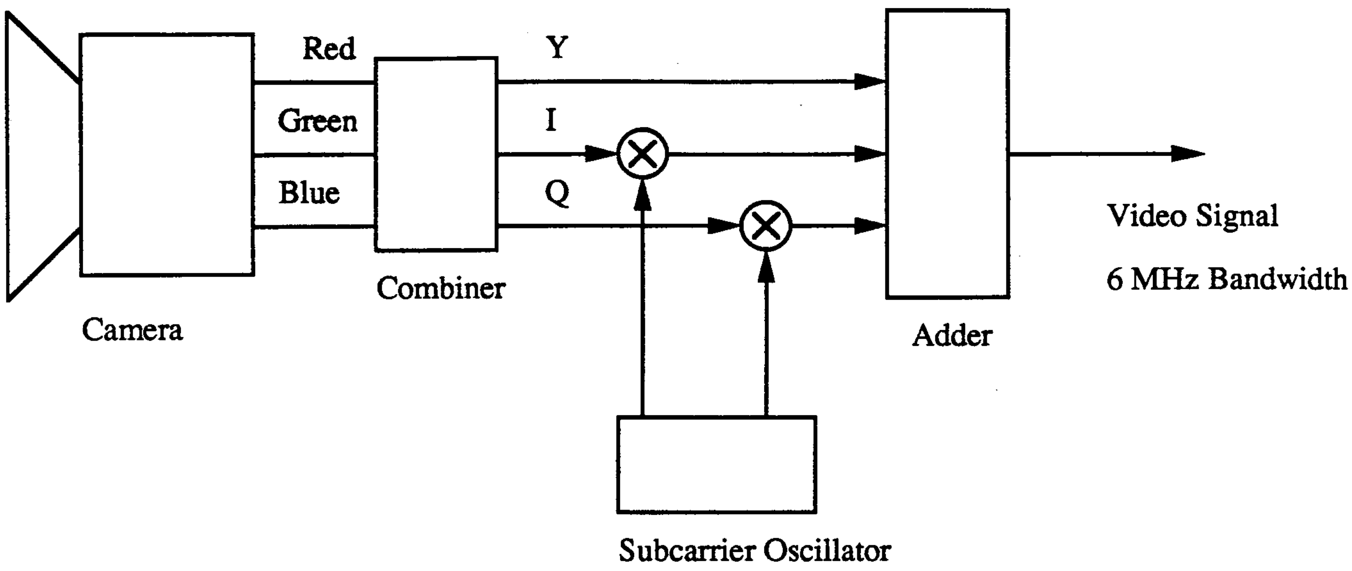


Figure 2.17 : Video Sampling and Digitization.

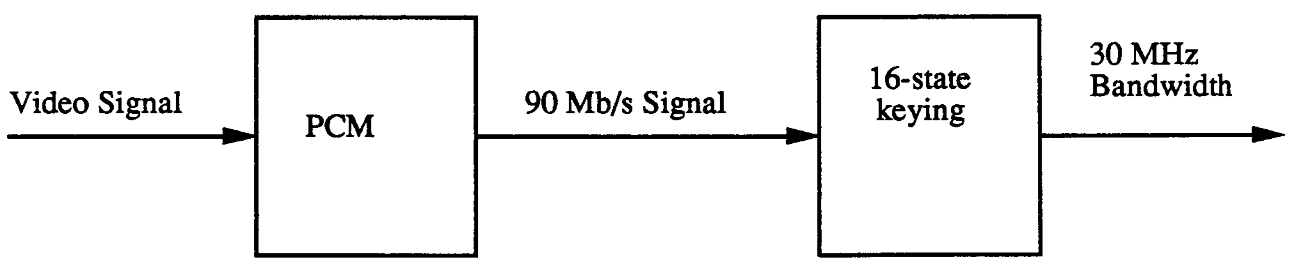


$Y = 0.3\ R + 0.59\ G + 0.11\ B$
4.2 MHz Bandwidth

$I = 0.6\ R - 0.28\ G - 0.32\ B$
1.5 MHz Bandwidth

$Q = 0.21\ R - 0.52\ G + 0.31\ B$
0.6 MHz Bandwidth

Bandwidth of Television Video.



Bandwidth Necessary for Digital Video Signal.

Figure 2.18 : Bandwidth of Video (Analog vs. Digital).

(1) Introduce Layered Architectures

CHAPTER 3

Multimedia Interfaces

There is presently an extensive body of techniques that allows for the interfacing with multi media systems. The development of the use of windows, the application of high resolution screens and the capability to transmit at Gbps rates allows for the implementation of much more sophisticated technology. In this chapter, we focus on all the interfaces to the multimedia elements, those that are direct to the end user and those that are more closely linked to the processors.

In the last chapter we focused on the different types of presentation alternative afforded to the end user and in the process developed sizing models for the different types of media objects. In this chapter we focus on the human interaction elements with a multimedia terminal. This interaction has two basic bounding elements. The first is the understanding and sophistication of the end user and the second is the complexity of the application that is being used in this part of the overall environment. Typically we desire that the end user have a minimal amount of training for the system, however, this often goes contrary to the desire to have a fully flexible design for the system. The second desire is to have all applications share a common context of use rather than developing a visual and tactile dissonance with the user. Such a goal may be desirable but doubtful ever reachable.

This in this chapter we develop and overall architectural paradigm for the interface to the multimedia system and then give some examples of interfaces that exist today as examples. We then extend the effort into the modeling of the interfaces and the modeling of the total end user source, including the effects of the image characterization. This modeling effort is a critical element in the analysis of the overall design of multimedia system since it helps answer the many system and human factors questions that have been developed over time.

In the paper by Hartson and Hix, they clearly state that the objective of interface design is not just to construct good interface but to develop environments in which good interfaces can be generated. In this chapter, we focus not on the interface as an end in itself but as a means to develop better interfaces. Thus our first efforts are directed at analyzing interfaces and their performance. We specifically develop models for interfaces so that we may determine their performance relative to a set of predefined performance criteria.

We then use that performance modeling as a tool to develop a synthesis of the the interface design, knowing the methodology of analysis. Interface design is almost a Marxian process or Hegelian dialectic, thesis or first attempts, antithesis or responses, and hopefully a synthesis of the key elements of a good design. To reach that however, we need to understand what constitutes a good design. Our approach is to focus on as many tangible issues that are possible.

3.1 Interface Architectures

Many works have been written on the human factors issues of man machine interfaces. The simplest rule however is the one that states that the interface should be intuitive and lead to a minimal amount of sensory dissonance. Sensory dissonance is the phenomenon that results when the human user is asked to do something in a way that is radially different that he has been accustomed to. A typical example of tactile dissonance is found in the area of operator service stations in the telephone companies where calls are made to obtain telephone numbers. The key entry devices are not in the standard QWERTY format but in an ABCDEF format. This was done after it was found that the incoming operators could not be required to have typing skills. The net result is that when an experienced person tries to use the system there is sever dissonance and the response time is dramatically reduced.

The development of interface architectures may proceed along many lines. Sutcliffe has described the Command Language Generation methodology with its four levels. These levels are:

o Task Level: The level at which we have input and output entries and we focus on what is being done. Specifically this level focuses on the need that are satisfied by the user in applying the interface mechanism.

o Semantic Level: At this level we deal with the meaning of the processes that we are performing and we deal with the underlying semantics of the tasks. Many elements of this approach deal almost with the Chomsky like analysis of language and the ability of the language to convey meaning.

o Syntactic Level: This relates to the formal relationship of the formal interrelationships between the symbols or words in the interface language or grammar. The syntactic level deals with the semiotic elements that focus on the general philosophical theory of signs and symbols and their use in natural languages.

o Interaction Level: This is the simplest level in that is focuses on the simplest manual entries of the end user. What key movements are made and how is the mouse moved to enter the data.

Thus we must consider developing an architectural construct for the multimedia interface. This construct should first start with a clear understanding of the requirements of the processes being executed and the sophistication of the user executing them.

3.1.1 Requirements and Specifications

The design of interfaces requires and understanding of the application and the level of expectation son the part of the end user. There are however several factors that are general in nature and are common in all user interface requirements. These elements are as follows:

o Stability: The interface operations should be stable in that actions on the part of the user do not lead to fault states that are unrecoverable. A typical stable interface requirement is one that sates that there shall be no set of inputs that can occur that would lead to a locking up of the system.

o Recoverable: This system should have the capability of providing a resolvable mode of operation that allows the end user the ability to return to where they came from without excess operations. Typical in this area is the use of the ESC key in PC applications.

o Consistency: This implies that the same result will occur if a transaction is entered in the same way no matter what state the system is in at the time of the transaction entry. It means that there is no level of ambiguity of response that should be anticipated by the end user.

o Non-Ambiguous: The interfaces requests for responses should be clear and unambiguous and the level of semantic complexity should be kept at a minimum. If two alternatives are presented to the end user then both alternatives responses should be equally stated in the presentation.

o Minimizable: This implies that the transitions needed to effect certain actions should be of minimum duration. The steps necessary to effect any transaction should not be excessively long.

o Extensible: The system should be such that all states are reachable\e from any other state,.

3.1.2 Elements and Alternatives

The architectural models of the interface that we are developing in this sections are composed of several layers of elements. We can consider these to fall into two general categories. They are the static and dynamic characteristics.

The static characteristic of the interface describe how the interface is viewed in a single interaction. If we look at Figure 3.x we see a common user interface screen for a typical application. This represents one part of the static environment.

Figure 3.x Static User Interface

We can look at the static user interface to be composed of the following elements:

o Input: This represents the method, means and technique to enter information into the system. It can range fro a mouse entry to a keyboard entry and include touch screen or even digital scanning devices. The entry technique depends upon the specific application, the nature of the end user and the environment.

A classic example of a total mismatch of input mechanisms is the attempt to put a mouse on the floor of an investment baking house trading station. In that environment, and device that is not tied down to withstand hurricane forces can and will become a projectile. Thus the first sets of mouse type entry devices found their way flying across many a trading floor. It sounded like a great idea for the UNIX designer but for the actual user was appropriate for other applications.

o Output: This represents the means, methods and techniques to retrieve information from the system. The output in the static environment represents the one time application of the information transferred to the end user.

o Presentation: The presentation element of the static design represents how the information is presented in form and design to the user for the purpose of inputting dat obtaining output or moving into a more dynamic mode of operation.

The dynamic elements are as follows:

o States

o Tasks

o Dialogs

o Transitions

o Interactions

3.1.3 Performance Issues

In evaluating interface systems, as we stated earlier in this chapter, we are basically developing models for the interface and showing how the interfaces relates to and performs within a larger and overall multimedia communications environment. Specifically, the interface system should ensure that a rapid and error free level of performance is attainable by the user.

Thus as part of the development of a multimedia source interface description, we need to determine the following types of performance measures:

o Source Rate: The source rate of the total interface can be generated from the combination of source characterization of the complex images as well as the models of the source within the context of the human interface.

o Response Time: The time between requests and responses is termed the response time. Thus the time that the user requests an image and the time that the image is delivered is one measure of the response time. Another, more general abstraction, is the response time that is from the time the user begins the inquiry to the time the inquiry is complete. This is more than the time from the request of image to response of image. The image is only part of the inquiry process. We must be capable of modeling and accounting for the entire inquiry process, including but not limited to that of a specific multimedia image.

3.2 Presentation Interfaces

There has been a significant development of various presentation interfaces over the past few years. This development has focused on the need for a set of standard and portable interfaces and development environments that can be used for a wide variety of applications. The driver for this has been in many cases the world of UNIX and C. UNIX is fundamentally an operating system that was developed in response to the need for a multiuser/multitasking environment that could function on a smaller size machine (PDP 7). C was the language that was developed to implement C and satisfied the needs of the developers as an elegant character and string manipulation medium. Neither was developed in the context of dealing with less sophisticated end users and moreover neither envisioned the growth of sophisticated display and I/O devices. Thus it has become necessary to develop overlays to these basic elements of the computing world.

This sections details several of the more common examples and presents them in the layered architectural context that we had developed in the previous section.

3.2.1 X Windows

The X Window interface is a current example of the end user interface development capability that is available for use on many platforms. X Windows was a joint development effort that was done at MIT for the purpose of running on top of UNIX type operating system and affording the C programmer a more effective end user interface capability. X Windows provides and interface to the applications program and in connected by calls to the X Library of calls. The text by Schleiffer, Geddys and x provides significant detail of the operations of the X interface.

X is en environment, an environment to develop the user interface. It is not itself a user interface. It is a fairly complex environment that allows for the interfacing not only with the user but with the applications program and the network of other users in the system as well as complex form of multimedia information.

X is composed of the following sets of elements. They are:

o Application: The application is the end user program that is intended for use by the user for effecting several transactions on the system.

o X Server: The X server is a program that is run on the users machine or another machine but it acts as the intermediary between the applications program and the handling of input and output to the display. The X Server is the key X ingredient as an operative agent in the communications network.

o Workstation: This is the physical device that an application is run on. Frequently in the X environment, there is one X Server per workstation.

o Clients: The clients are all the users on the X environment and may usually be represented by other applications programs.

o X Library: This is a set of functions presented in the form a X primitives that support the X windows applications. The X Library (X lib) is structured to support a specific C language environment.

o X protocol: This is a set of tools that allow a single user to provide other users interfaces to be independent from other clients.

o X Window Manager: This provides for the management facility for the window layout and interaction and assists when many windows are in action.

o X tool Kit: This is a user interface subroutine library that does certain complex tasks using already written X code that employs the primitives from the X lib.

o Events: These are time stamped results of users actions such as the entry of a key stroke or movement of a mouse.

Figure 3.x depicts the relationship of the various X elements of the architecture.

Figure 3.x X Architecture

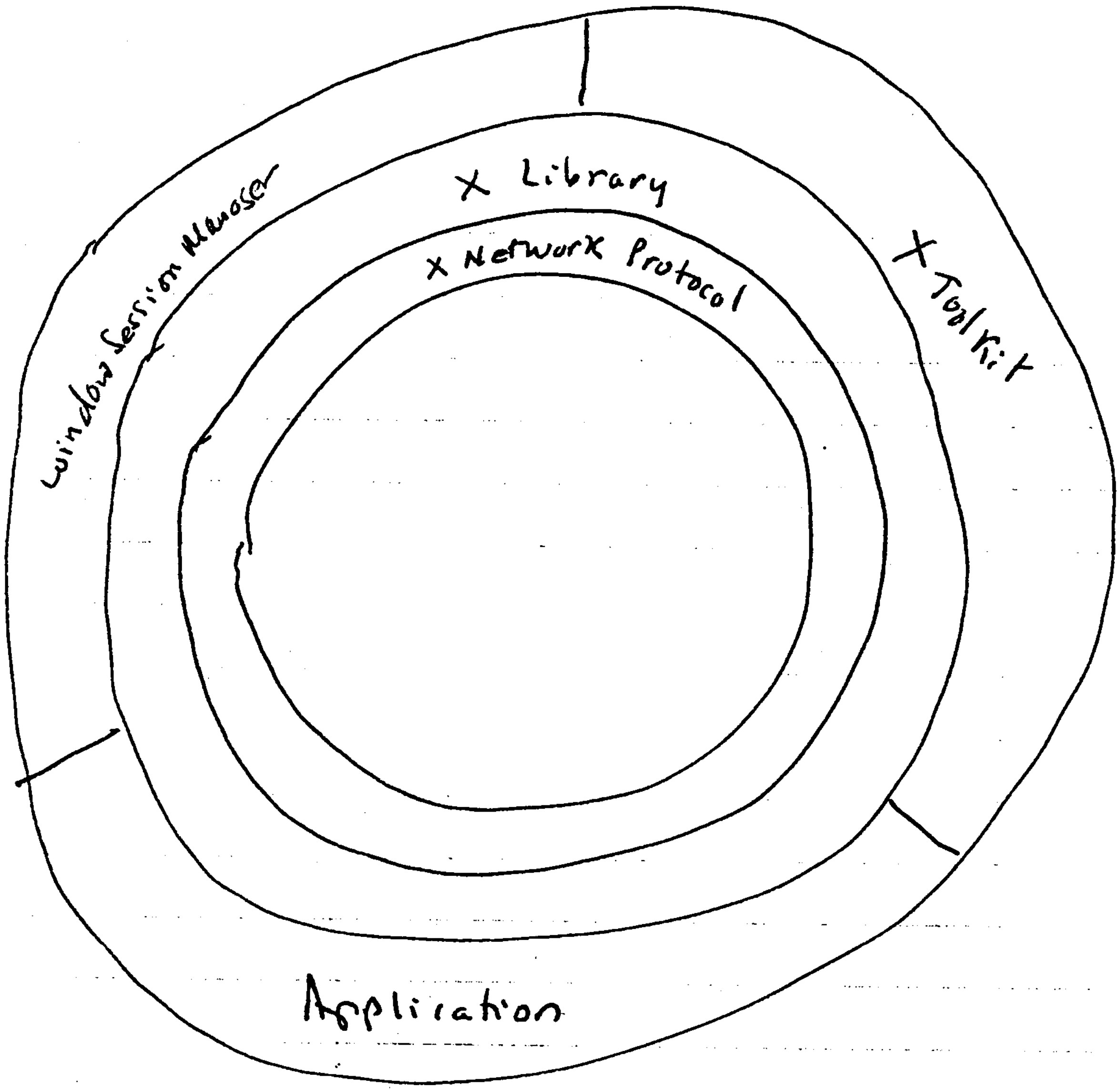


Fig: Overall X Architecture

Figure 3.x depicts the overall resource architecture available in the X environment. This resource architecture allows for the access of such elements as:

- o Windows
- o Graphics Contexts
- o Fonts
- o Color Maps
- o Pixel maps
- o Cursors

Figure 3.x X Resource Architecture

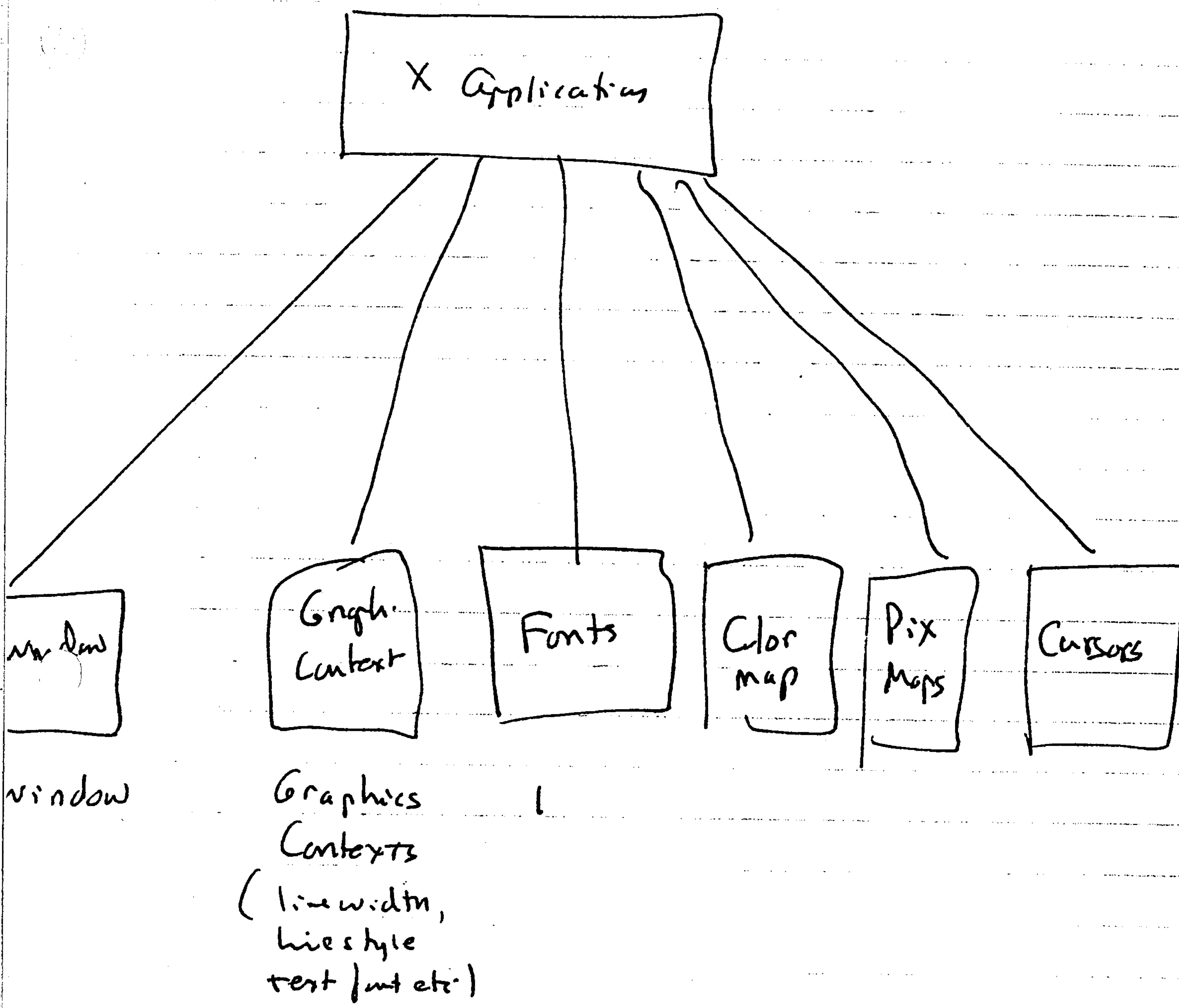


Fig X Resource Architecture

A typical X systems architecture is shown in Figure 3.x. This shows the relationship of the workstations and their related X servers and shows how this flow into the network through the X protocol. Using the X lib functionality, we see how X provide access to the other elements of the X architecture. The X protocol provides for the control over the flow of ll of the multimedia elements in the system. There is a significant disadvantage of this layered architecture. It is that the X protocol can often cause thrashing of the data elements as the system tries to display complex multimedia objects. In the present configuration, it is not common for the X protocol calls to file servers to take 15 to 45 seconds to display a high resolution image.

There are many types of servers in an X environment. Figure 3.x depicts several of these server types. They range from printer servers, database servers, name servers and communications servers. Each can be addressable within the X context.

Figure 3.x X Server Configuration

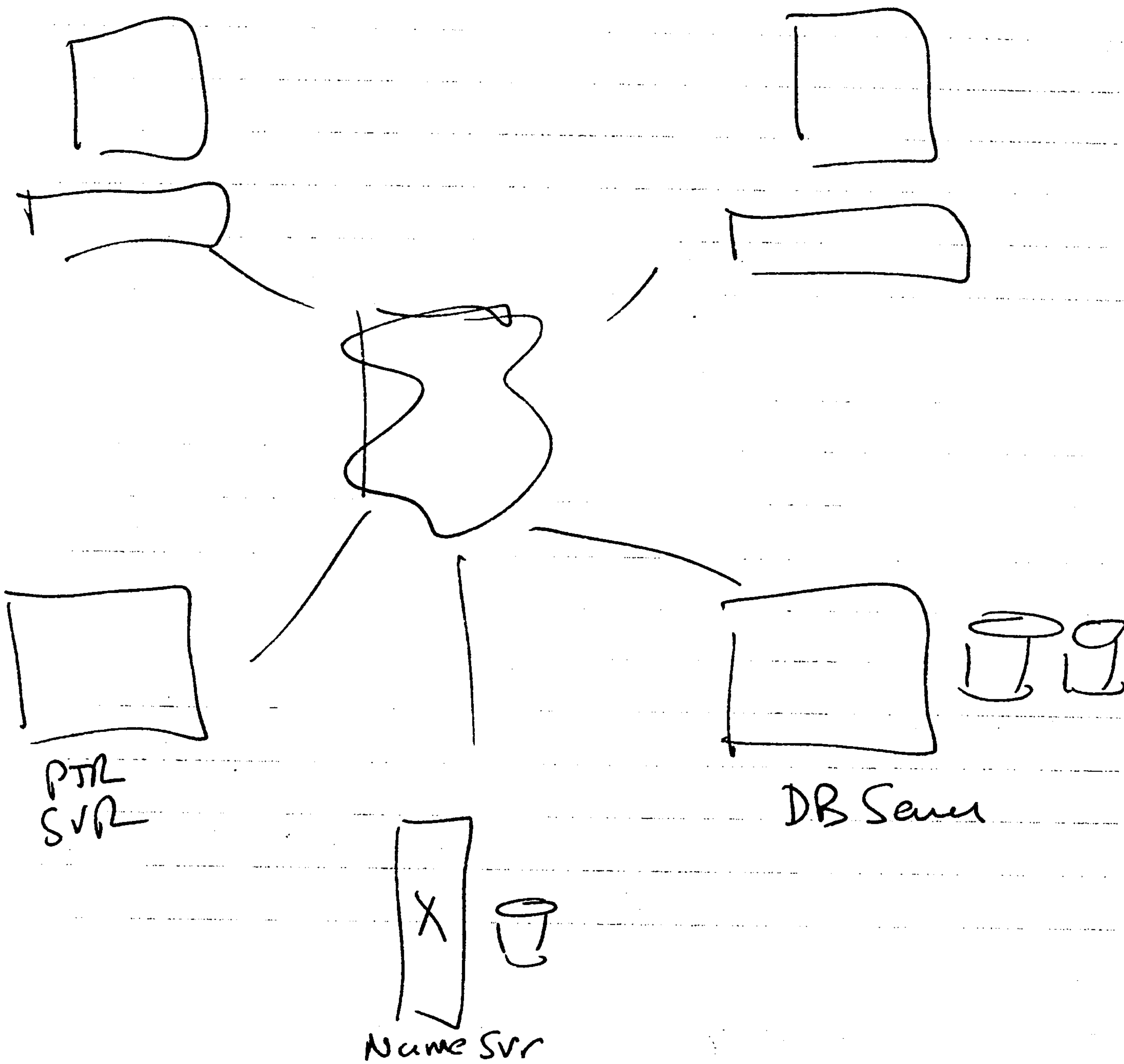


Fig   Servers (Ref. Ranade p.372)

the X window environment has a hierarchy of the windows themselves. The hierarchy is depicted in Figure 3.z. The hierarchy allows for the definition of a parent window, shown as W0 in this example. From this parent we can define a course of other windows that can be generated and manipulated.

Figure 3.x X Window Parent Hierarchy

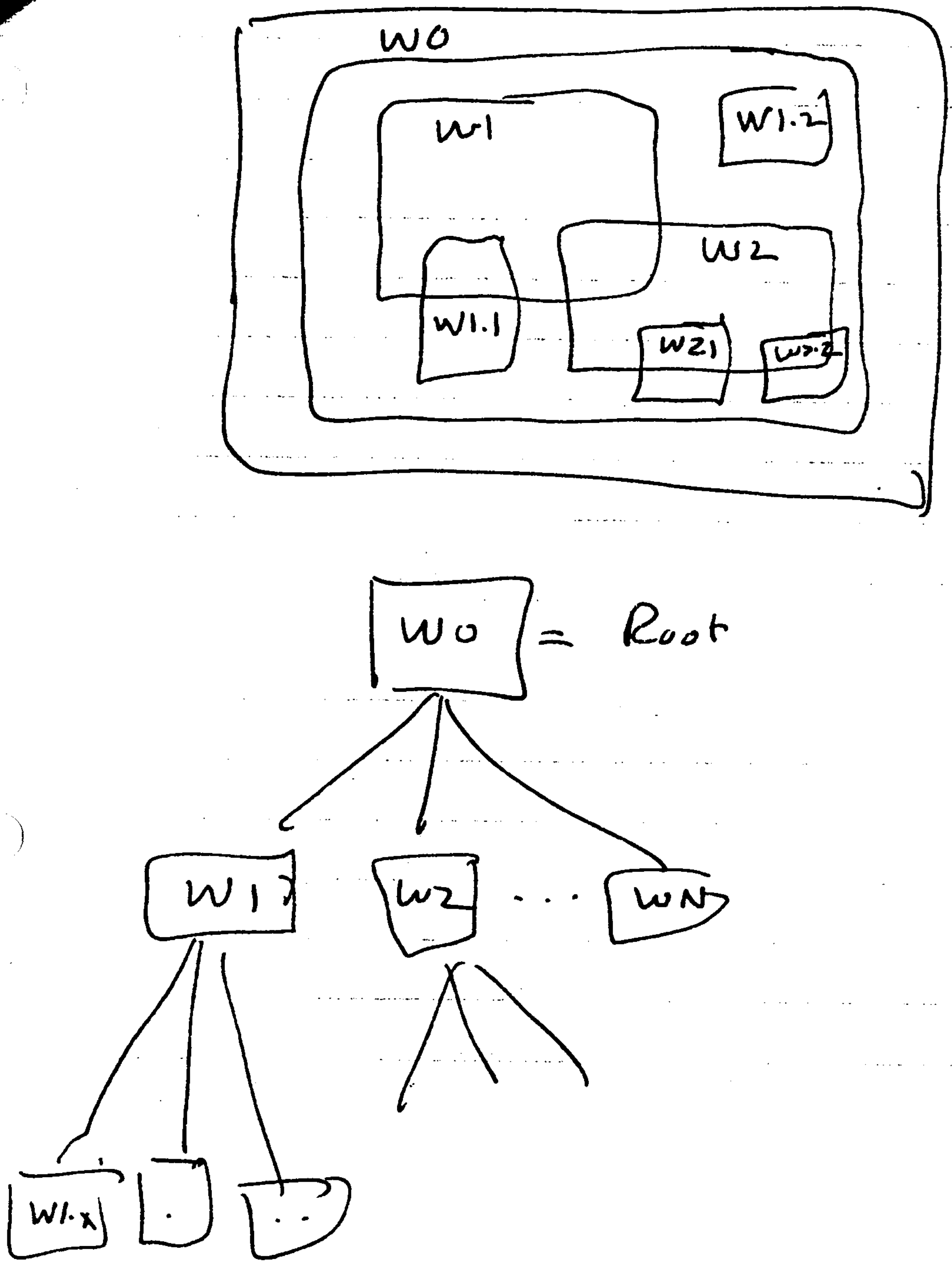


Fig X Window Parent Hierarchy

In Figure 3.x we depict the details of the X lib functionality. X lib is structured into the following nine major functions:

1. Display Functions

   o Opens displays

   o Places information onto the display

   o Closes the display

2. Window Functions

   o Create

   o Destroy

   o Map

   o Unmap

   o Attribute assign

   o Configure

   o Translate

3. Window Information Function

   o Obtain window information

   o Manipulate information

   o Change information

   o Manipulate selection

4. Graphics Resource Function

   o Create and destroy color maps

   o Allocate and modify colors

   o Read

- o Create and free pixel maps
- o Create, destroy etc graphics data

5. Graphics

- o Clear and copy areas
- o Draw points
- o Fill areas
- o Manipulate forms
- o Draw text
- o Transfer images
- o Manipulate cursor

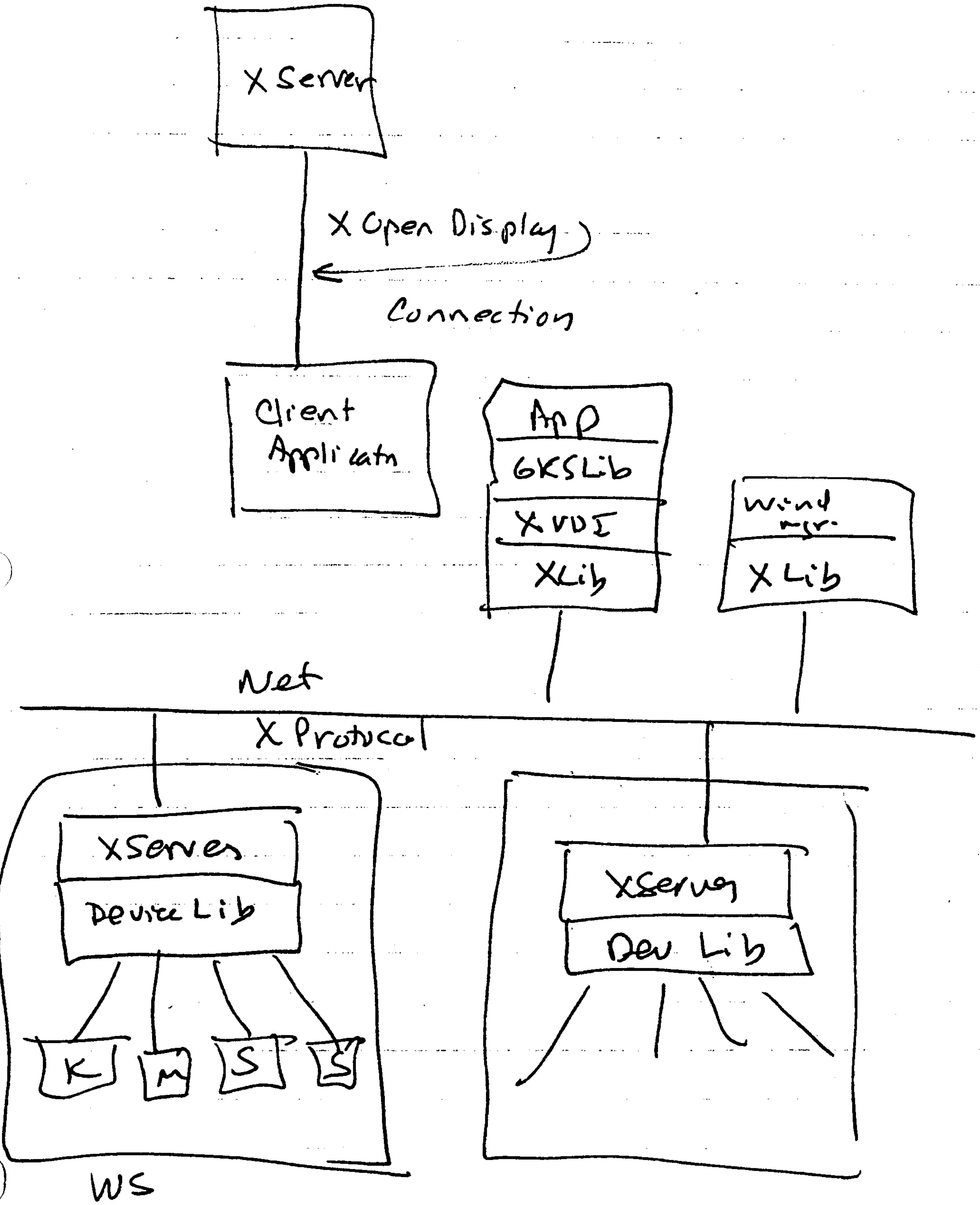
6. Window Manager Function

- o Change parent window
- o Control lifetime
- o Grab pointer, keyboard, server.
- o Control event processing
- o Control host access
- o Manipulate keyboard

7. Event Handling

- o Select events
- o Handle and queue events
- o Send and get events
- o Handle event errors

8. Predefined Property Functions



X Server
X Open Display
Connection
Client Applicatn
App
GKS Lib
X VDI
XLib
Wind mgr.
X Lib
Net
X Protocol
XServer
Device Lib
K
M
S
S
XServer
Dev Lib
WS

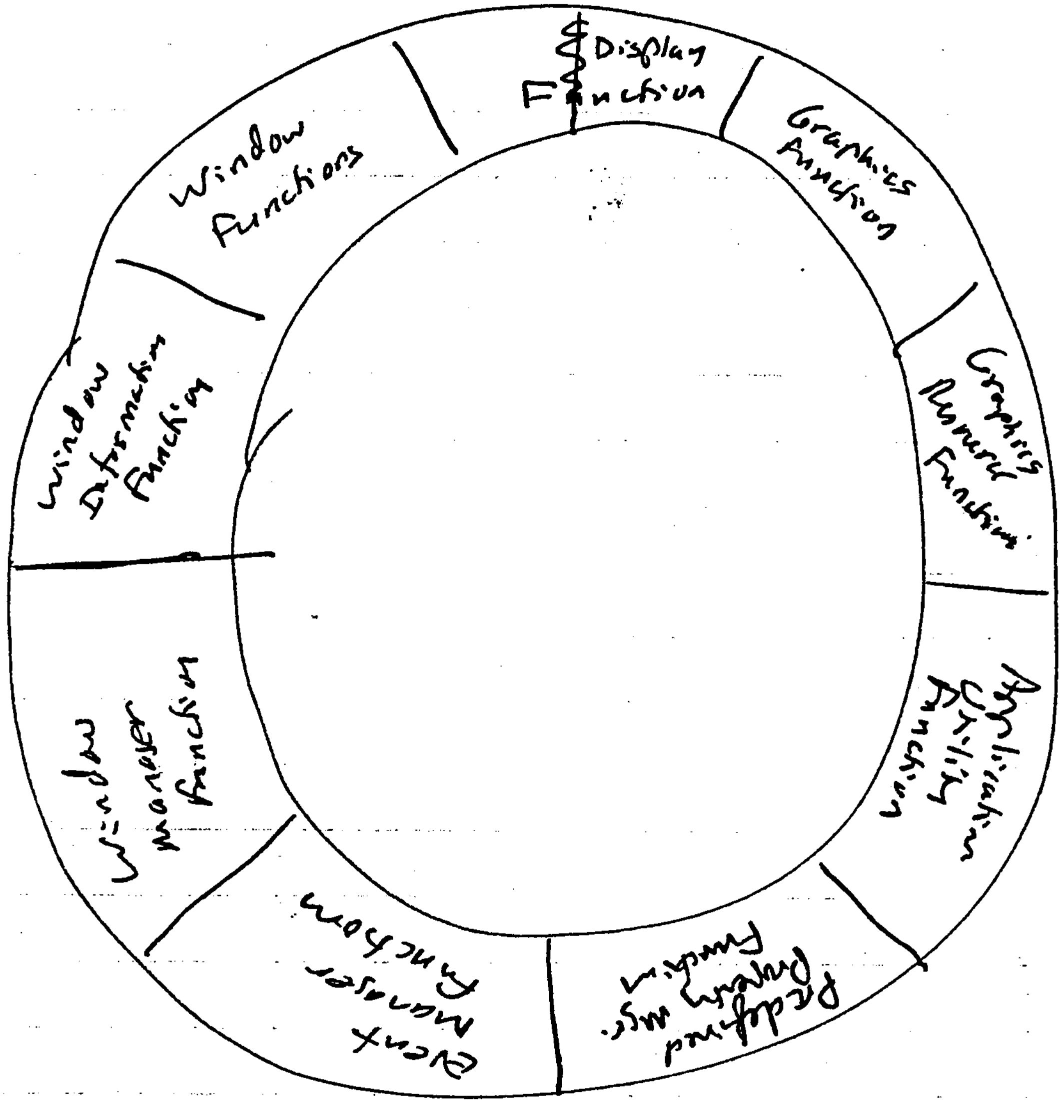


Fig X Wind S/W Architecture.

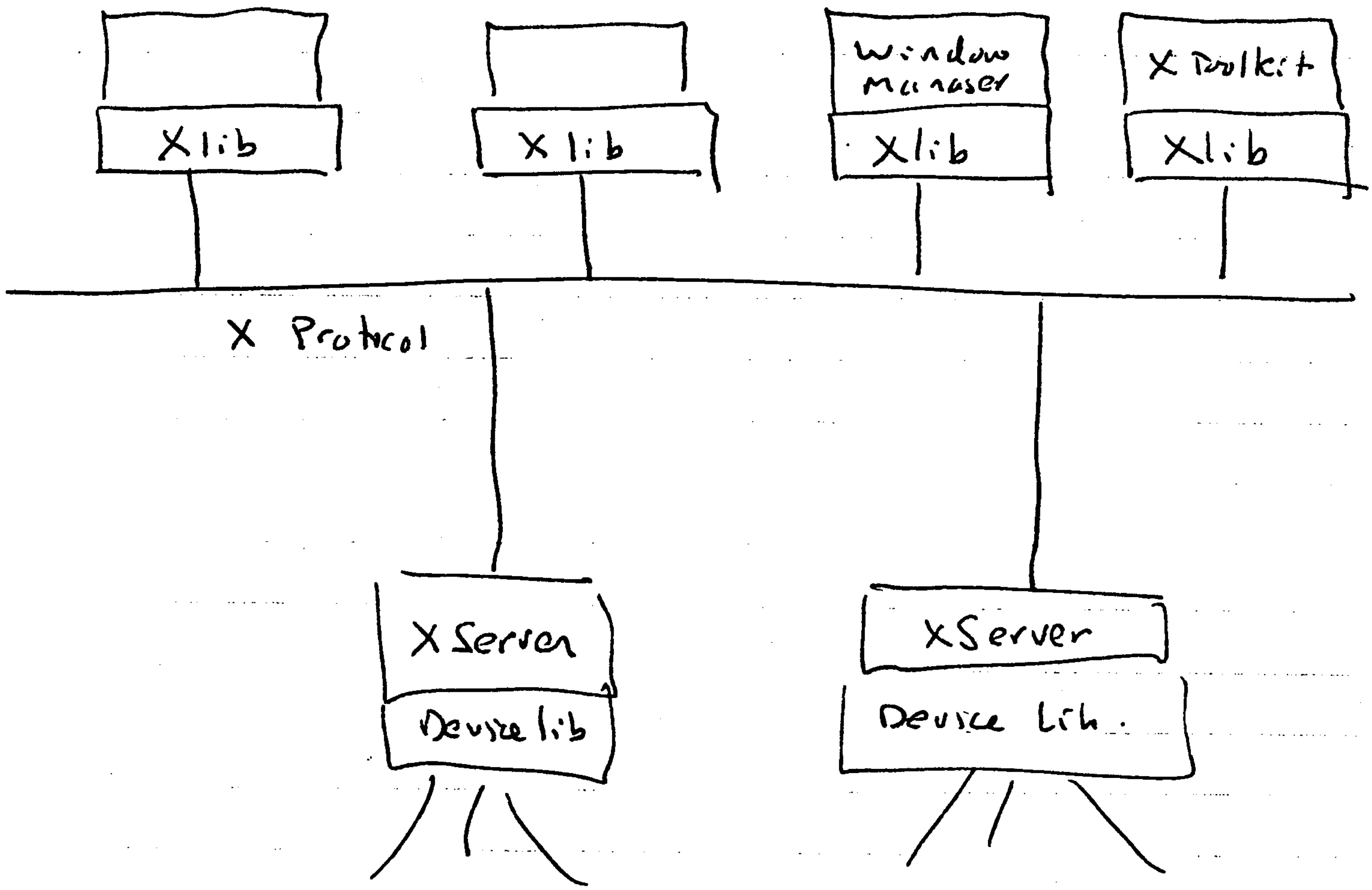


Fig X Architecture (ref Scheifler et.al p XX)

- o Communicate with window functions
- o Manipulate colormaps

9. Applications Utilities

- o Keyboard events
- o Generate and manipulate regions
- o Manipulate images and bitmaps

3.2.2 DEC Windows/UIL

DEC Windows and UIL are higher level protocols that allow for more rapid entry into the X environment.

3.2.3 Graphics User Interface (GUI)

We have seen that X windows and the more applications oriented support of DEC windows or UIL allows for the development and support of the end user interface. This has been formalized in the structure of the Graphics User Interface architecture (GUI) that shows how these elements relate to the hardware, operating system and ultimate the applications and the end user.

The general architecture of the GUI contains the following elements:

o Hardware: This is the standard platform on which the system will operate.

o Operating System: The standard operating system of choice. Frequently this will be UNIX, PS/2, VMS or DOS.

o Graphics System: This element is the actual graphic system used for the generation of the display graphics. X in X lib has a certain graphics system capability but is very limited as a user friendly graphics generator. PHIGS, GKS or PostScript has the better graphics capability.

o Window Environment: X is a typical window environment. The wind approach allows for the opening and closing of specific applications oriented windows that allow fore the focusing of the users interest and application.

o Applications Manager and File Manager: This is the applications and often platform specific display interface that allows for the direct interaction between the screen and the window manager.

o User Interface: This is the direct end user interface. OSF Motif is a typical example.

o Applications: This is the end user applications.

Figure 3.x depicts the overall GUI architecture displaying the relationship between the layers.

Figure 3.x General User Interface (GUI) Architecture

	DEC	SUN
User Interface	OSF/Motif	Open Look (Sun/open wind)
Applic Mgr. + File Mgr.	DEC windows	File Mgr
Window Environmnt	Xwindows	Sun/Open
Graphics System	X windows	PostScript
Operating Sys.	ULTRIX	UNIX
Hardware	VAX 9000	SUN 5

Fig

(Ref Seymour PC wld. p 100)

There are advantages and disadvantages with the types of use interface development tools that we have just been describing. They are:

o Advantages

o Provides a modular and transportable development environment.

o Disadvantages

o Adds significant operations overhead on the system thus introducing delays and other operational factors.

o Does not allow for customization and real time optimization of the applications.

o Requires significant memory allocations to store the library functional and other parts of the GUI elements.

3.2.4 MM Services

Having developed some of the issues related to the development environment of the end user interface, we can now focus on several specific end user interfaces and review how they meet the requirements and specifications that we have discussed. All of these interfaces have been developed in an X windows context and thus have been developed within the development context discussed in this section.

3.2.3.1 Session

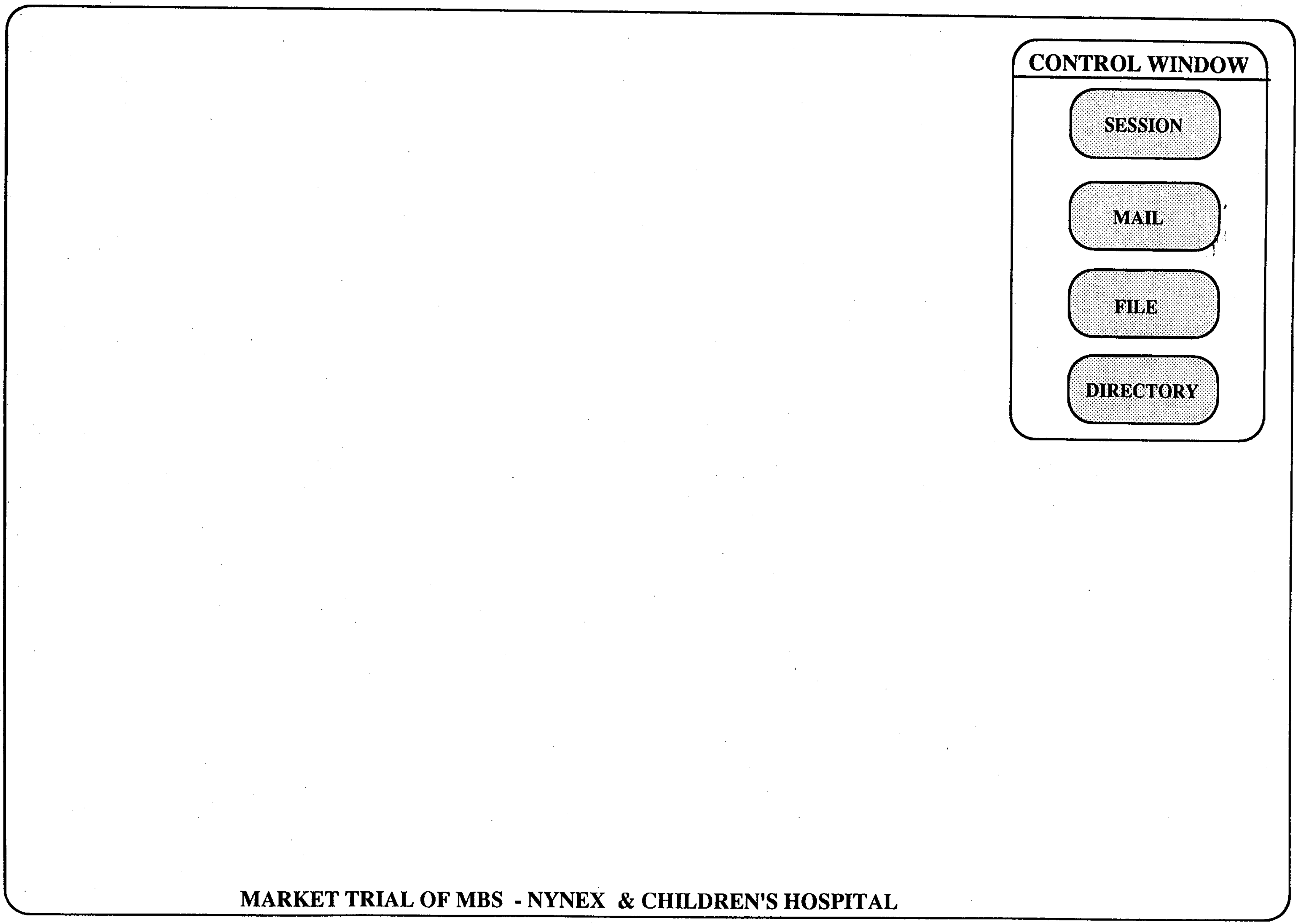
The session screens allow the end user to enter into sessions with other users, applications programs, or data bases. What we have shown is a simple set of directions for accessing and determining the specifics of the session operation.

In Figure 3.x we show the complete set of session screens that relate to this user interface.

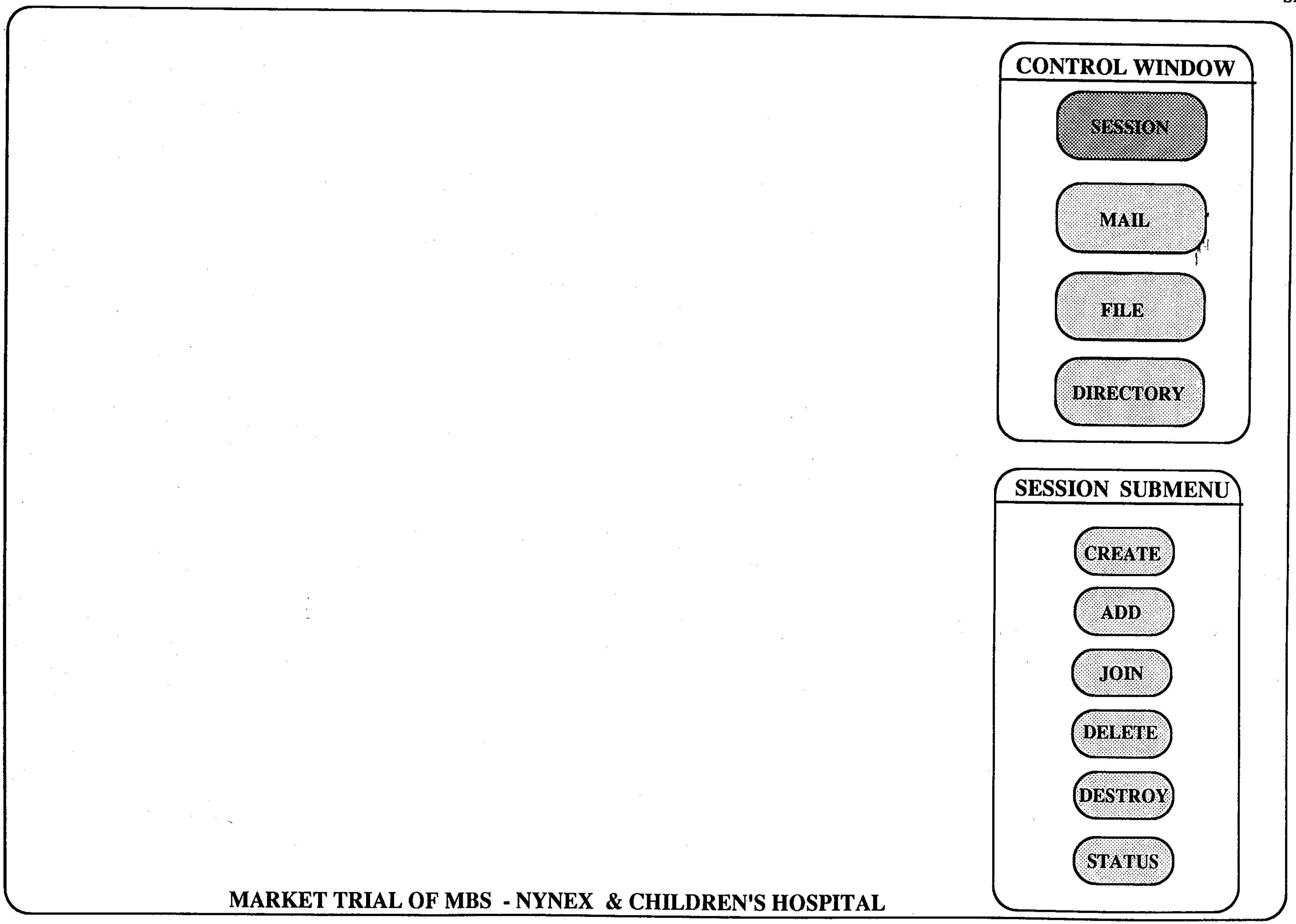
Figure 3.x Session Screen Layout

MBS INITIAL SCREEN WHEN WORKSTATION PROGRAM IS LOADED

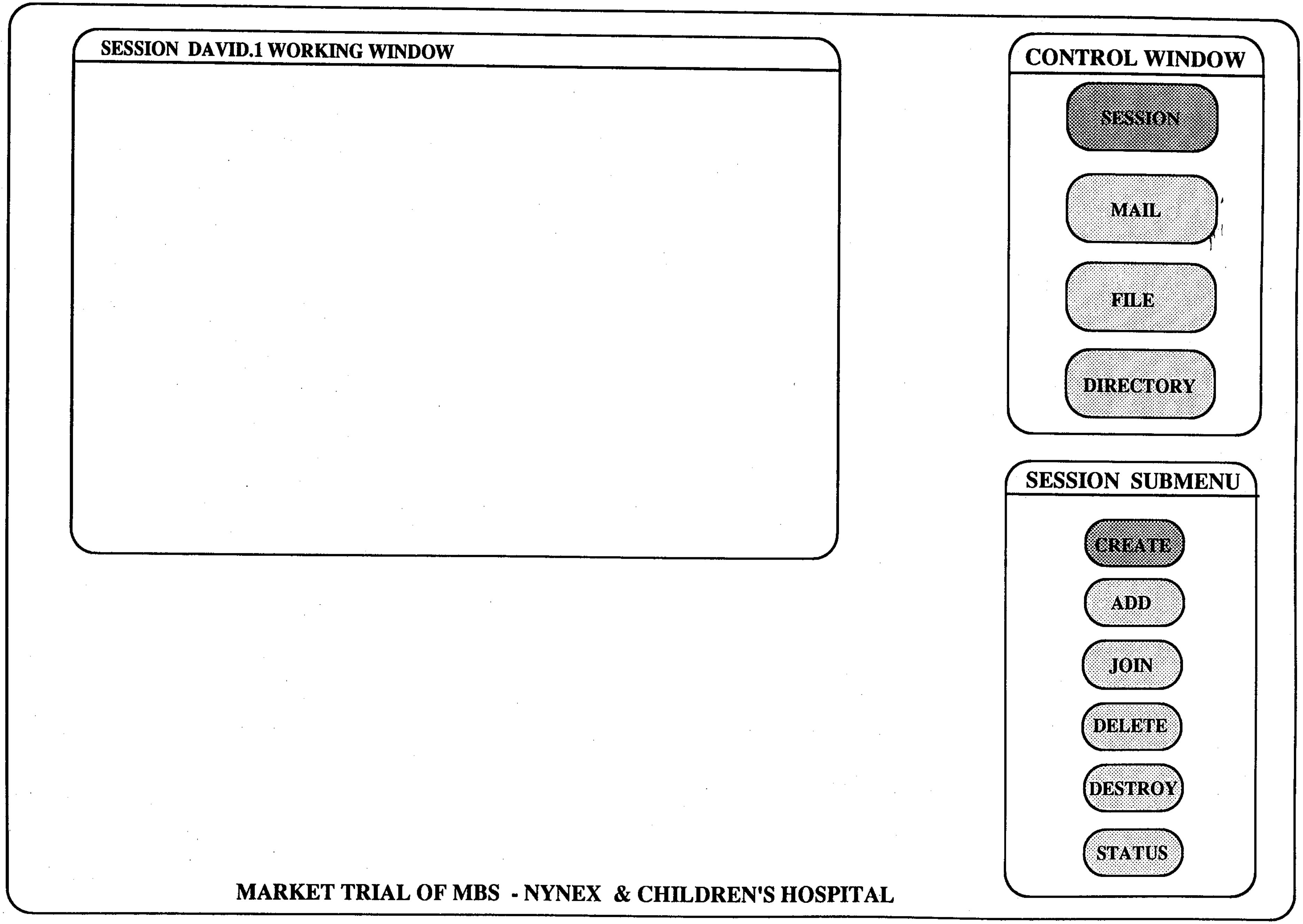
S1



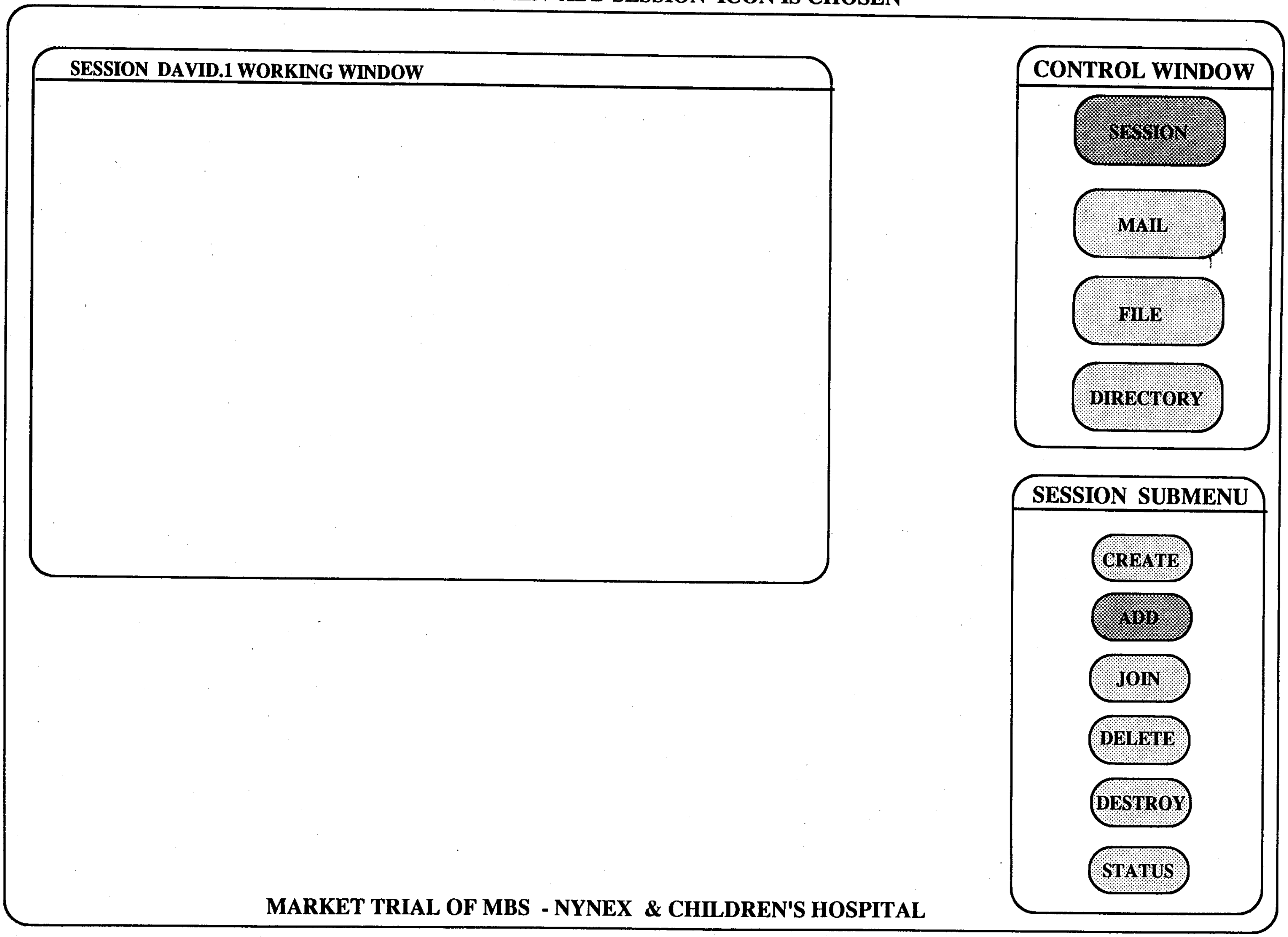
MBS SCREEN WHEN SESSION ICON IS CHOSEN



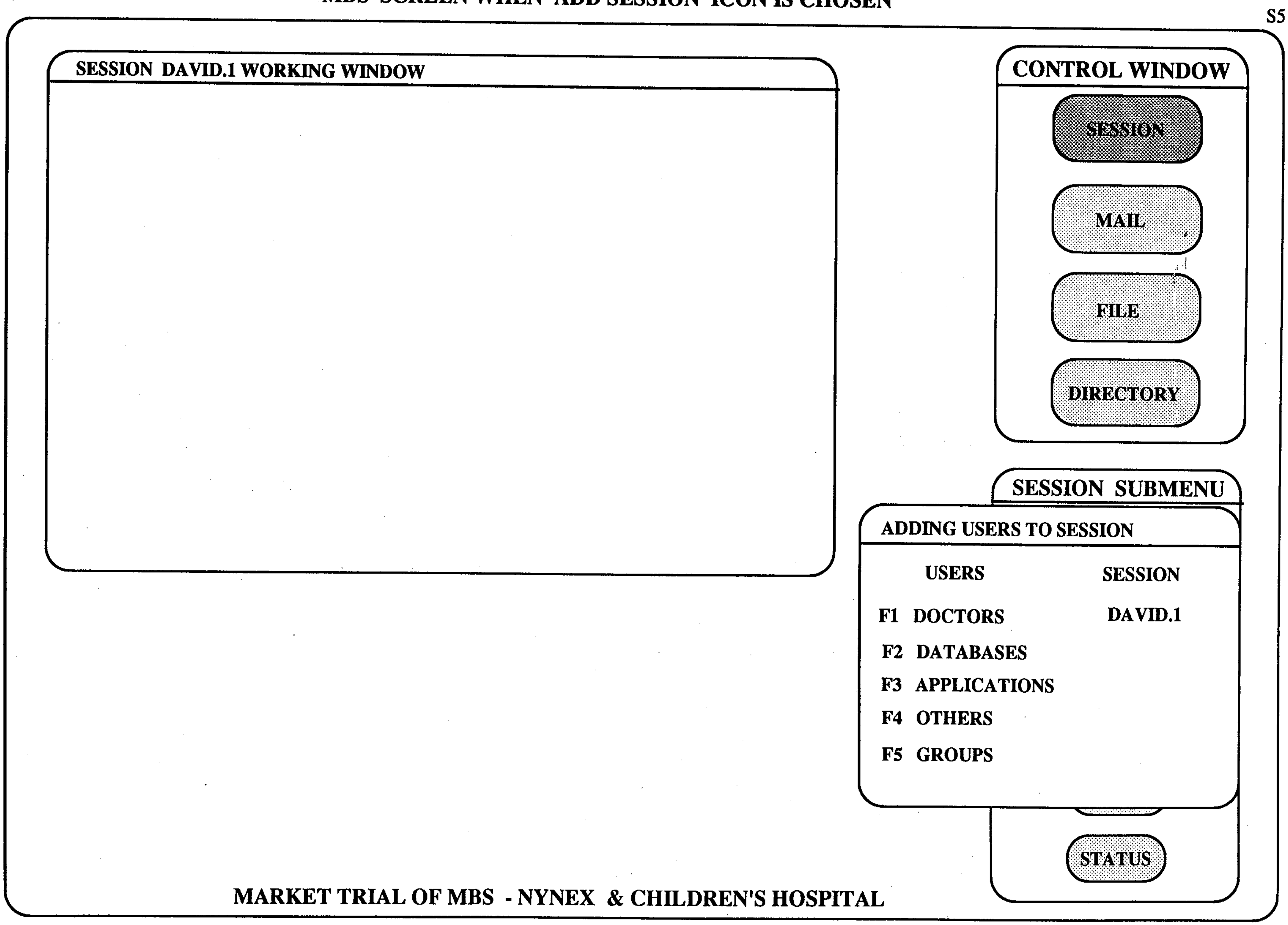
MBS SCREEN WHEN CREATE SESSION ICON IS CHOSEN



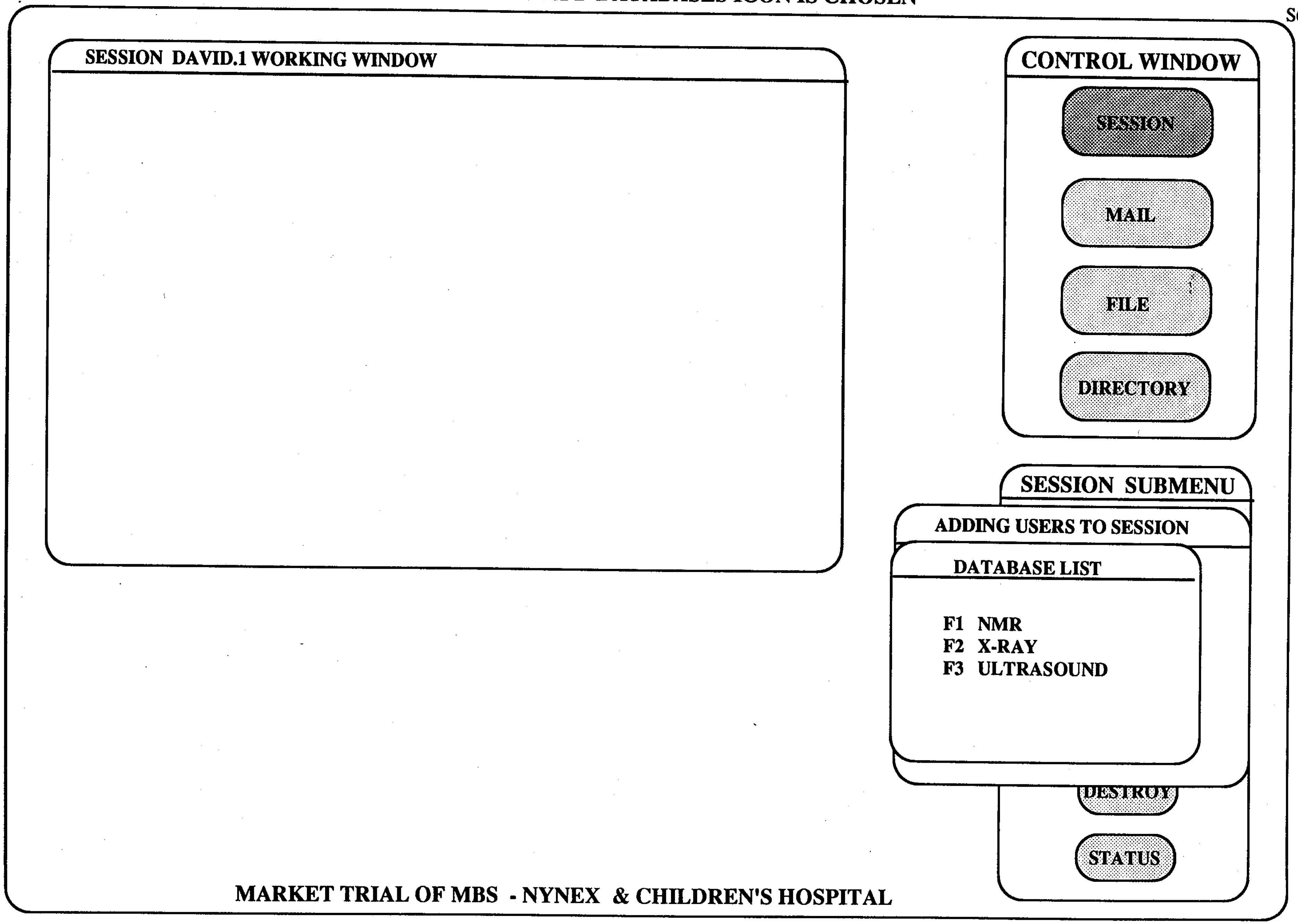
MBS SCREEN WHEN ADD SESSION ICON IS CHOSEN



MBS SCREEN WHEN ADD SESSION ICON IS CHOSEN

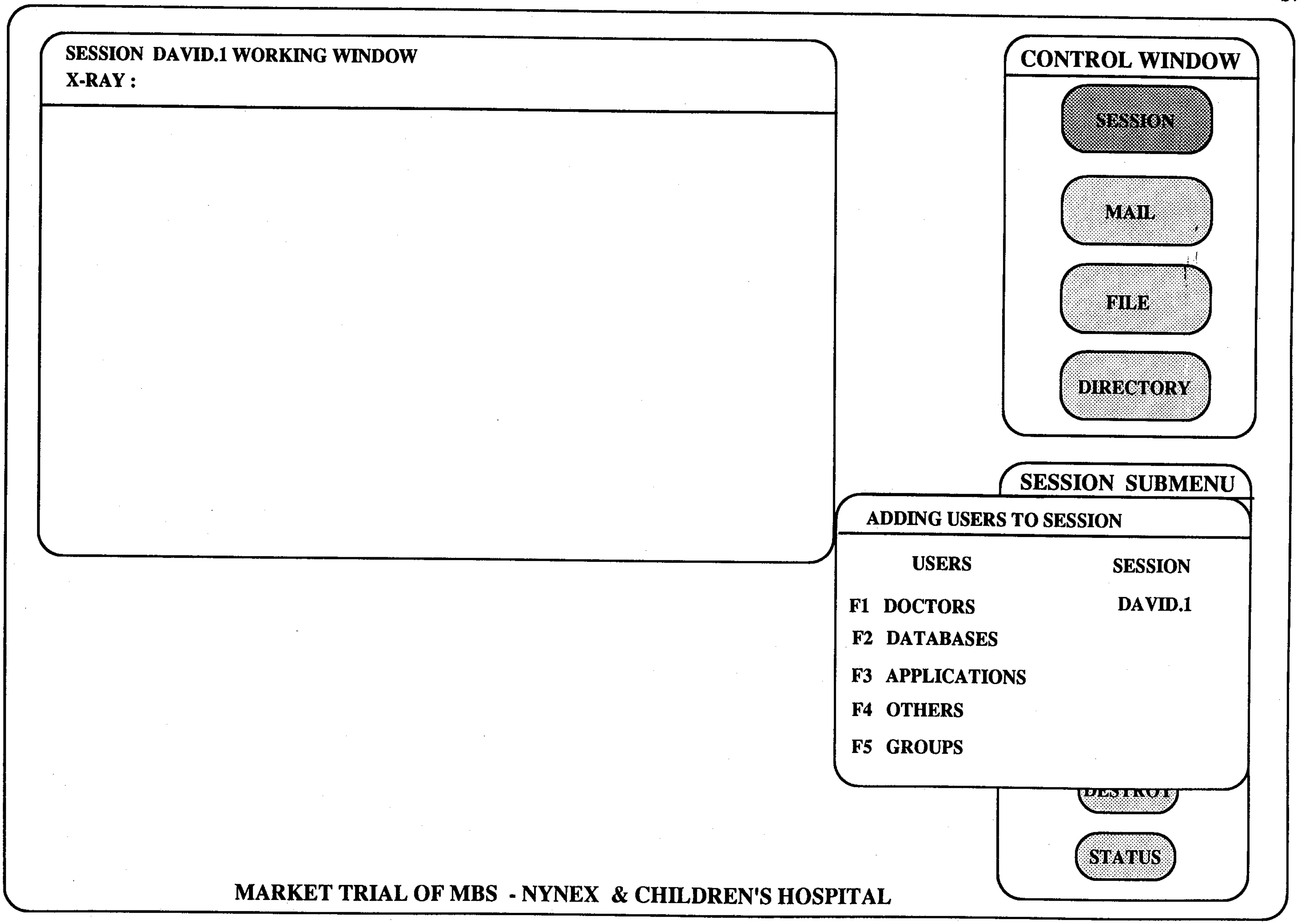


MBS SCREEN WHEN ADD DATABASES ICON IS CHOSEN

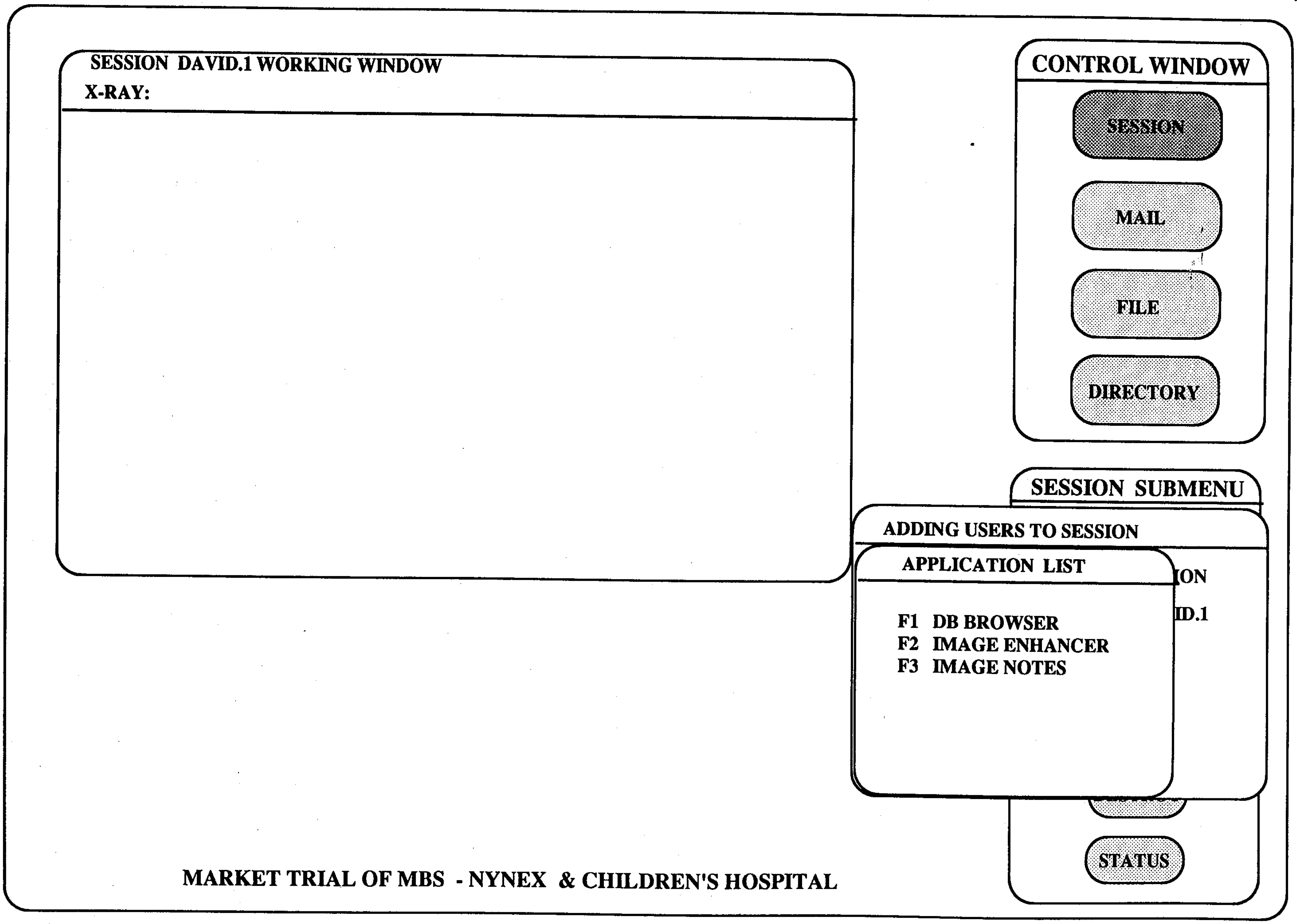


MARKET TRIAL OF MBS - NYNEX & CHILDREN'S HOSPITAL

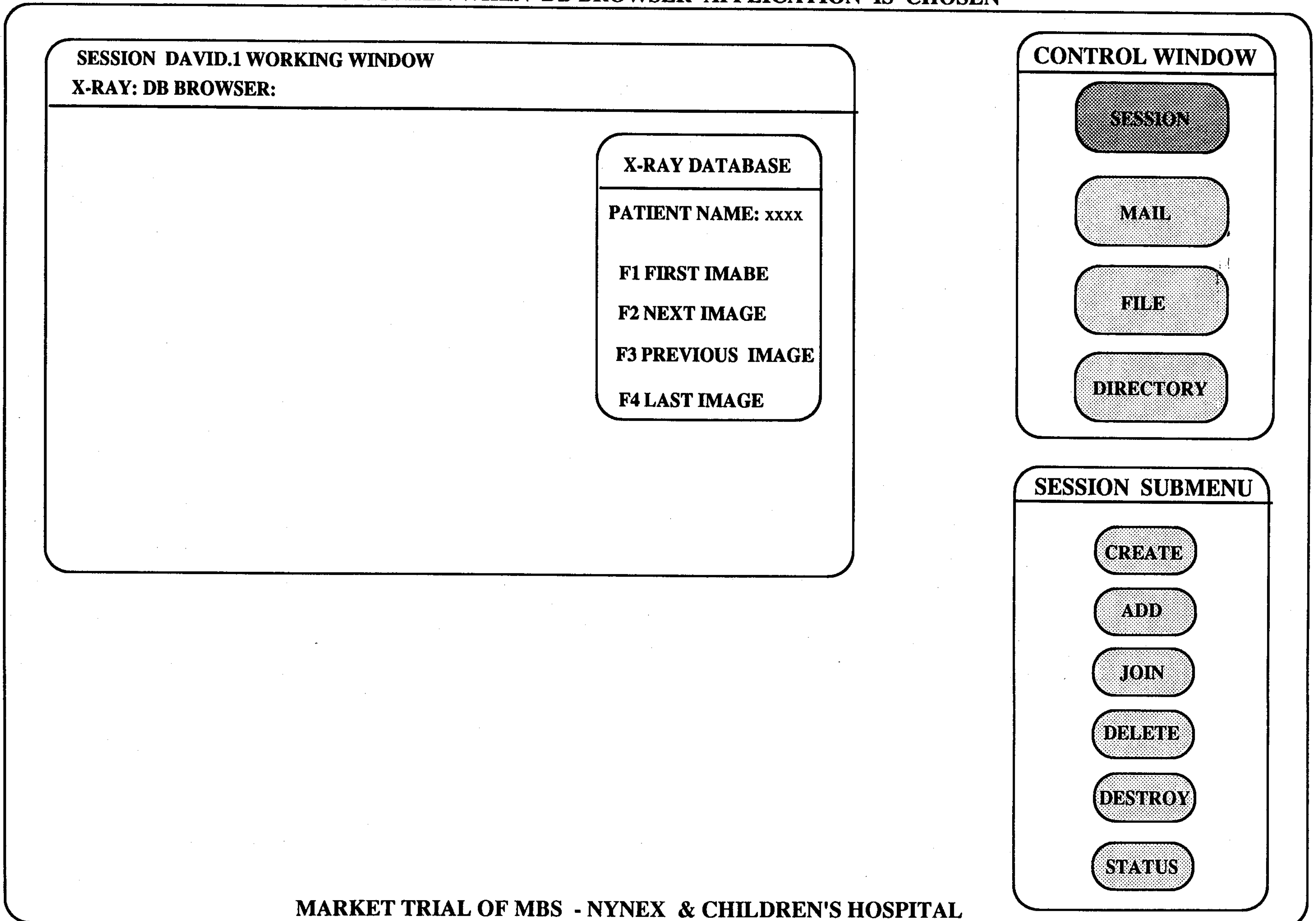
MBS SCREEN WHEN ADD DATABASES ICON IS CHOSEN



MBS SCREEN WHEN ADD APPLICATIONS IS CHOSEN



MBS SCREEN WHEN DB BROWSER APPLICATION IS CHOSEN



MARKET TRIAL OF MBS - NYNEX & CHILDREN'S HOSPITAL

MBS SCREEN WHEN FIRST IMAGE IS CHOSEN

SESSION DAVID.1 WORKING WINDOW
X-RAY: DB BROWSER:

X-RAY DATABASE

PATIENT NAME: xxxx

F1 FIRST IMABE
F2 NEXT IMAGE
F3 PREVIOUS IMAGE
F4 LAST IMAGE

CONTROL WINDOW

SESSION
MAIL
FILE
DIRECTORY

SESSION SUBMENU

CREATE
ADD
JOIN
DELETE
DESTROY
STATUS

MARKET TRIAL OF MBS - NYNEX & CHILDREN'S HOSPITAL

MBS SCREEN WHEN DR. MARGULIES WANTS TO ADD A USER

SESSION DAVID.1 WORKING WINDOW

X-RAY: DB BROWSER:

X-RAY DATABASE

PATIENT NAME: xxxx

F1 FIRST IMABE

F2 NEXT IMAGE

F3 PREVIOUS IMAGE

F4 LAST IMAGE

CONTROL WINDOW

SESSION

MAIL

FILE

DIRECTORY

SESSION SUBMENU

CREATE

ADD

JOIN

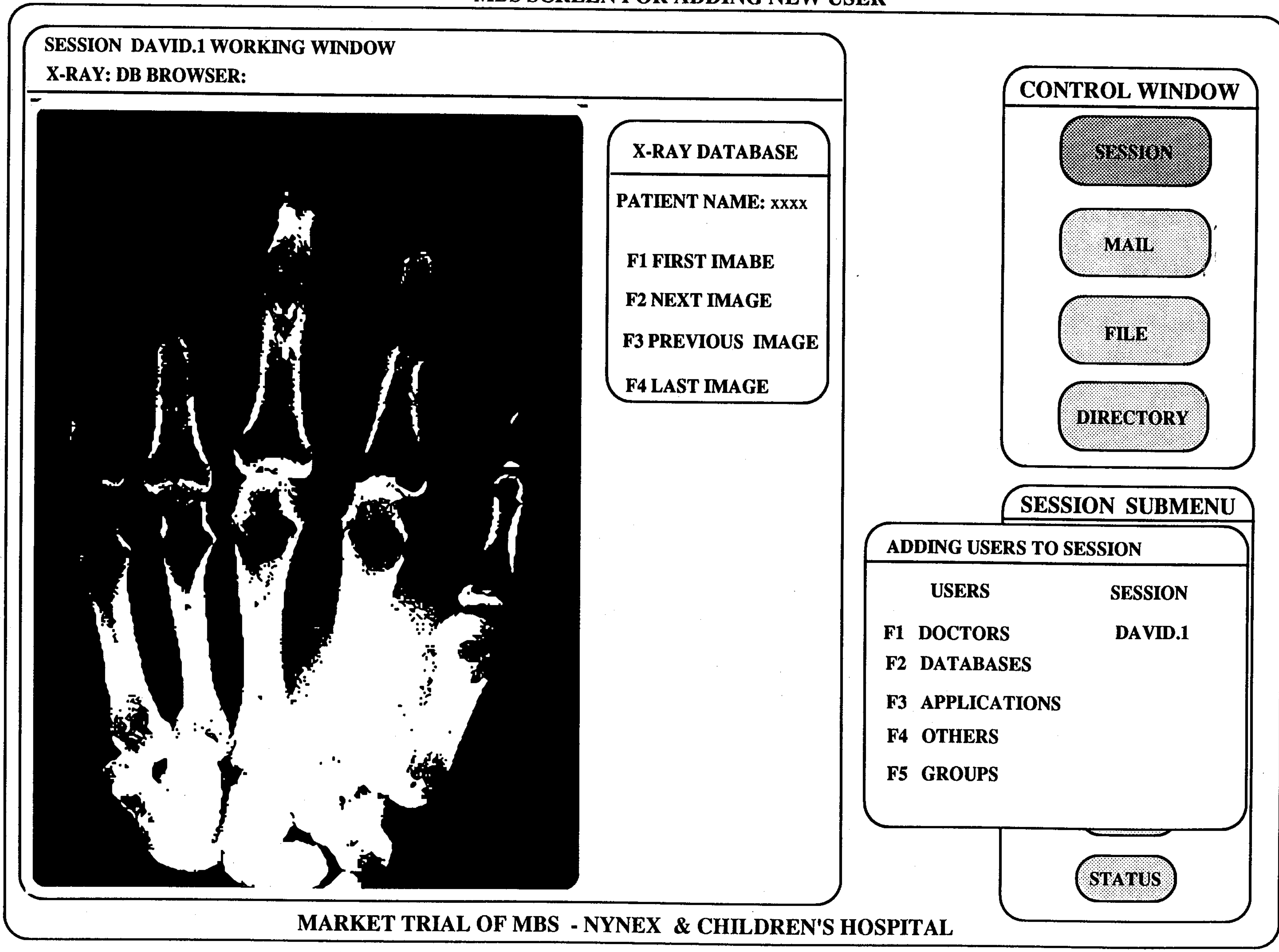
DELETE

DESTROY

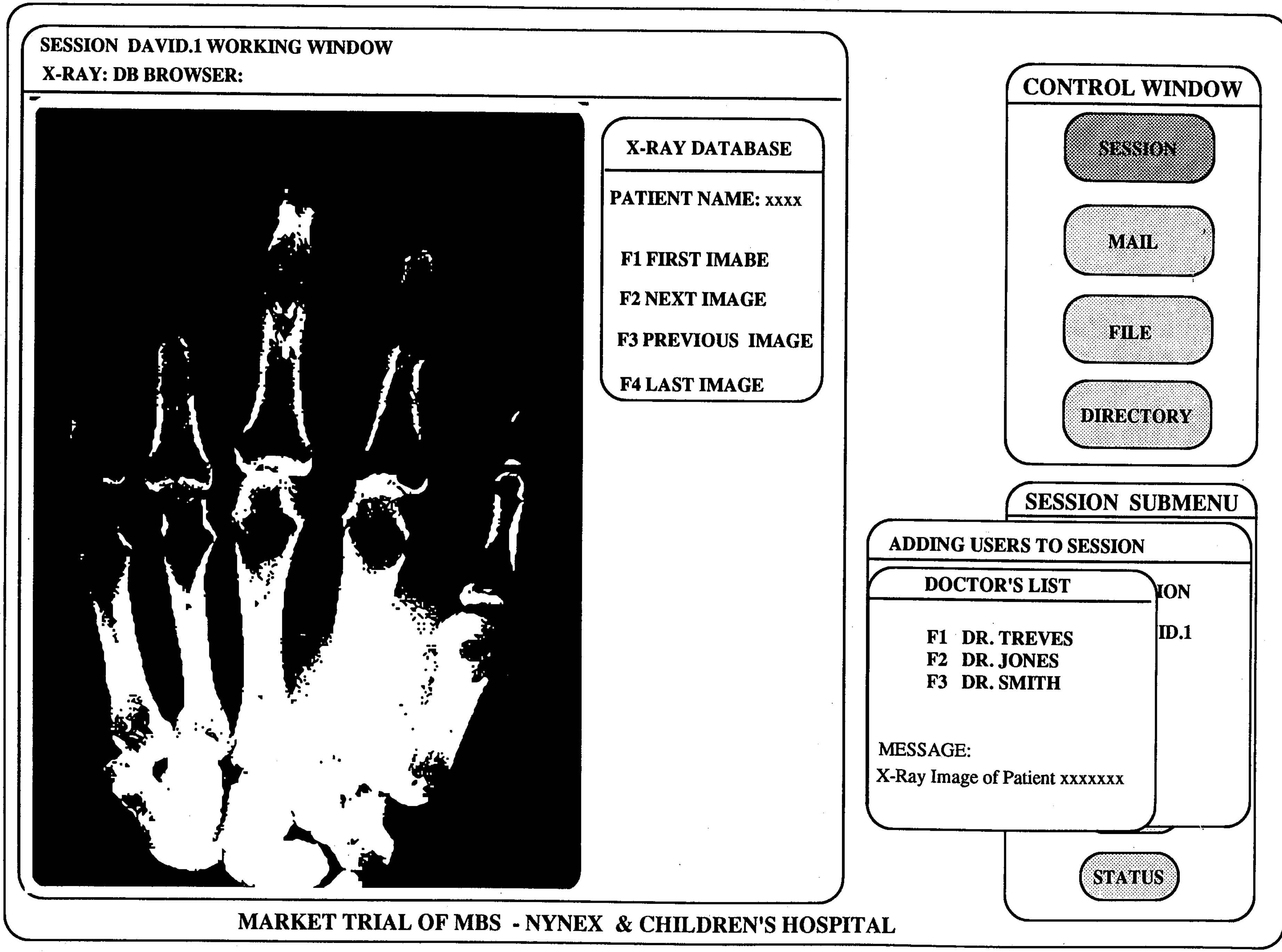
STATUS

MARKET TRIAL OF MBS - NYNEX & CHILDREN'S HOSPITAL

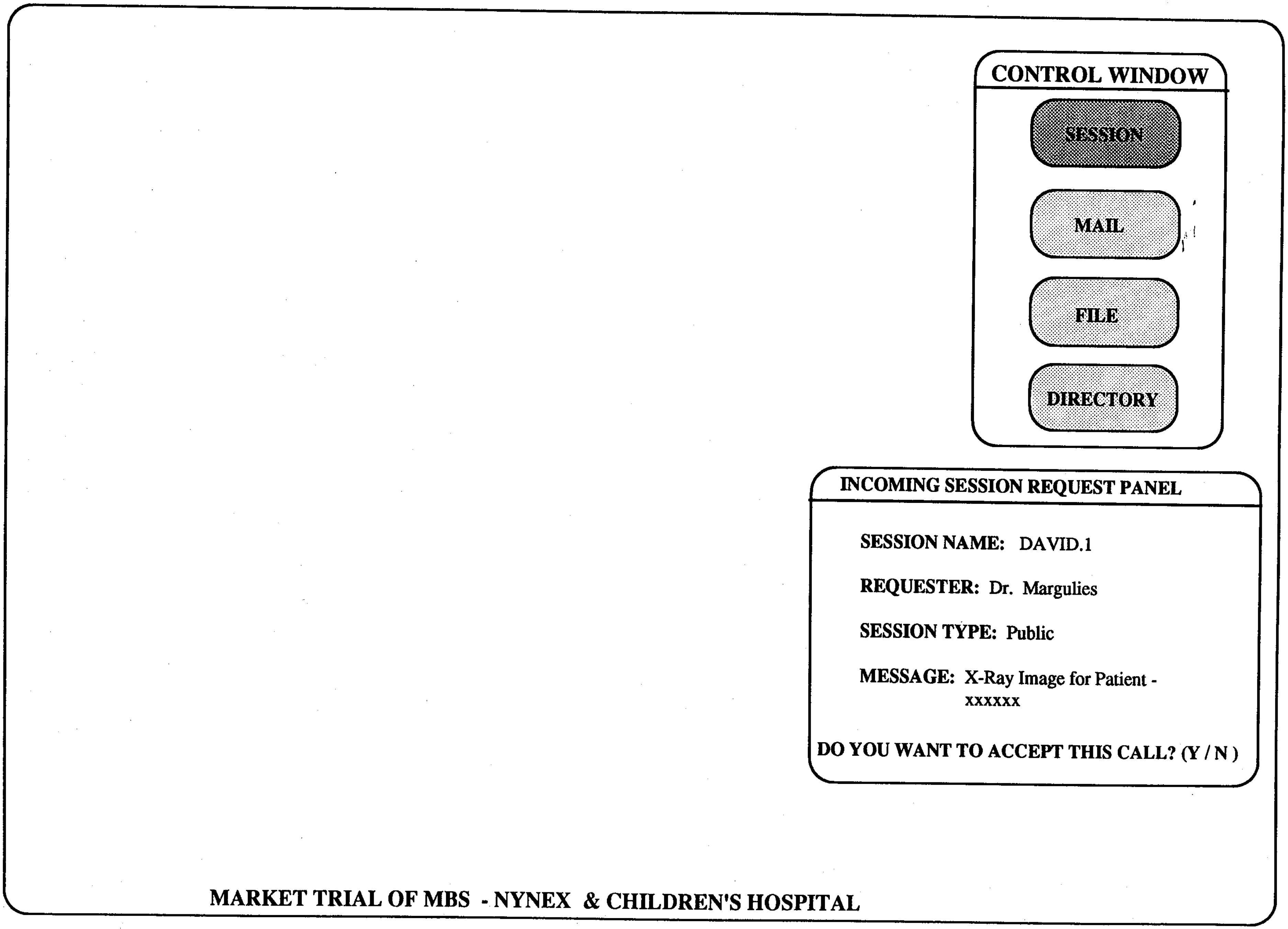
MBS SCREEN FOR ADDING NEW USER



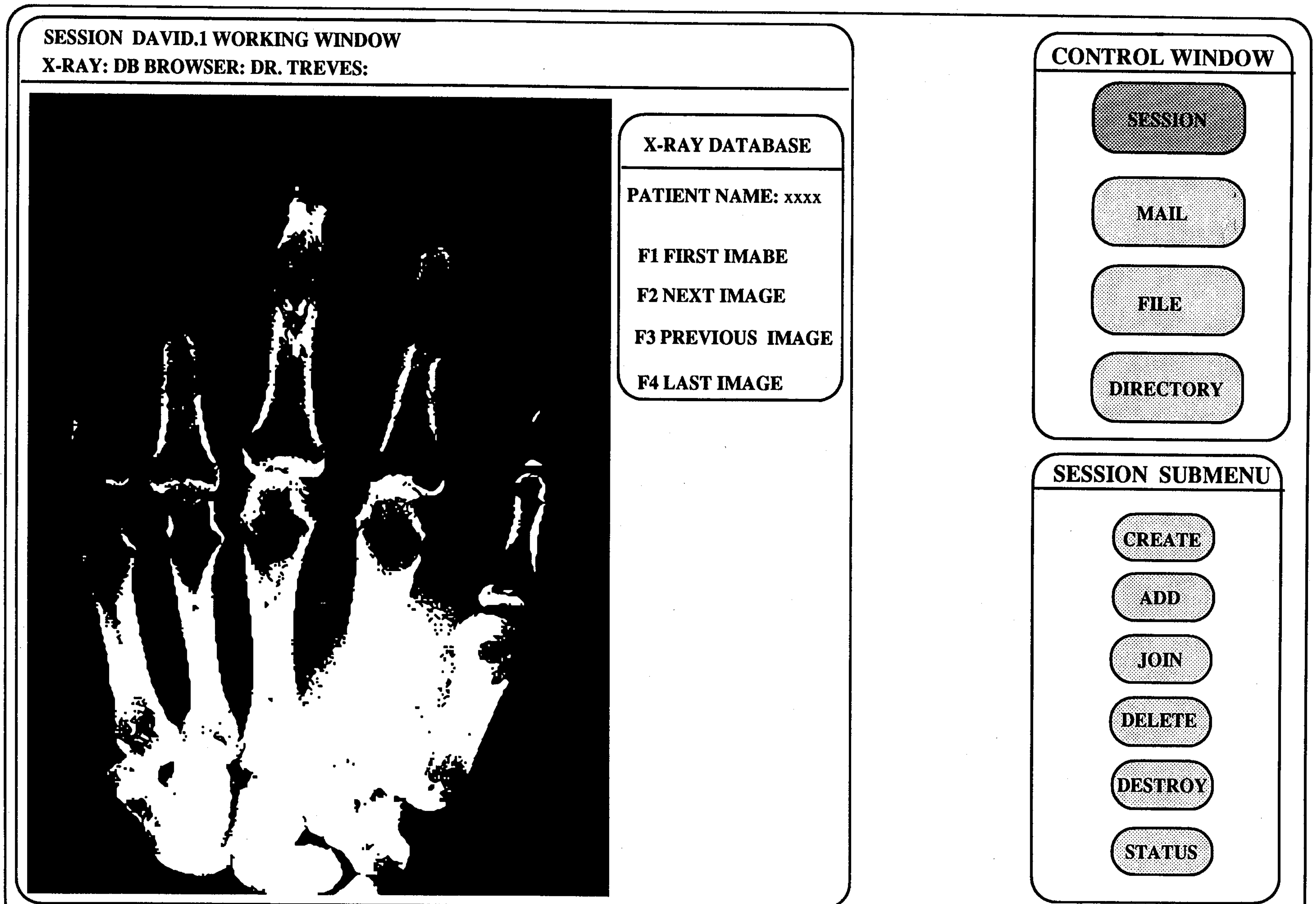
MBS SCREEN WHEN ADD DOCTORS IS CHOSEN



MBS SCREEN OF DR. TREVES WHEN DR. MARGULIES WANTS TO REQUEST TO JOIN SESSION DAVID.1

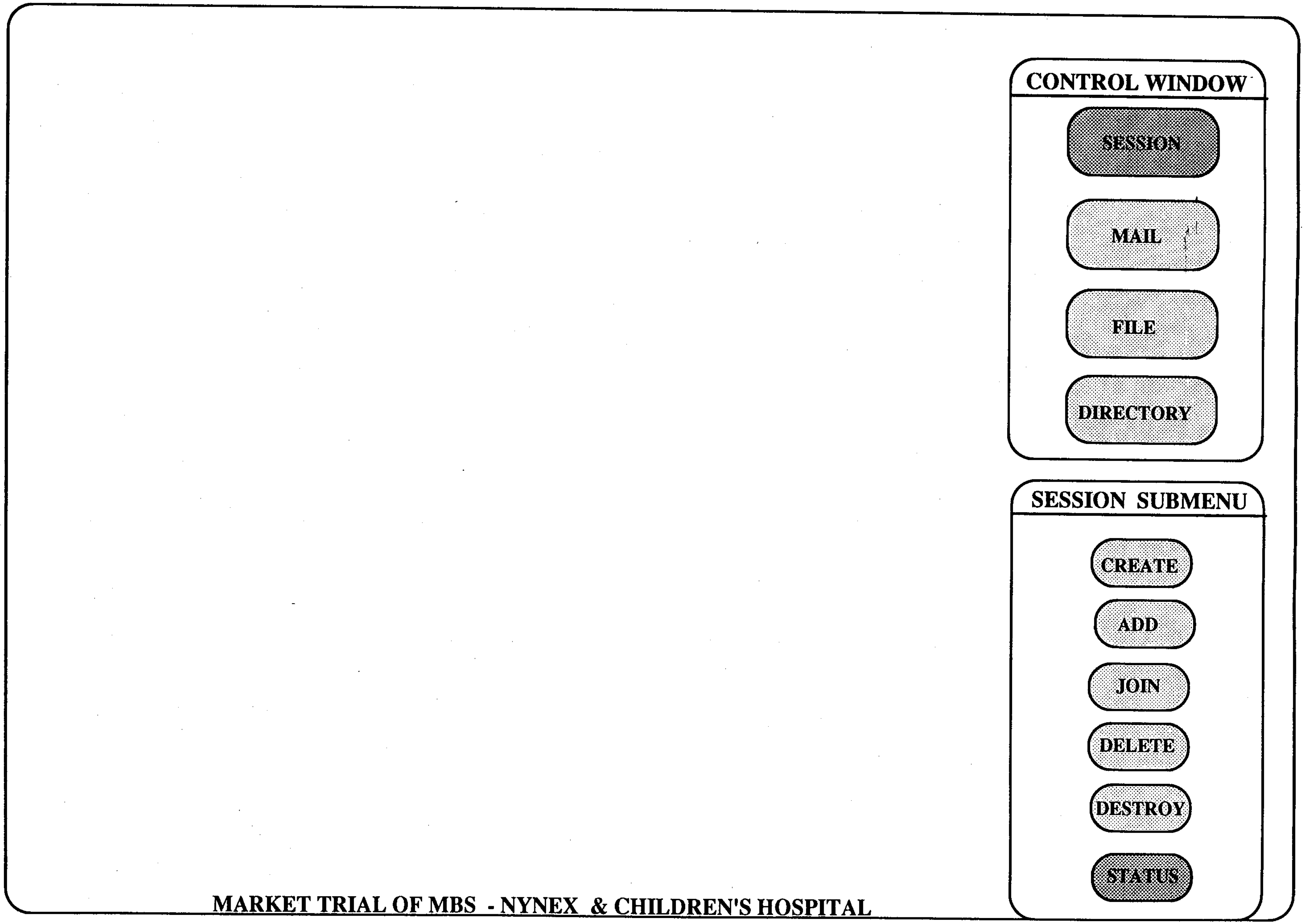


MBS SCREEN OF DR. TREVES WHEN HE ACCEPTS INCOMING SESSION REQUEST

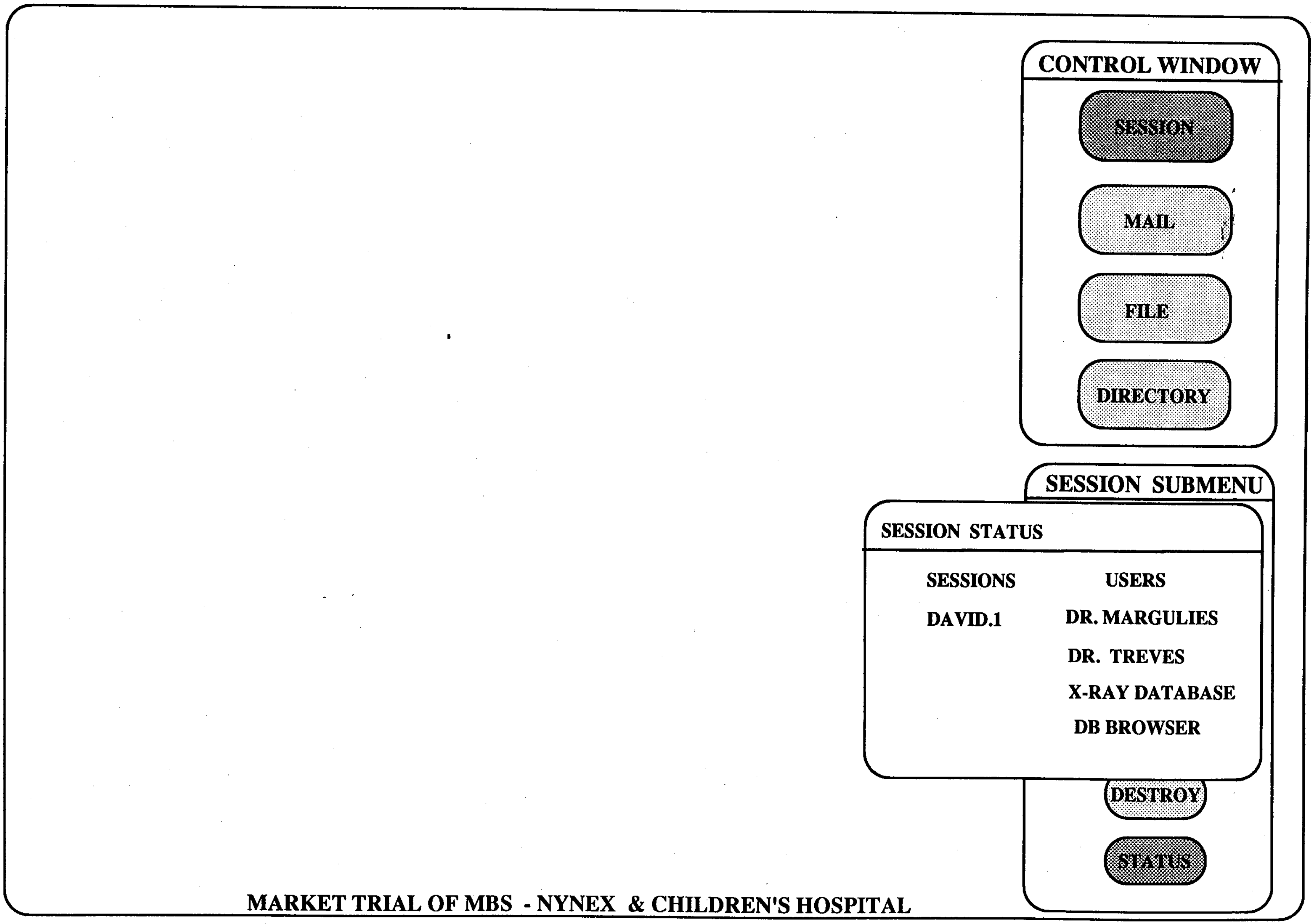


MARKET TRIAL OF MBS - NYNEX & CHILDREN'S HOSPITAL

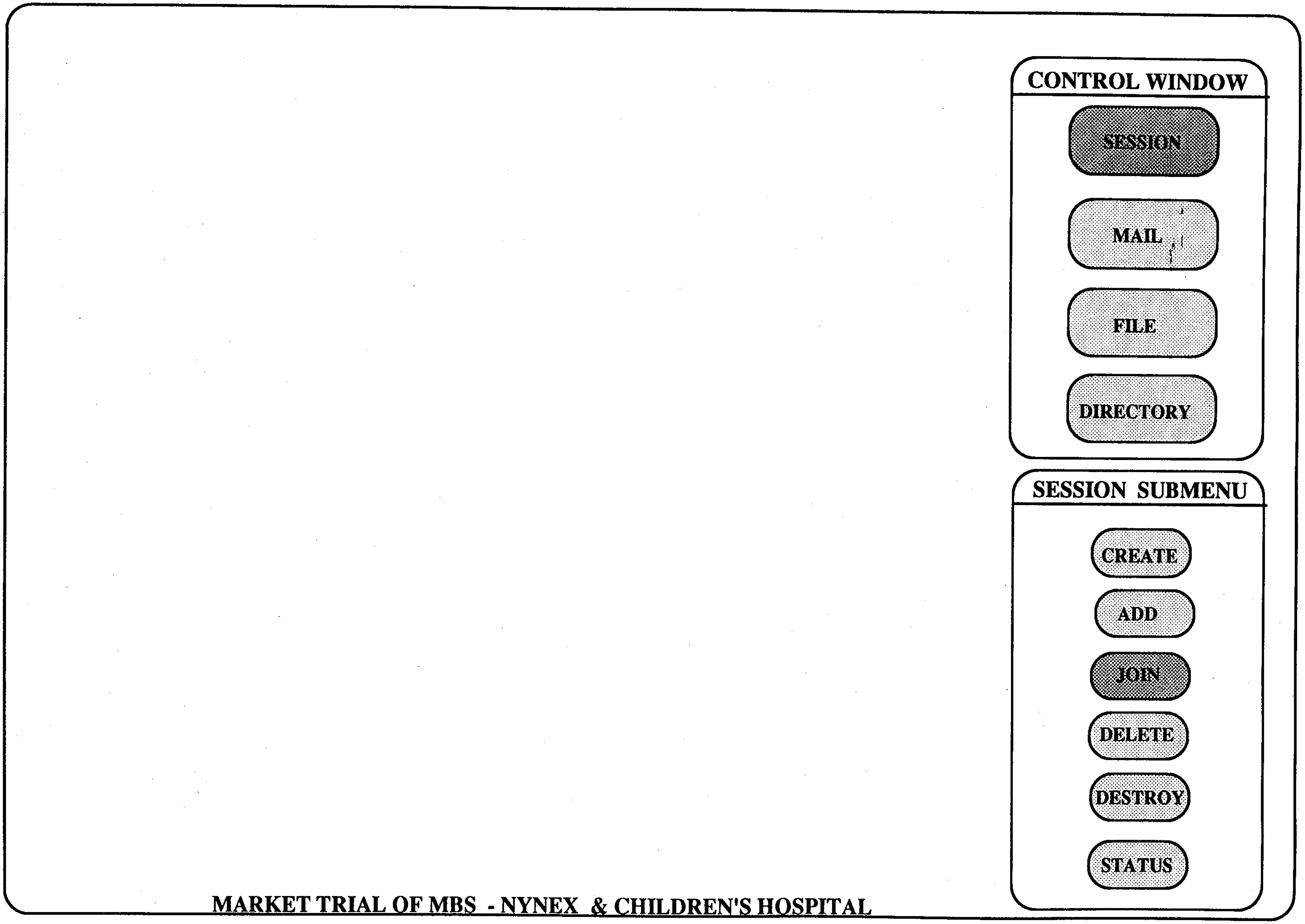
MBS SCREEN OF DR. JONES WHEN HE WANTS TO SEE THE STATUS OF THE ONGOING SESSIONS



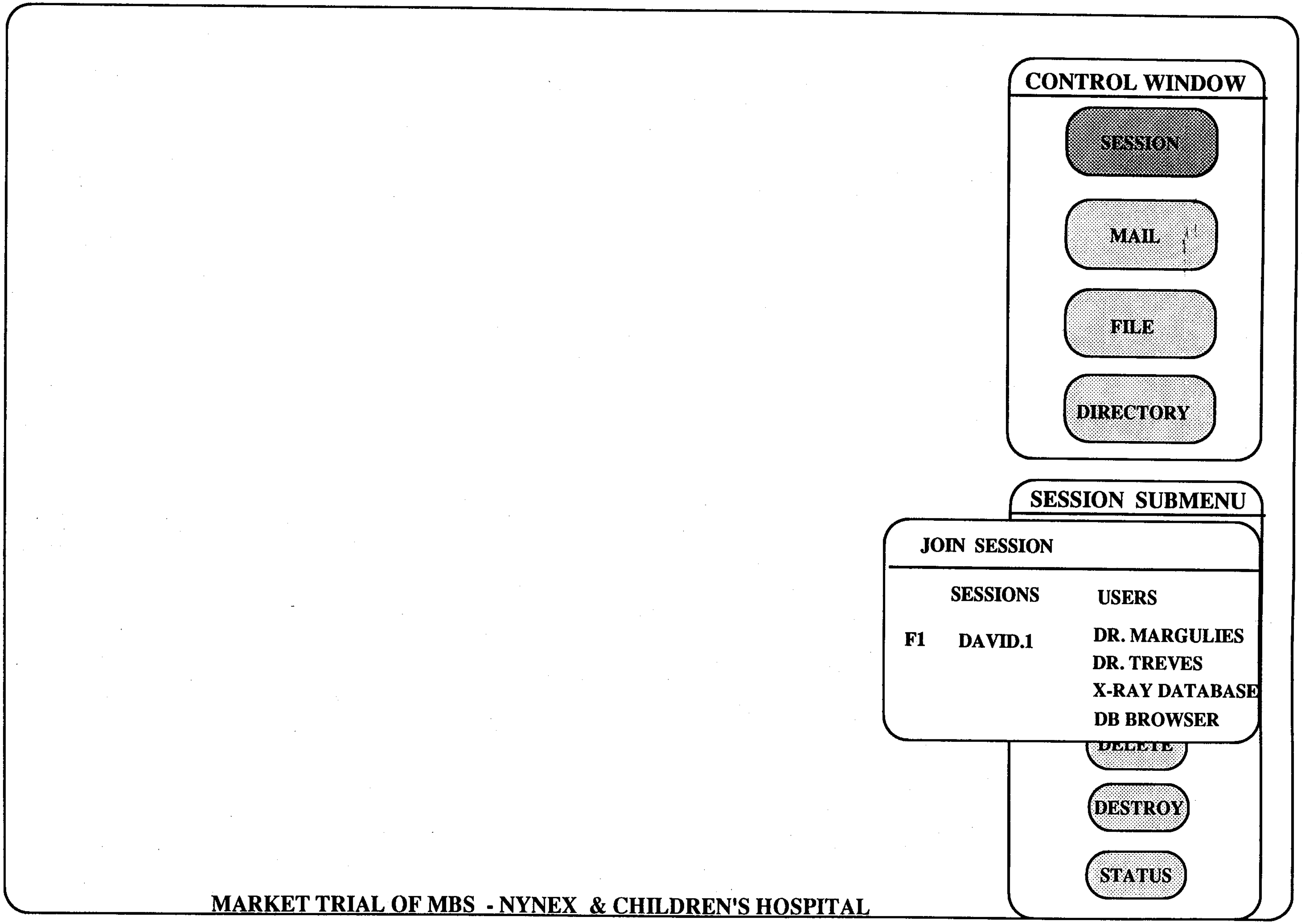
MBS SCREEN OF DR. JONES FOR VIEWING THE STATUS OF ONGOING SESSIONS



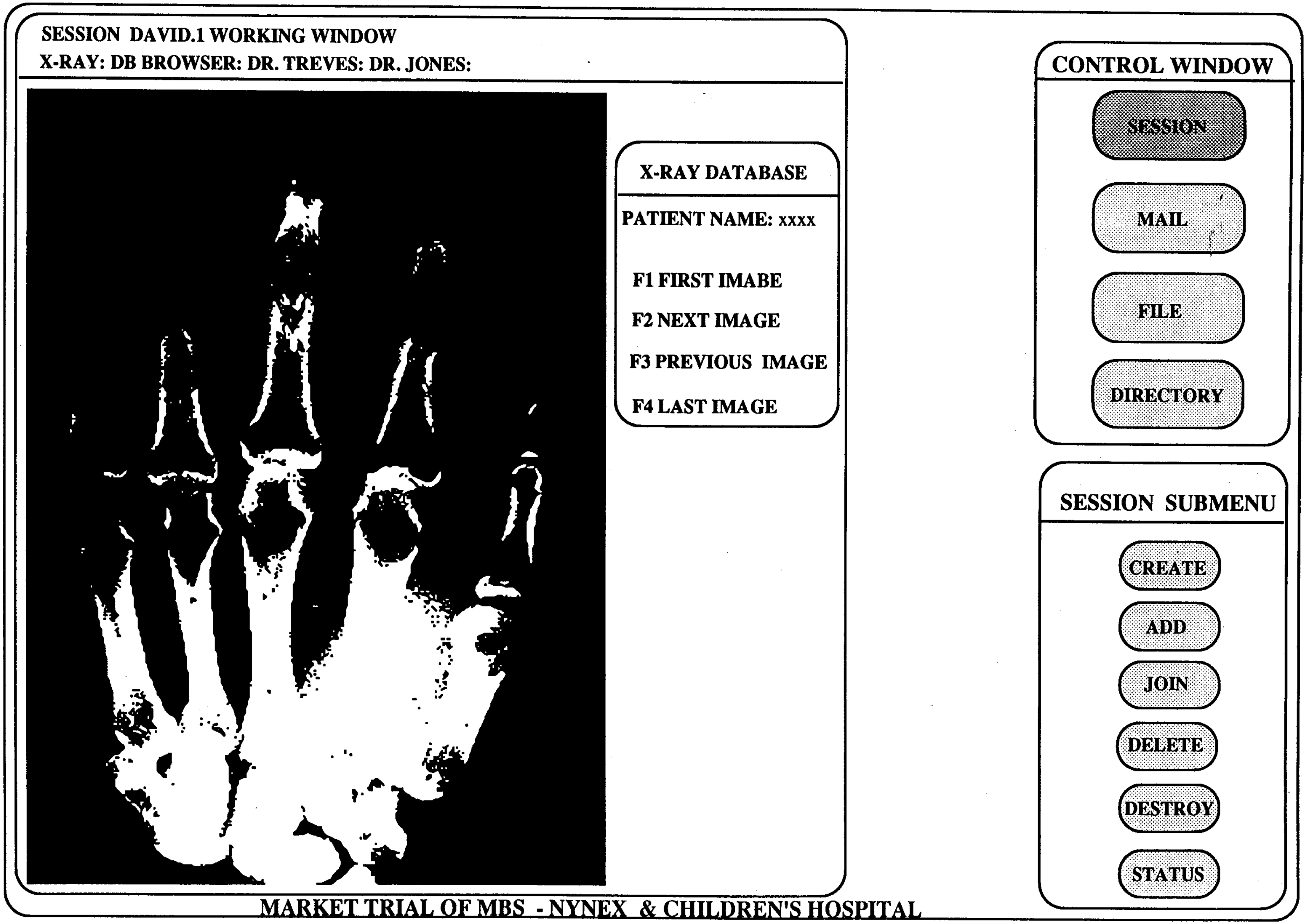
MBS SCREEN OF DR. JONES WHEN HE WANTS TO JOIN THE ONGOING SESSION DAVID.1



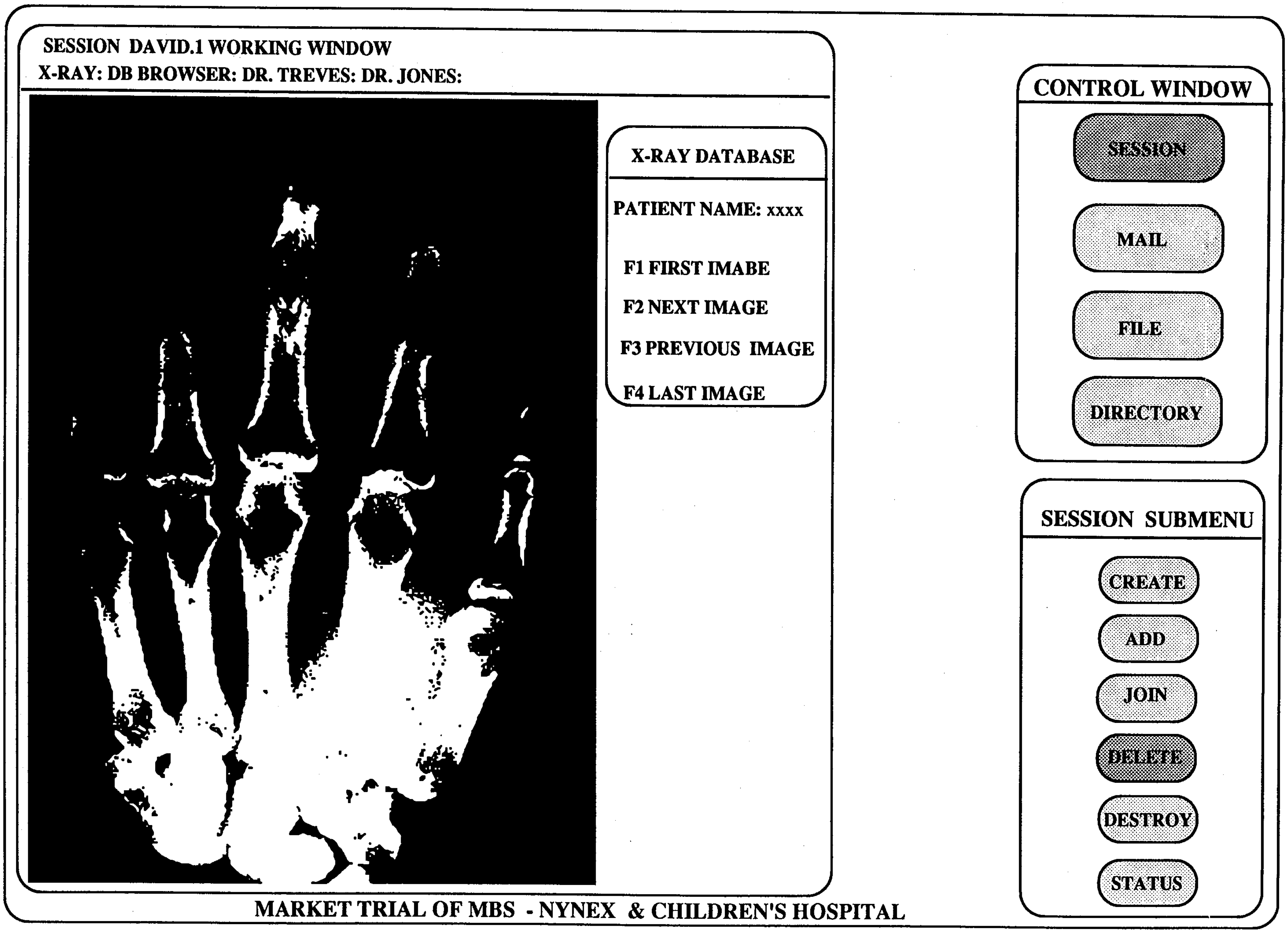
MBS SCREEN OF DR. JONES WHEN HE WANTS TO JOIN THE ONGOING SESSION DAVID.1



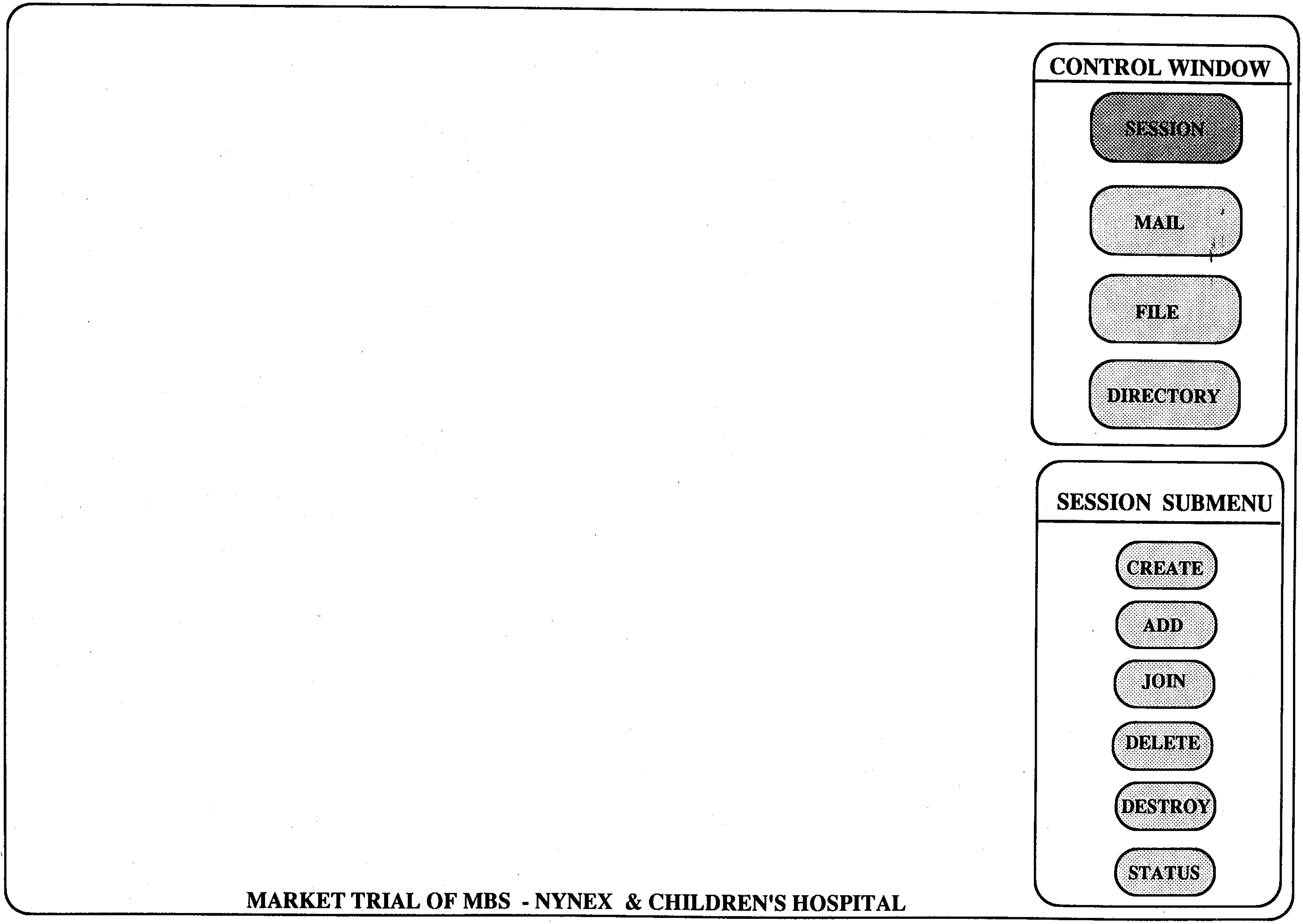
MBS SCREEN OF DR. JONES WHEN HE HAS JOINED THE ONGOING SESSION DAVID.1



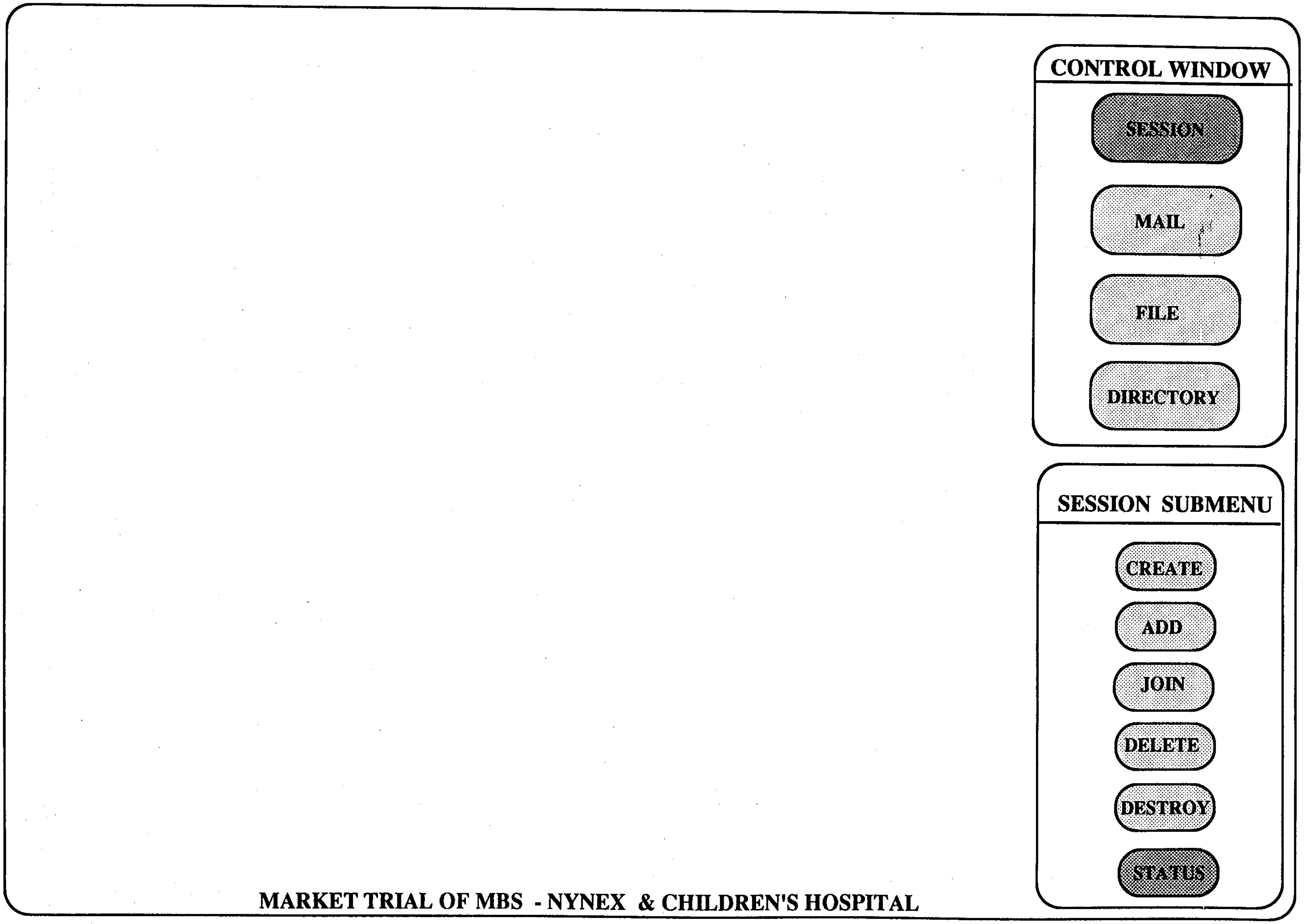
MBS SCREEN OF DR. JONES WHEN HE WANTS TO BE DELETED FROM THE ONGOING SESSION DAVID.1 S21



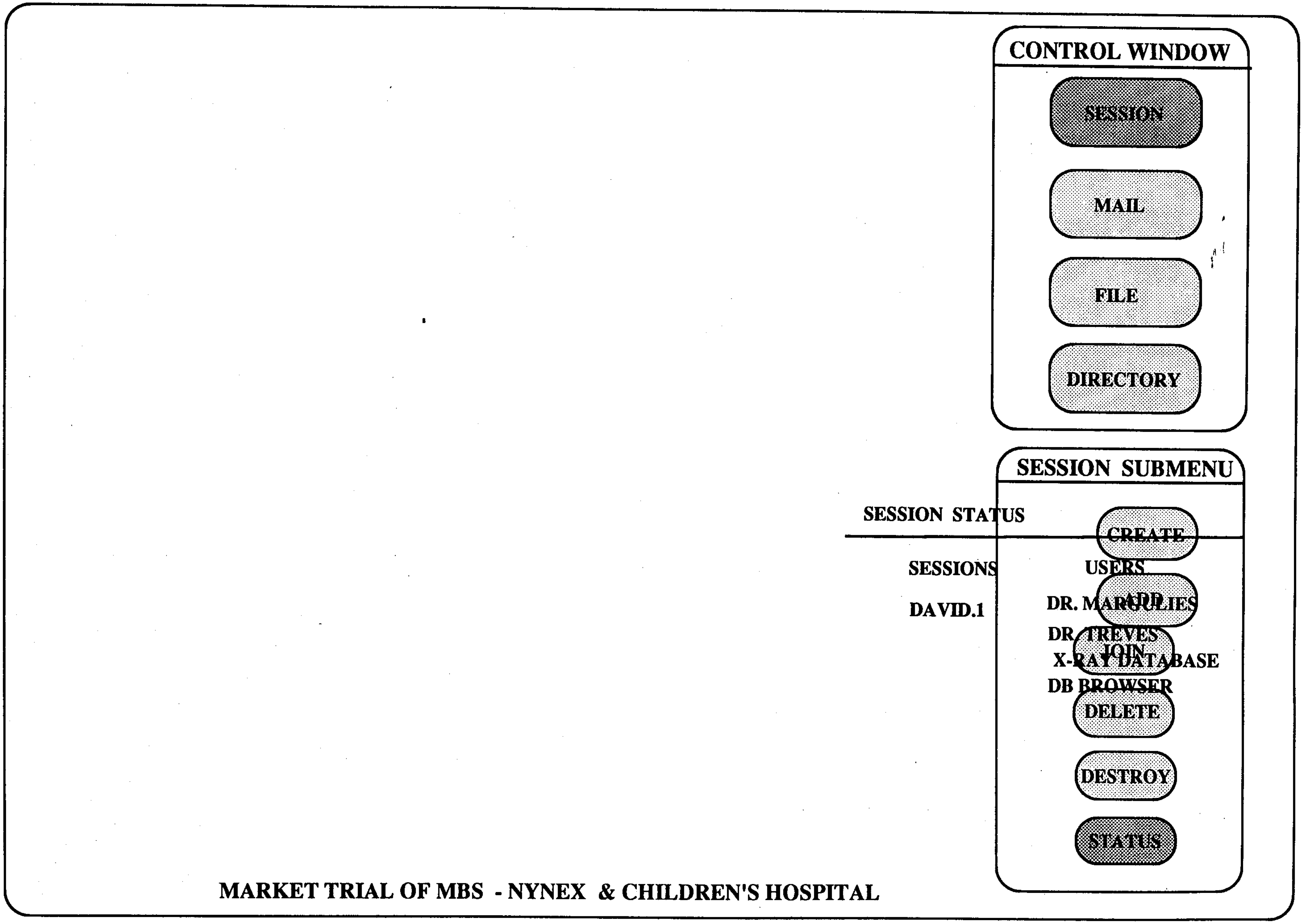
MBS SCREEN OF DR. JONES AFTER HE HAS DELETED THE ONGOING SESSION DAVID.1



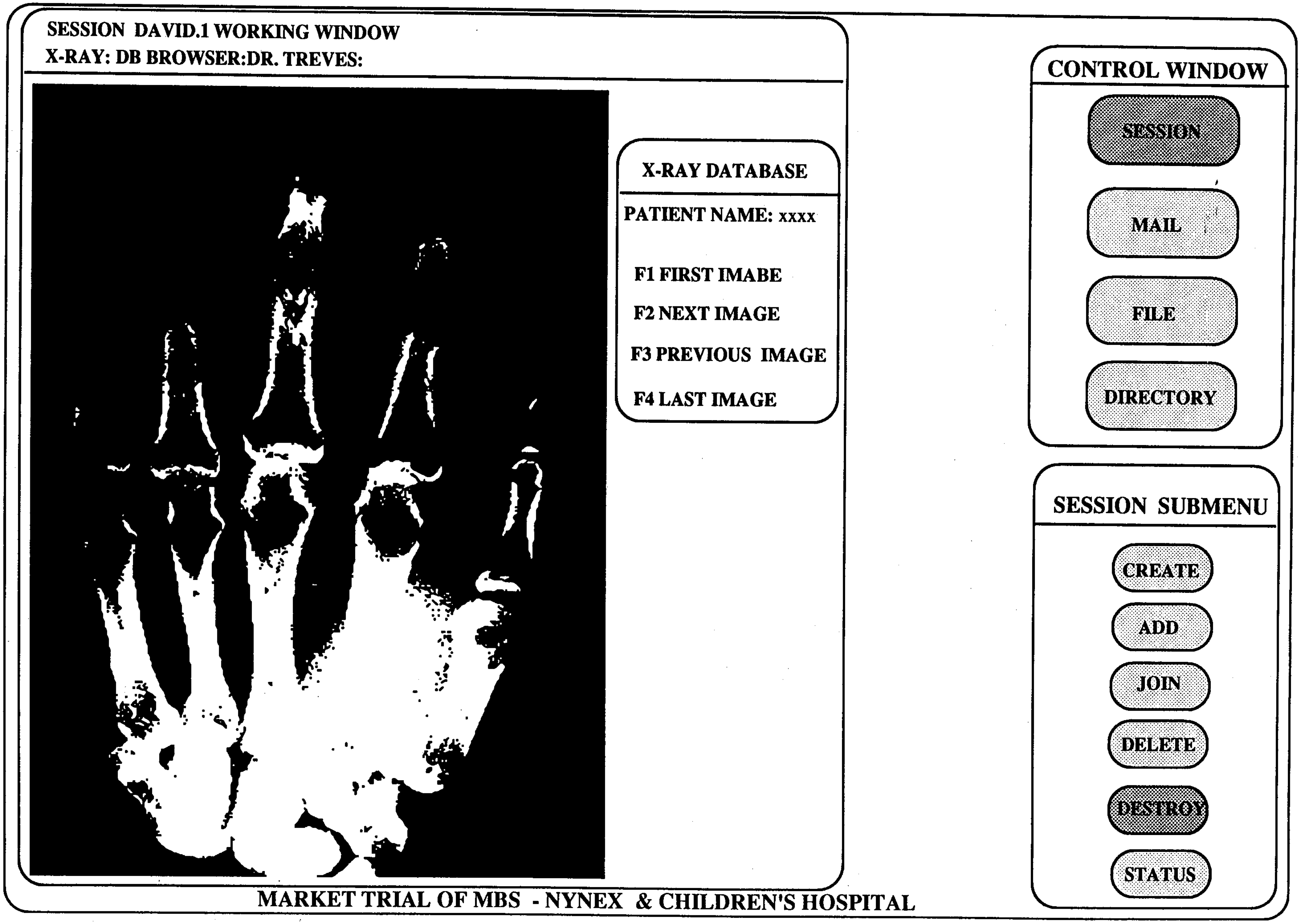
MBS SCREEN OF DR. JONES WHEN HE WANTS TO CONFIRM THAT HE IS DELETED FROM THE SESSION DAVID.1 S23



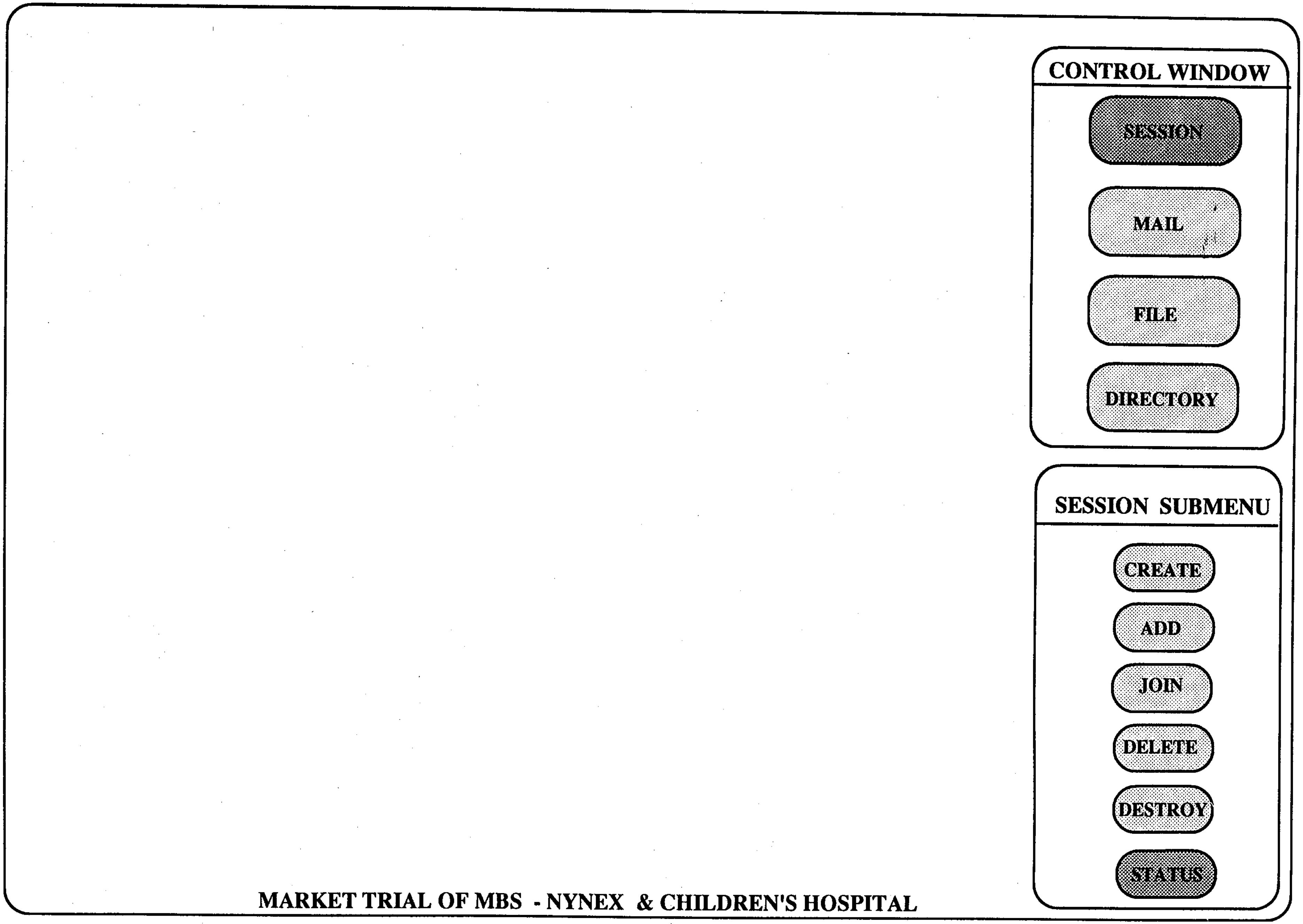
MBS SCREEN OF DR. JONES SHOWS THE CURRENT SCREEN STATUS



MBS SCREEN OF DR. MARGULIES WHEN HE WANTS TO DESTROY THE ONGOING SESSION DAVID.1



MBS SCREEN OF DR. MARGULIES AFTER HE CHOSEN DESTROY



MBS SCREEN OF DR. MARGULIES WHEN HE HAS CHOSEN STATUS

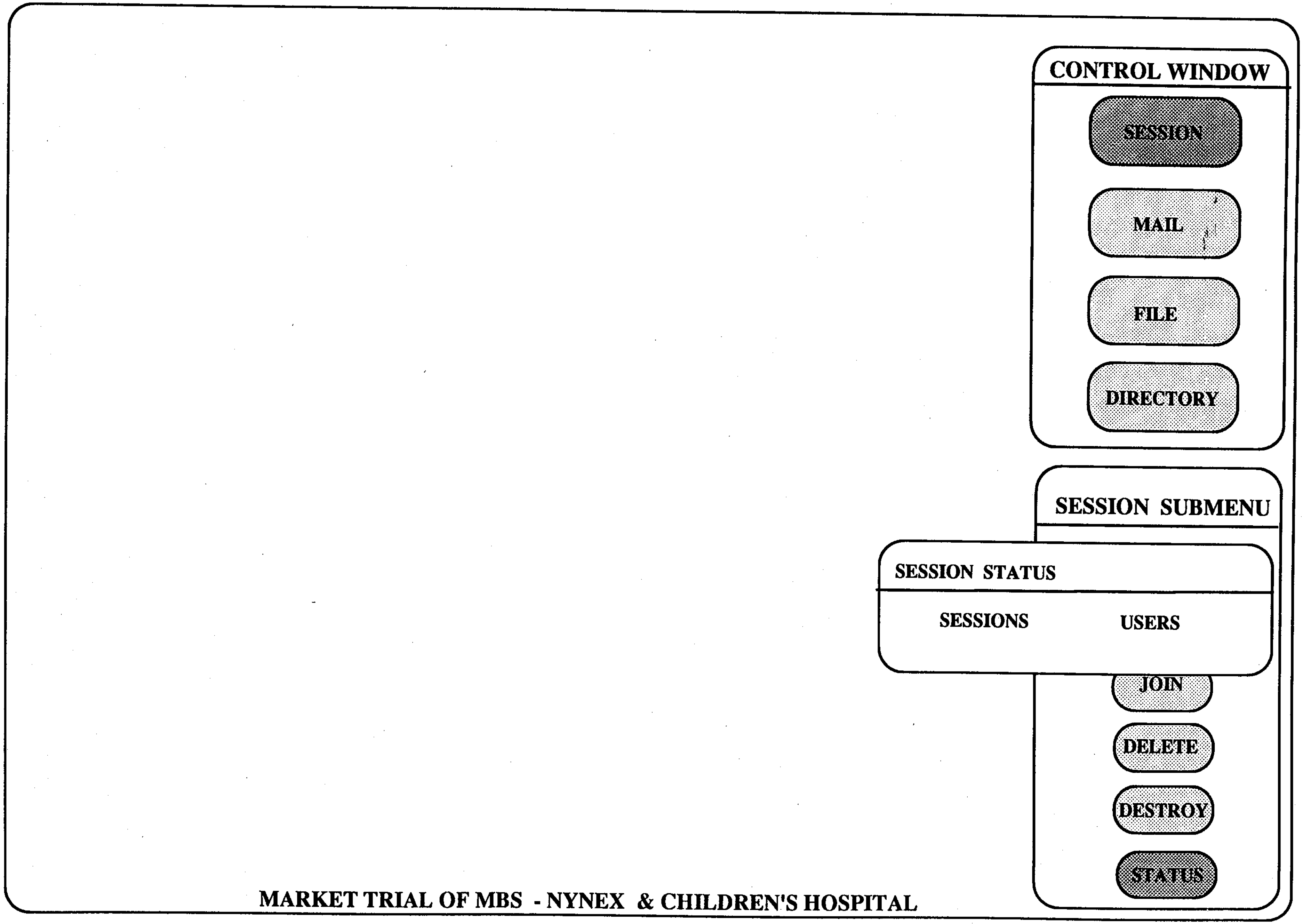


Figure 3.x depicts the state diagram for the session service. We show how the sessions can be generated by following the presentation screen layout formats.

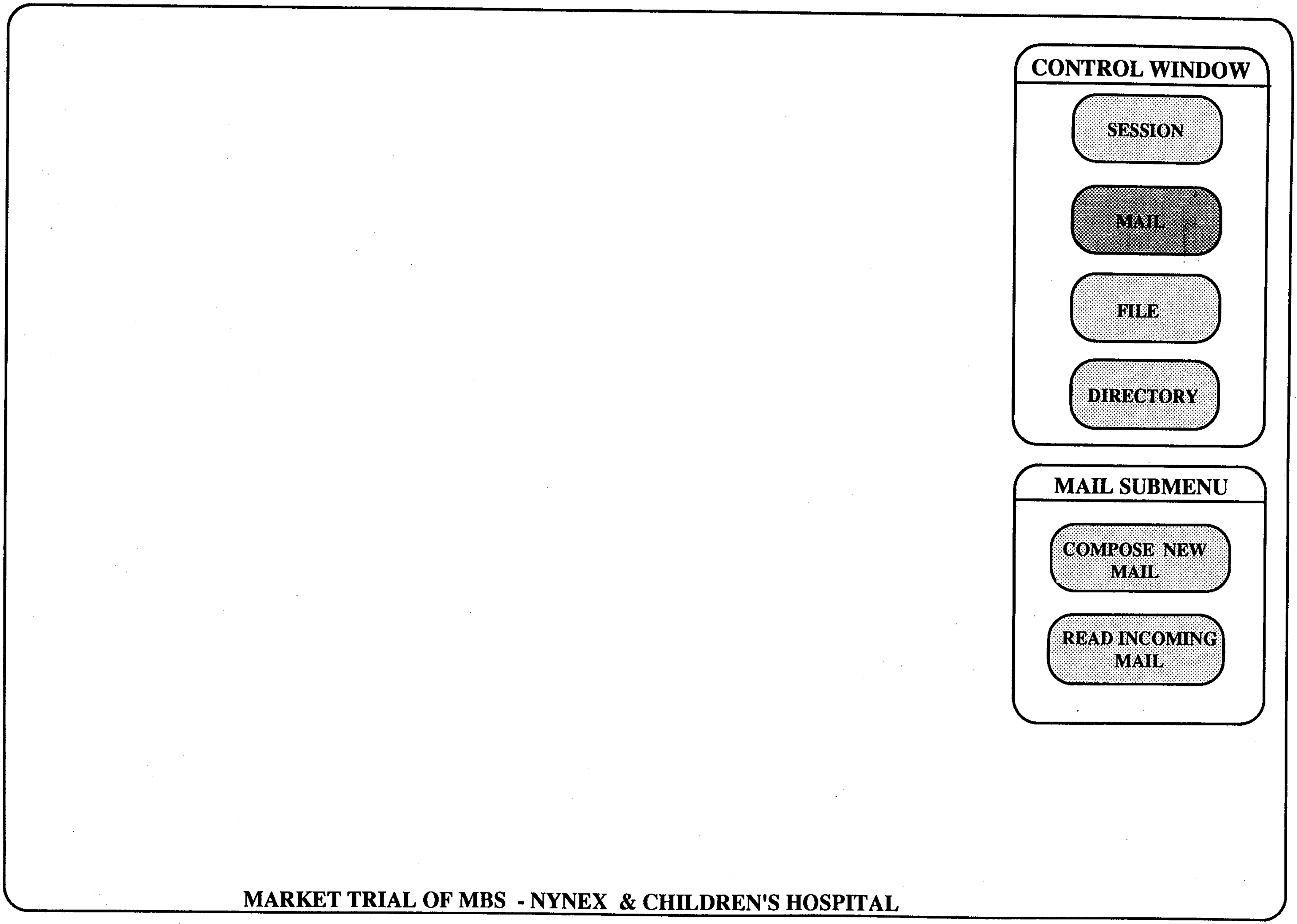
Figure 3.x Session State Model

3.2.3.2 Mail

In a similar fashion to the session implementations, we can extend this to the service of mail. This can be viewed as a similar fashion. Figure 3.x depicts all of the presentation formats on the mail service.

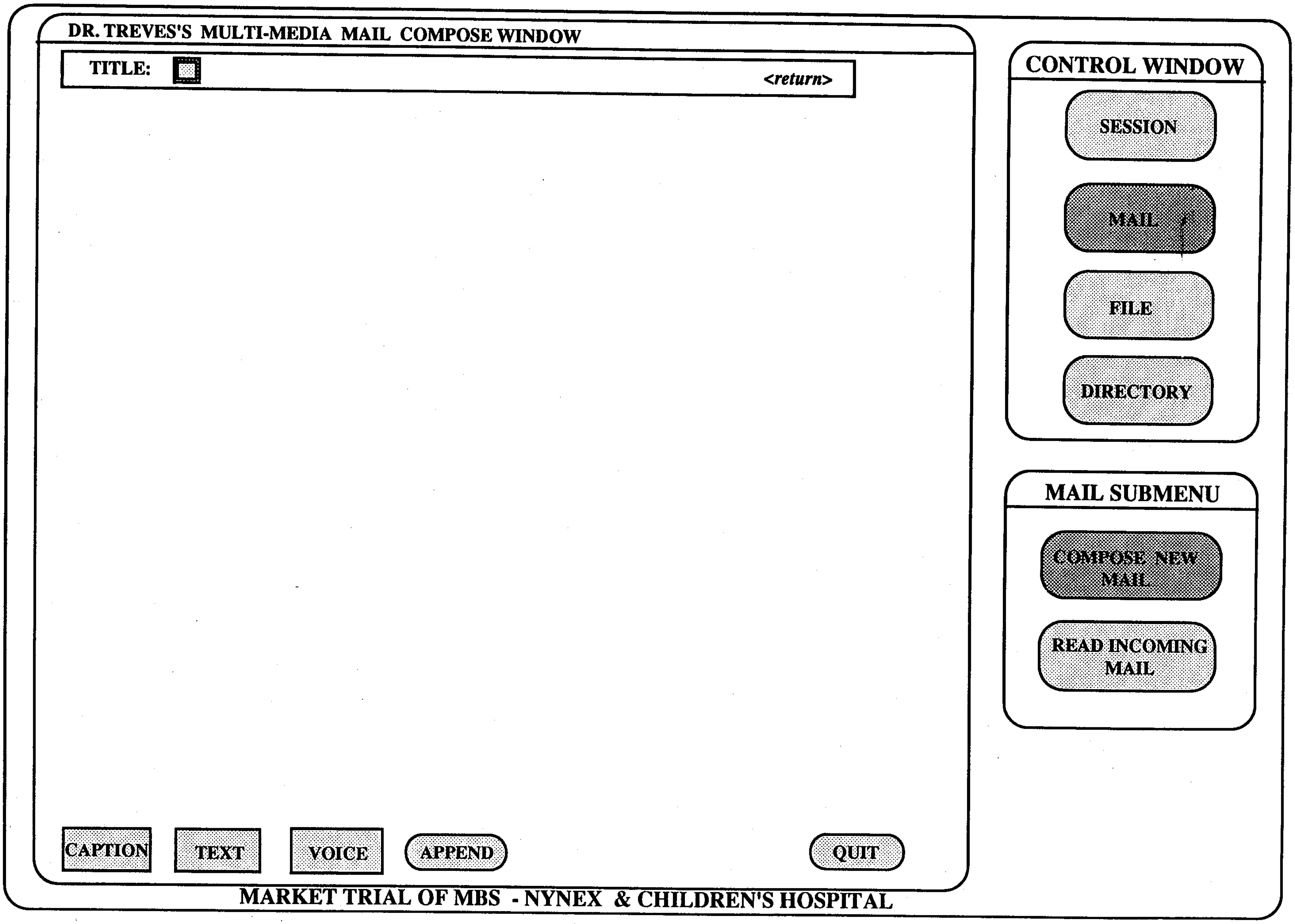
Figure 3.x Mail Screen Layout

MBS INITIAL SCREEN WHEN MAIL IS SELECTED

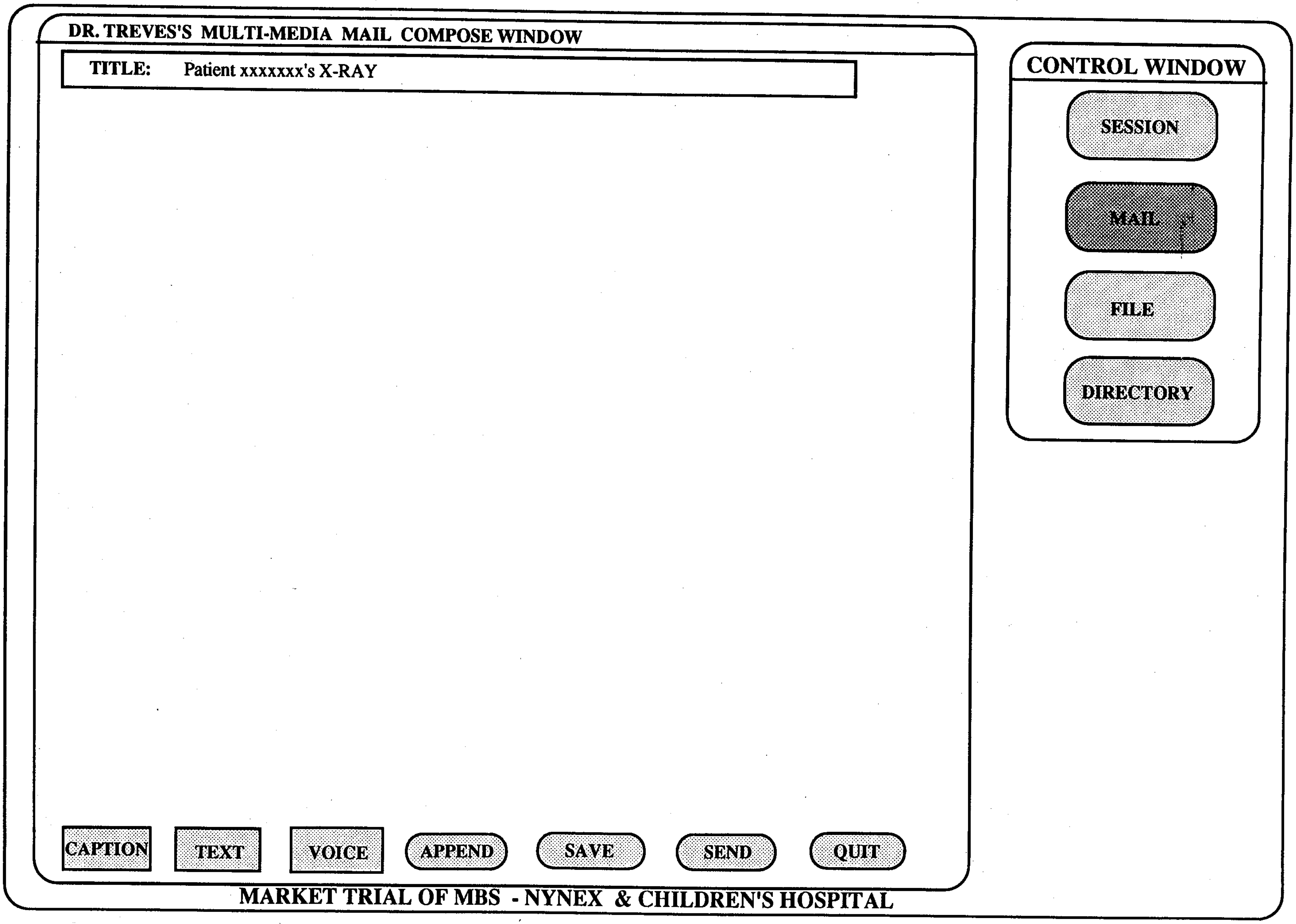


MARKET TRIAL OF MBS - NYNEX & CHILDREN'S HOSPITAL

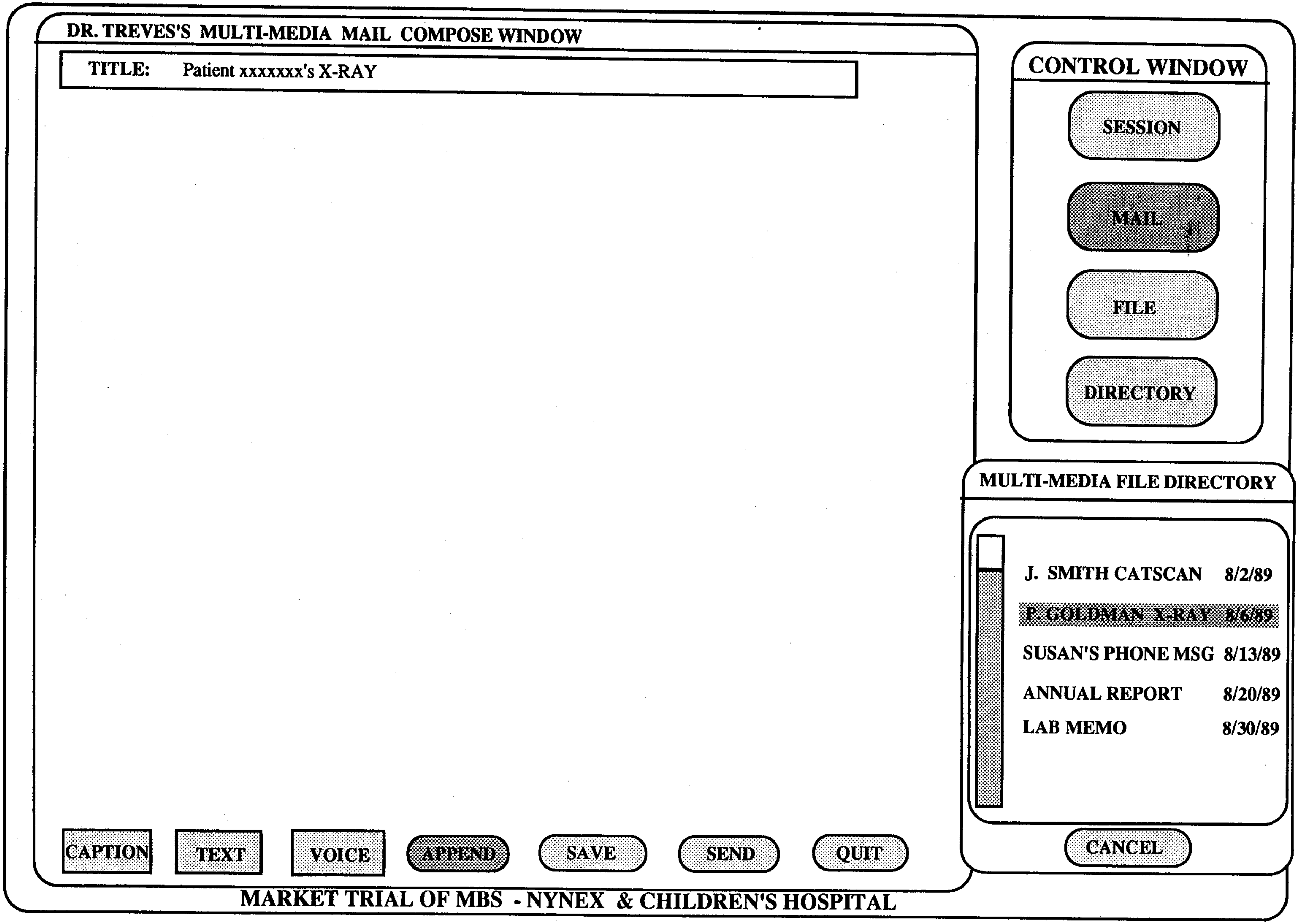
MBS SCREEN WHEN DR. TREVES WANTS TO COMPOSE A NEW MAIL MESSAGE



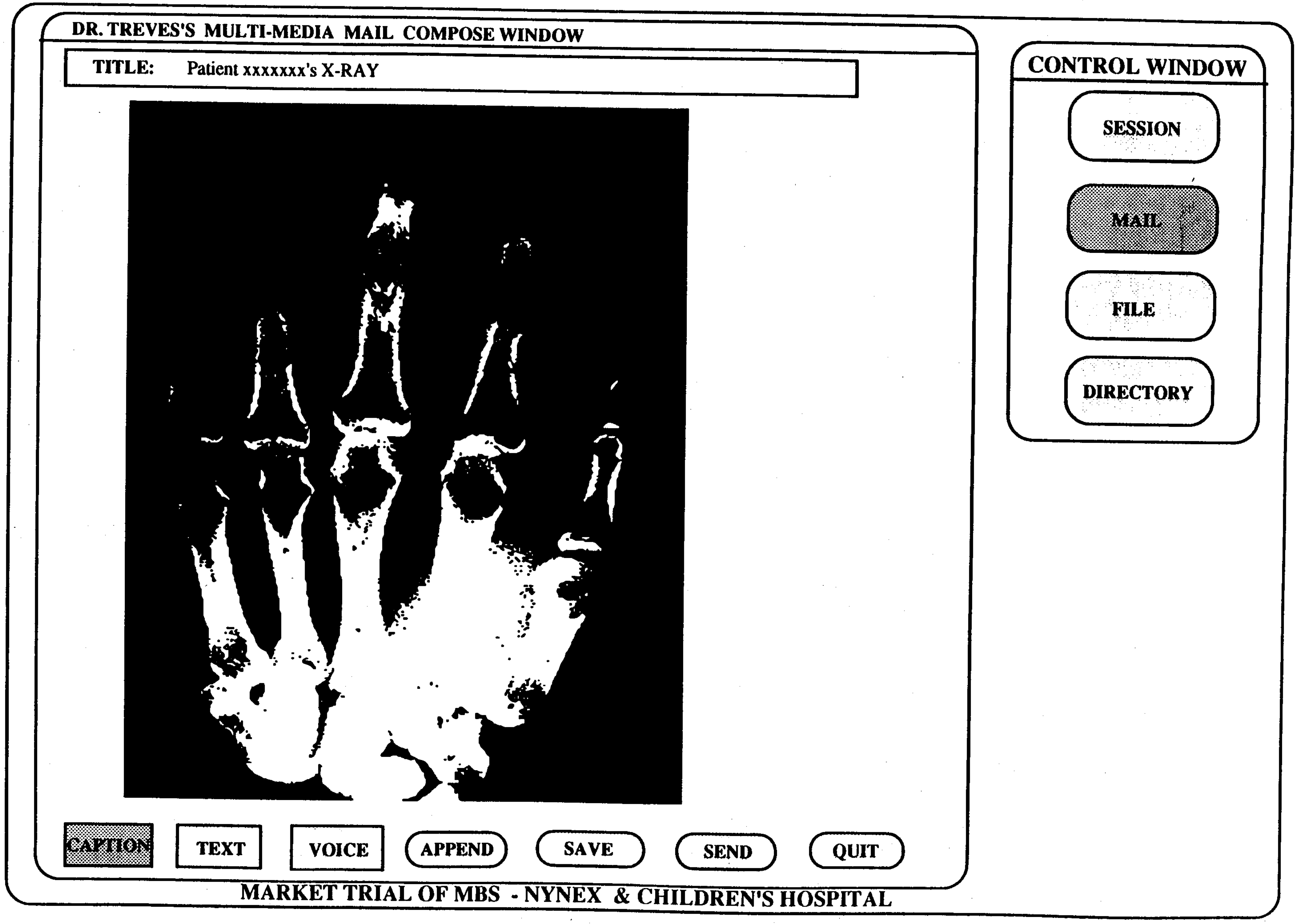
MBS SCREEN WHEN DR. TREVES WANTS TO START COMPOSING THE CONTENTS OF THE MAIL MESSAGE



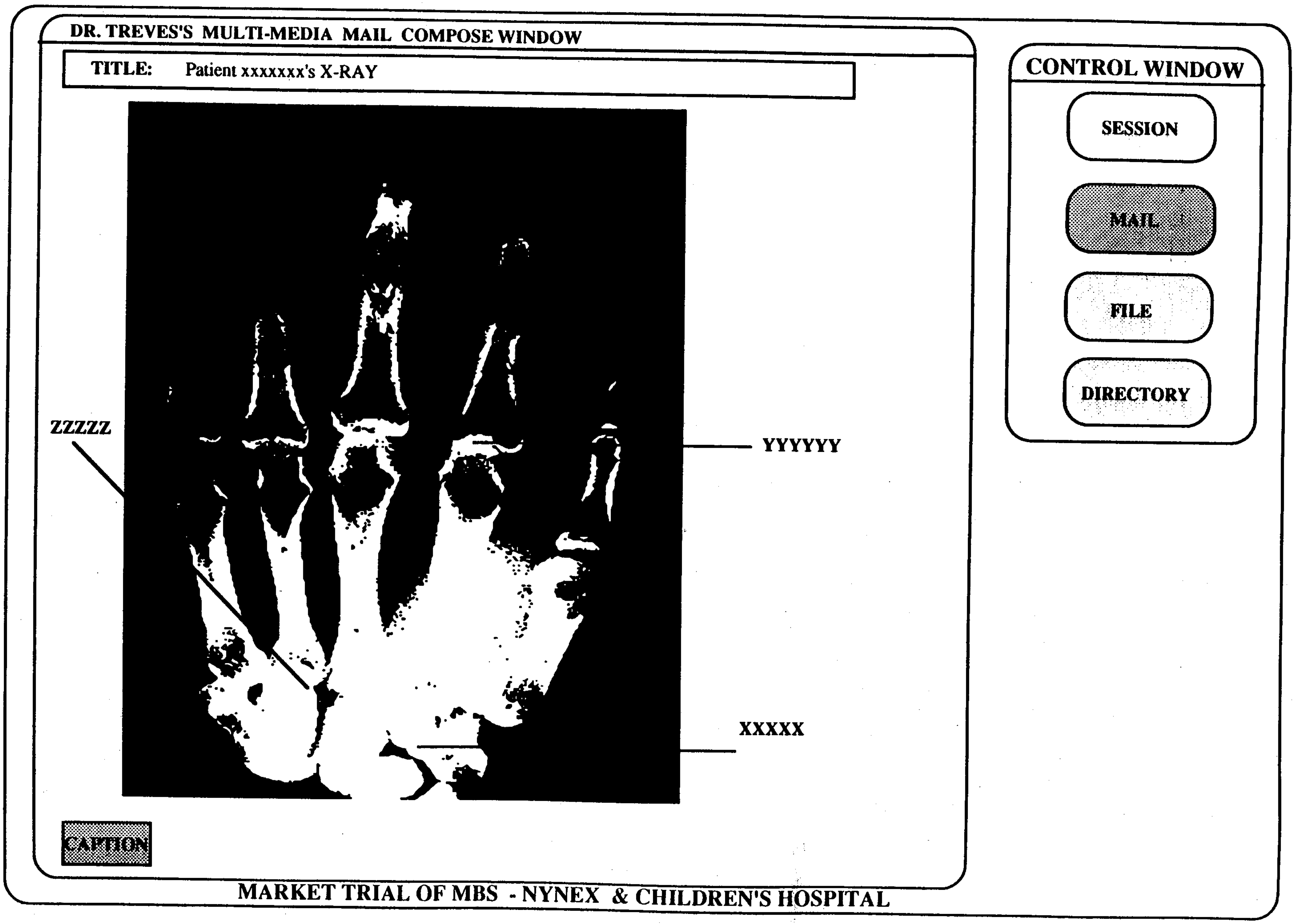
MBS SCREEN WHEN DR. TREVES WANTS TO APPEND THE CONTENTS OF SOME FILE TO THE NEW MESSAGE M4



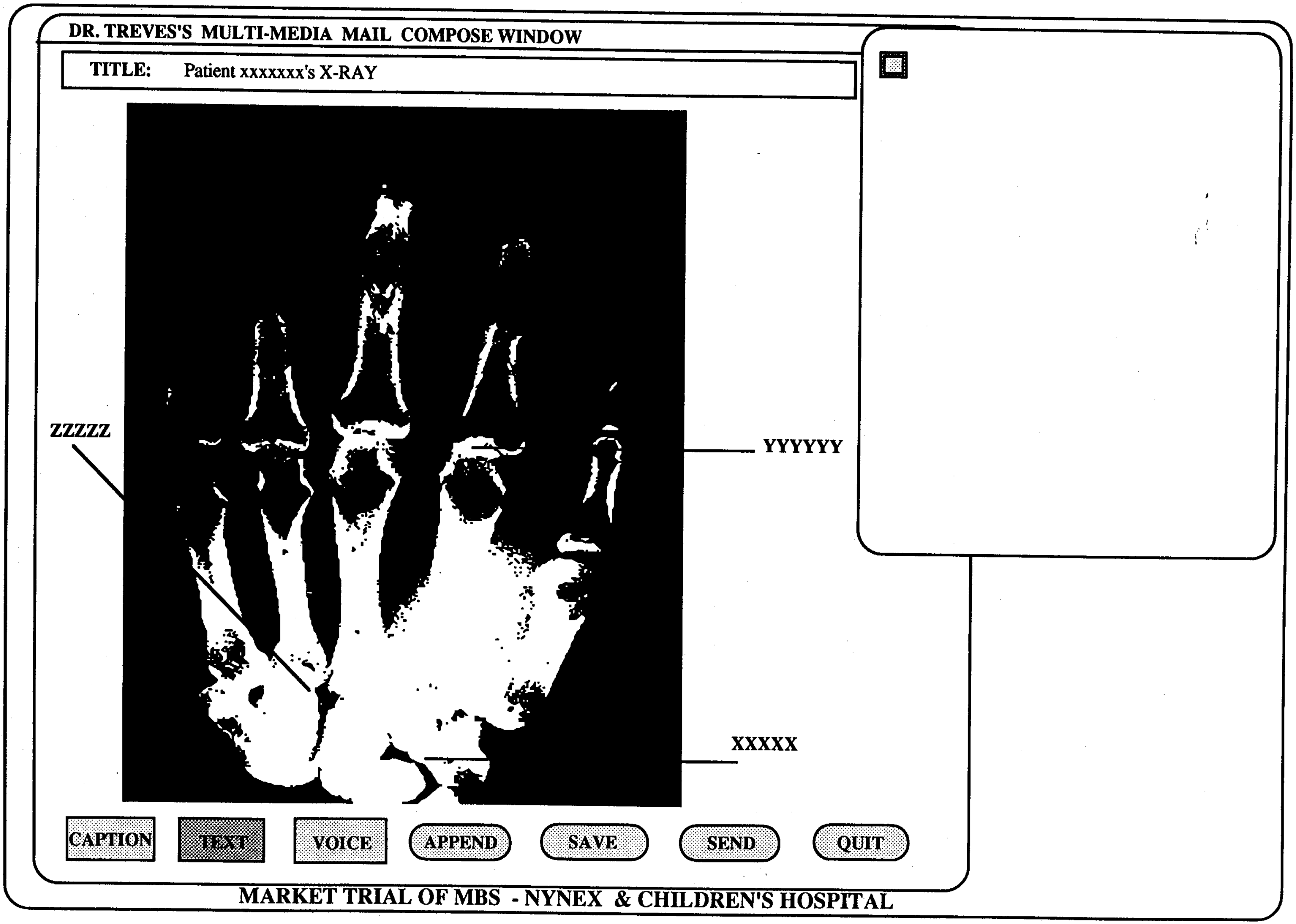
MBS SCREEN WHEN THE SELECTED FILE HAS BEEN APPENDED AND DR. TREVES WANTS TO ANNOTATE SOME CAPTIONS TO THE IMAGE



MBS SCREEN AFTER DR. TREVES HAS ANNOTATED THE DESIRED CAPTIONS TO THE IMAGE



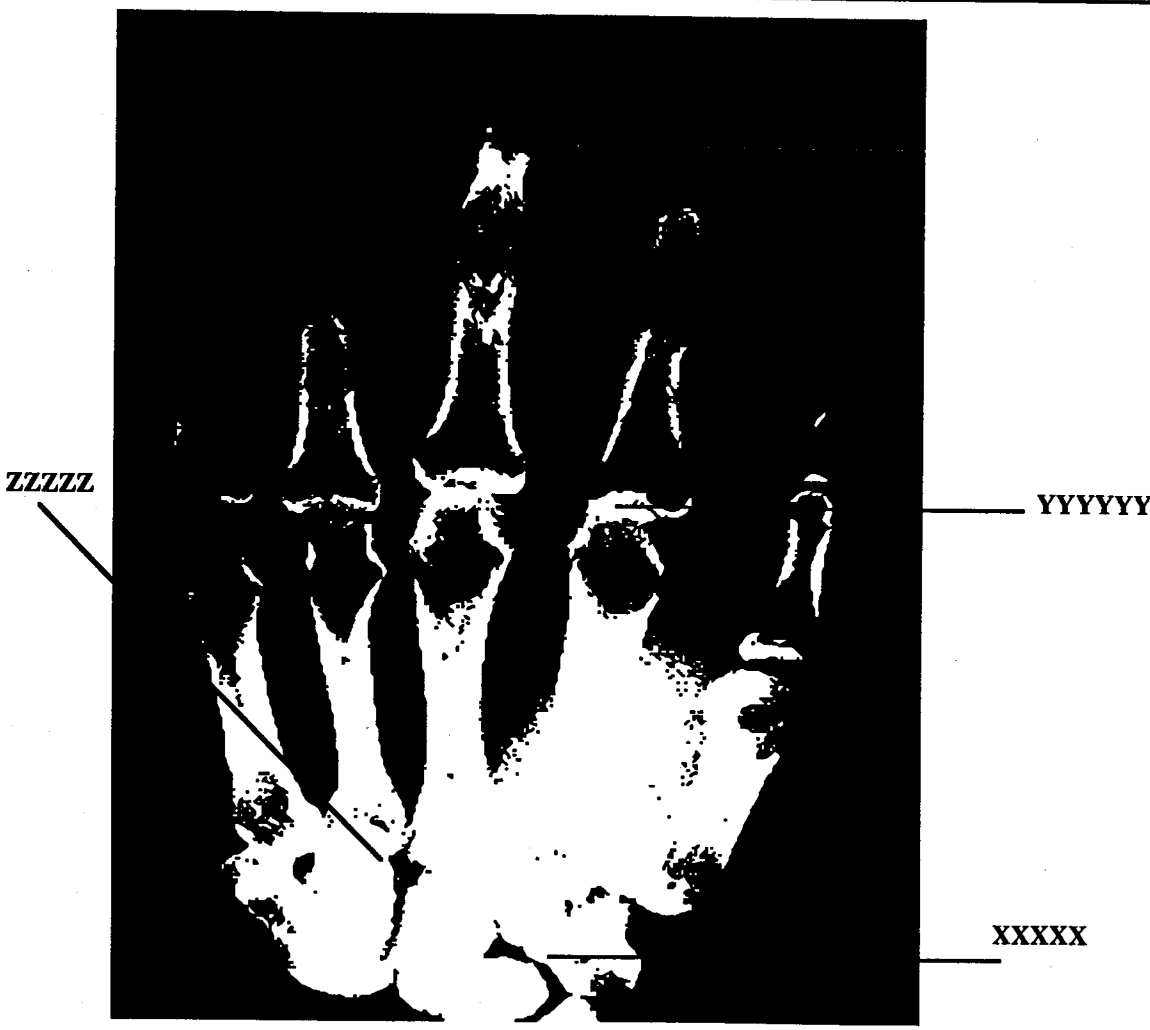
MBS SCREEN WHEN DR. TREVES WANTS TO INCLUDE SOME TEXT IN THE MESSAGE



MBS SCREEN WHEN DR. TREVES WANTS TO ANNOTATE A VOICE MESSAGE TO HIS MAIL MESSAGE

DR. TREVES'S MULTI-MEDIA MAIL COMPOSE WINDOW

SUBJECT: Patient xxxxxxx's X-RAY



LAB REPORT

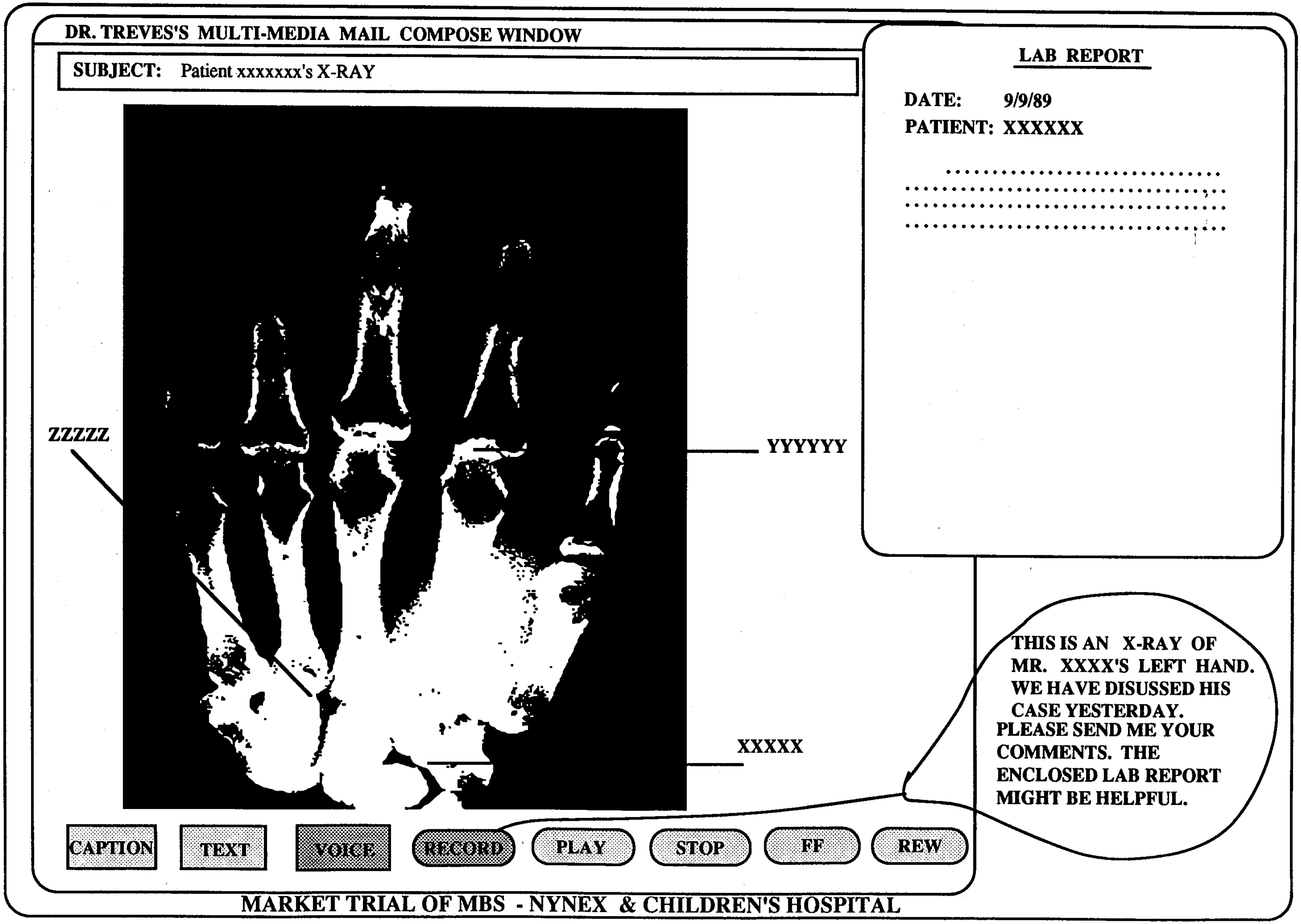
DATE: 9/9/89
PATIENT: XXXXXX

....
....
....
....

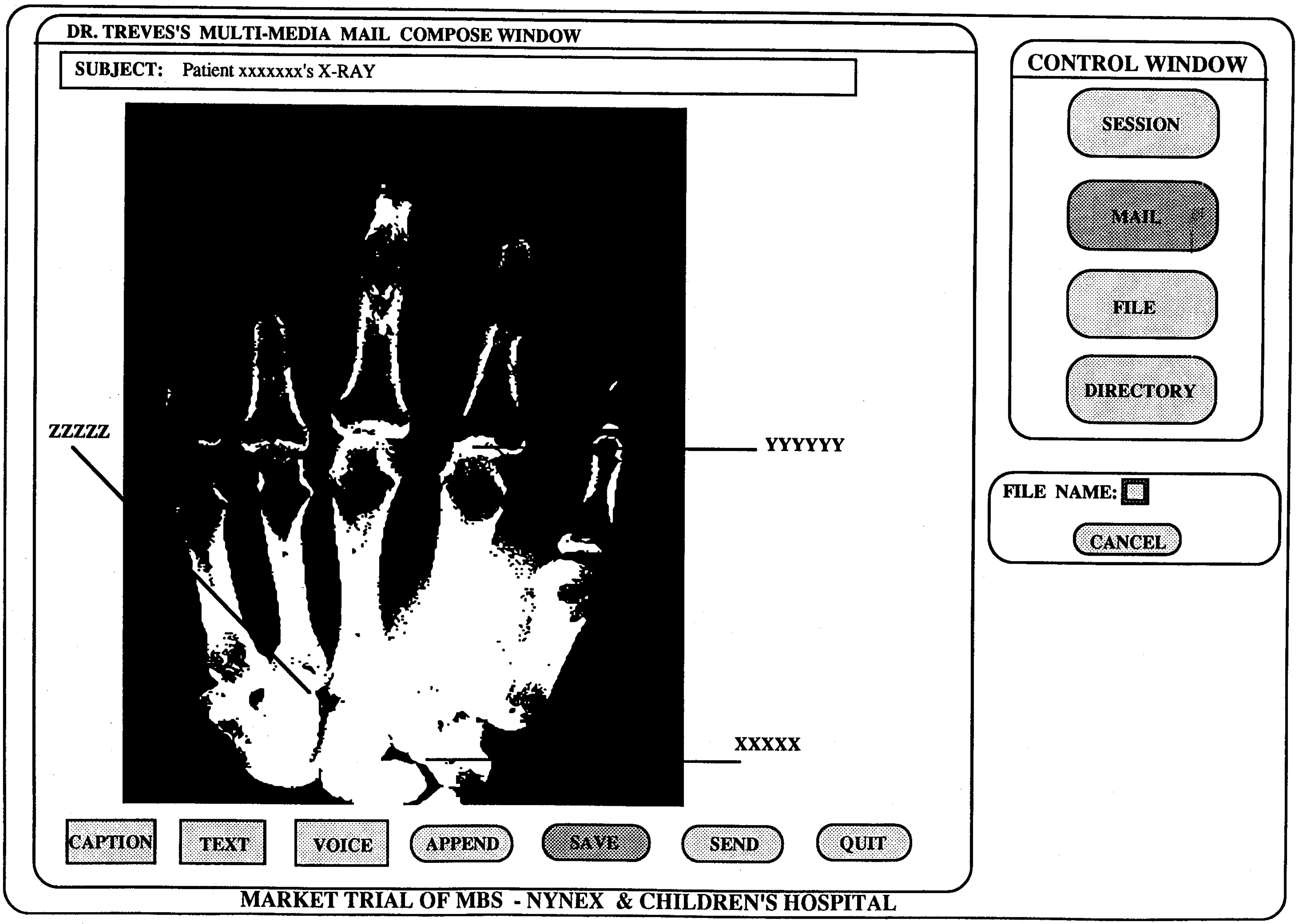
CAPTION | TEXT | VOICE | RECORD | PLAY | STOP | FF | REW

MARKET TRIAL OF MBS - NYNEX & CHILDREN'S HOSPITAL

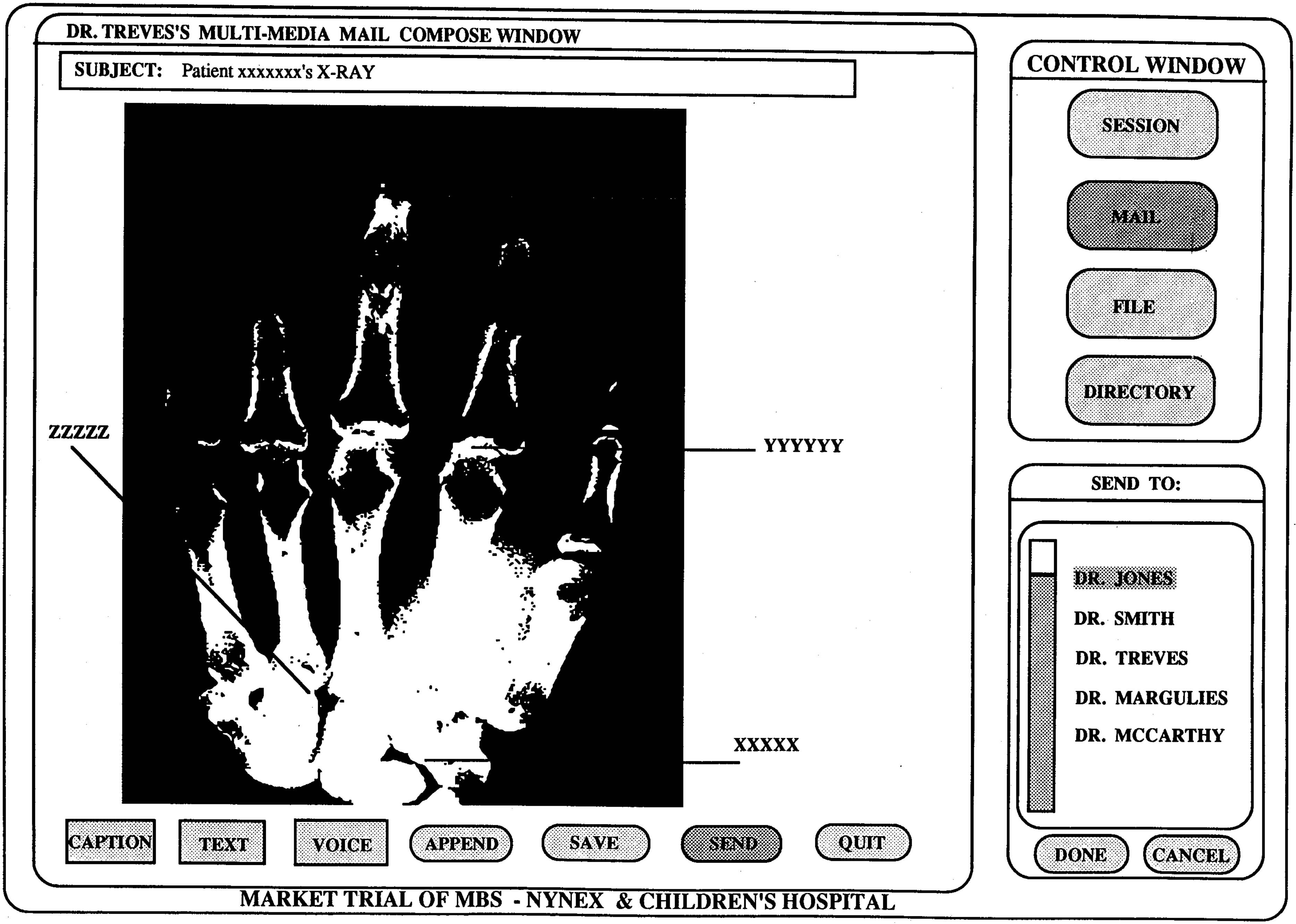
MBS SCREEN AFTER DR. TREVES HAS RECORDED HIS VOICE MESSAGE



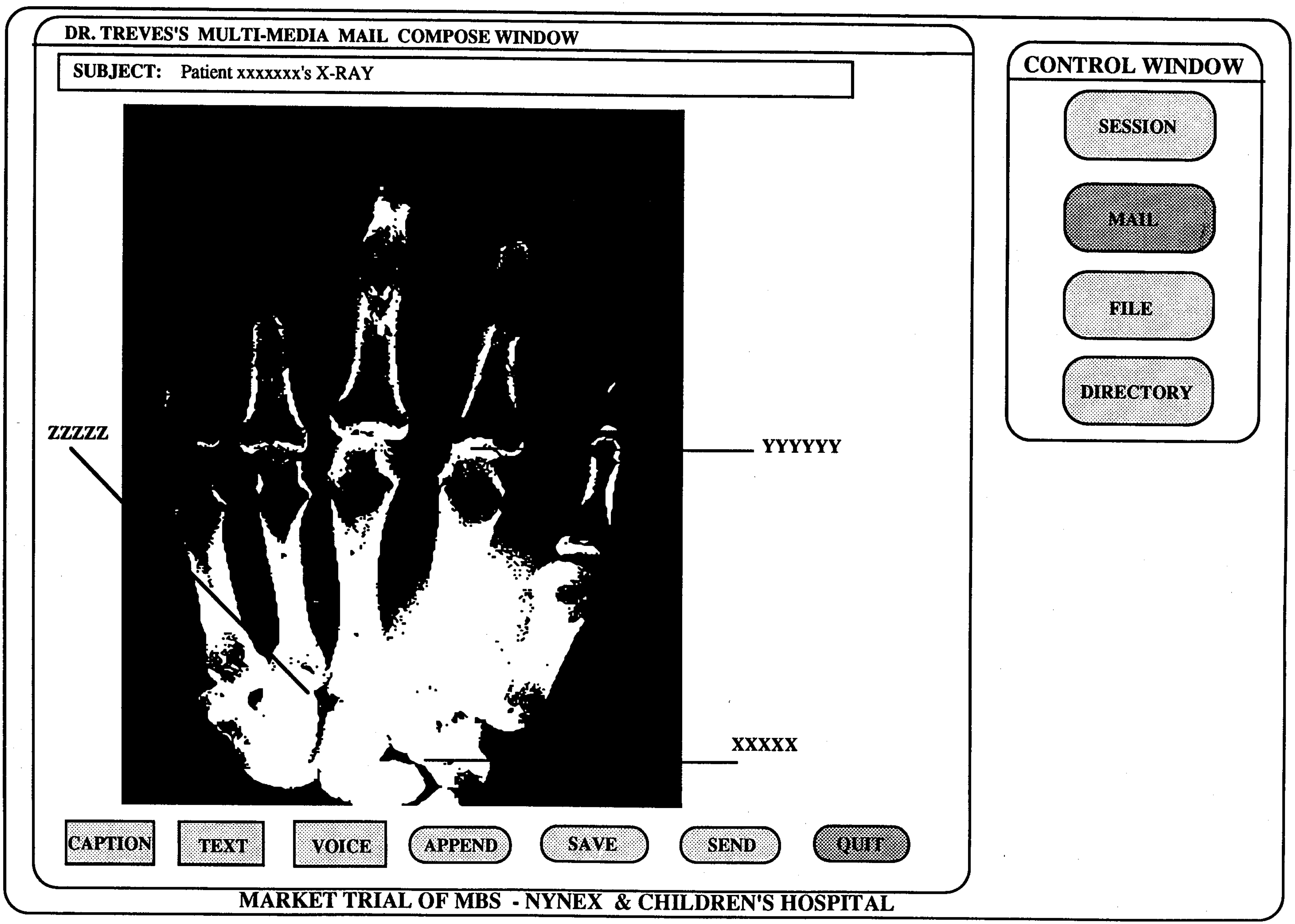
MBS SCREEN WHEN DR. TREVES WANTS TO SAVE THE COMPOSED MULTI-MEDIA MAIL MESSAGE



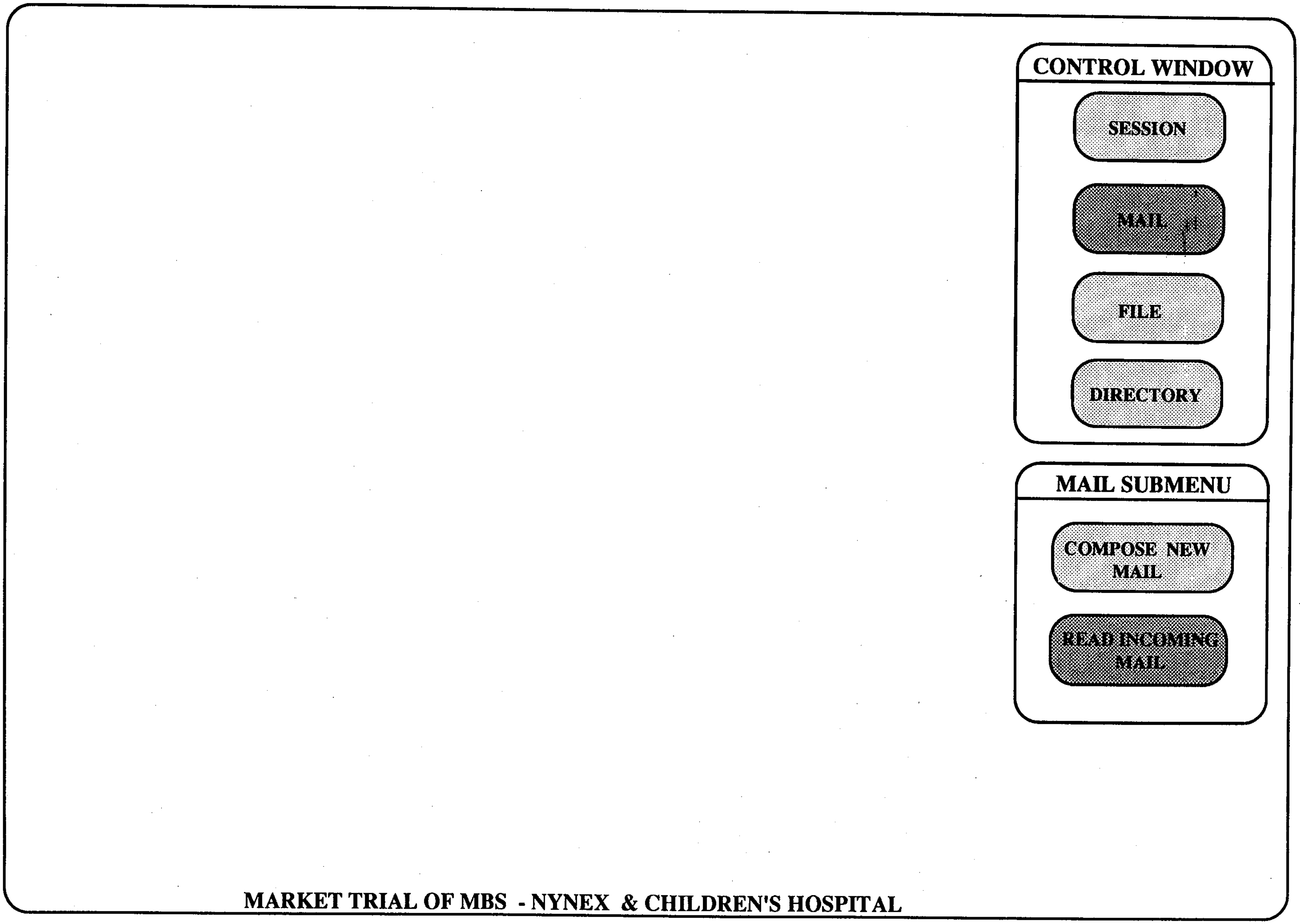
MBS SCREEN WHEN DR. TREVES WANTS TO SEND THE COMPOSED MULTI-MEDIA MAIL MESSAGE



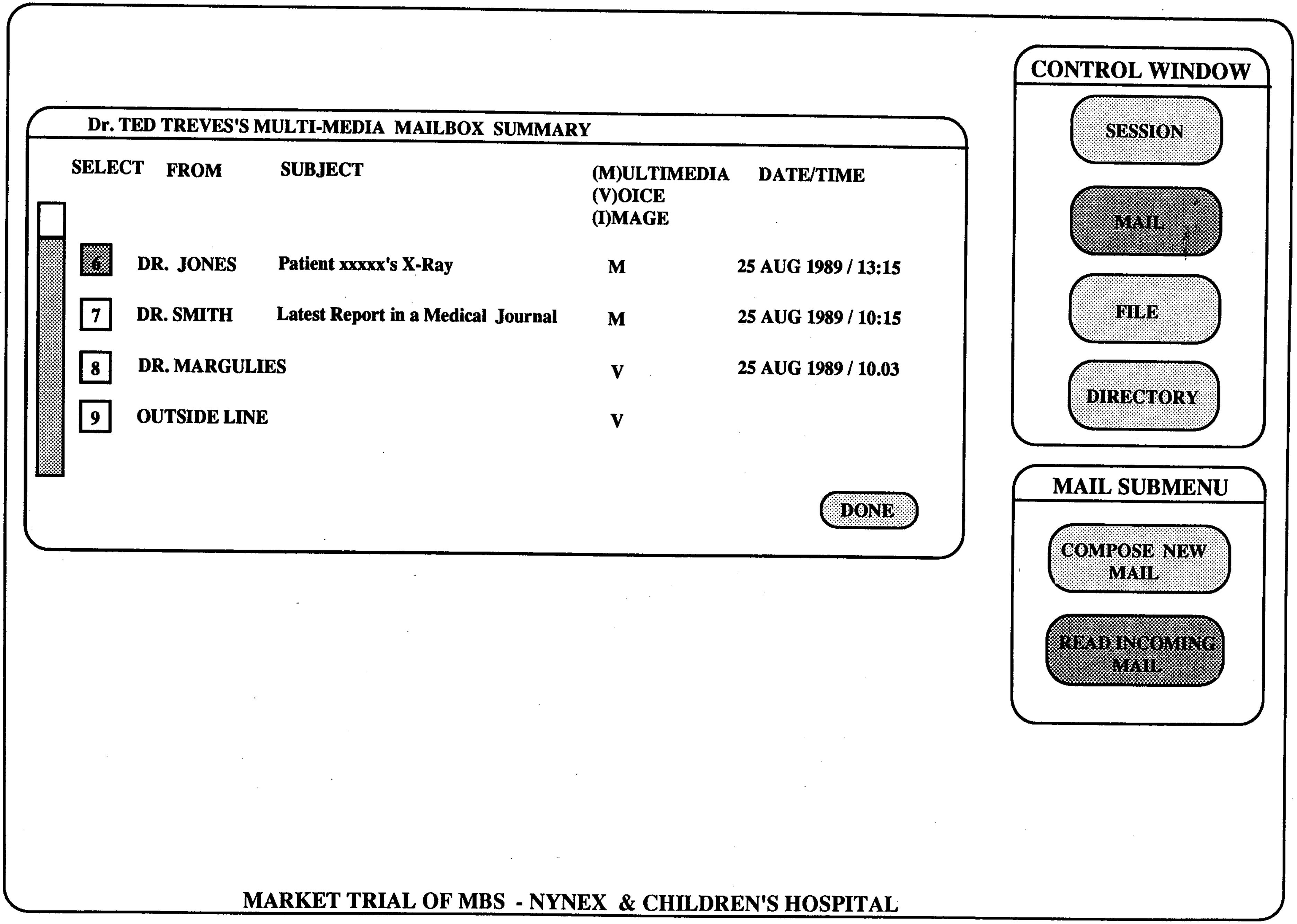
MBS SCREEN WHEN DR. TREVES IS DONE COMPOSING THE MULTI-MEDIA MAIL MESSAGE



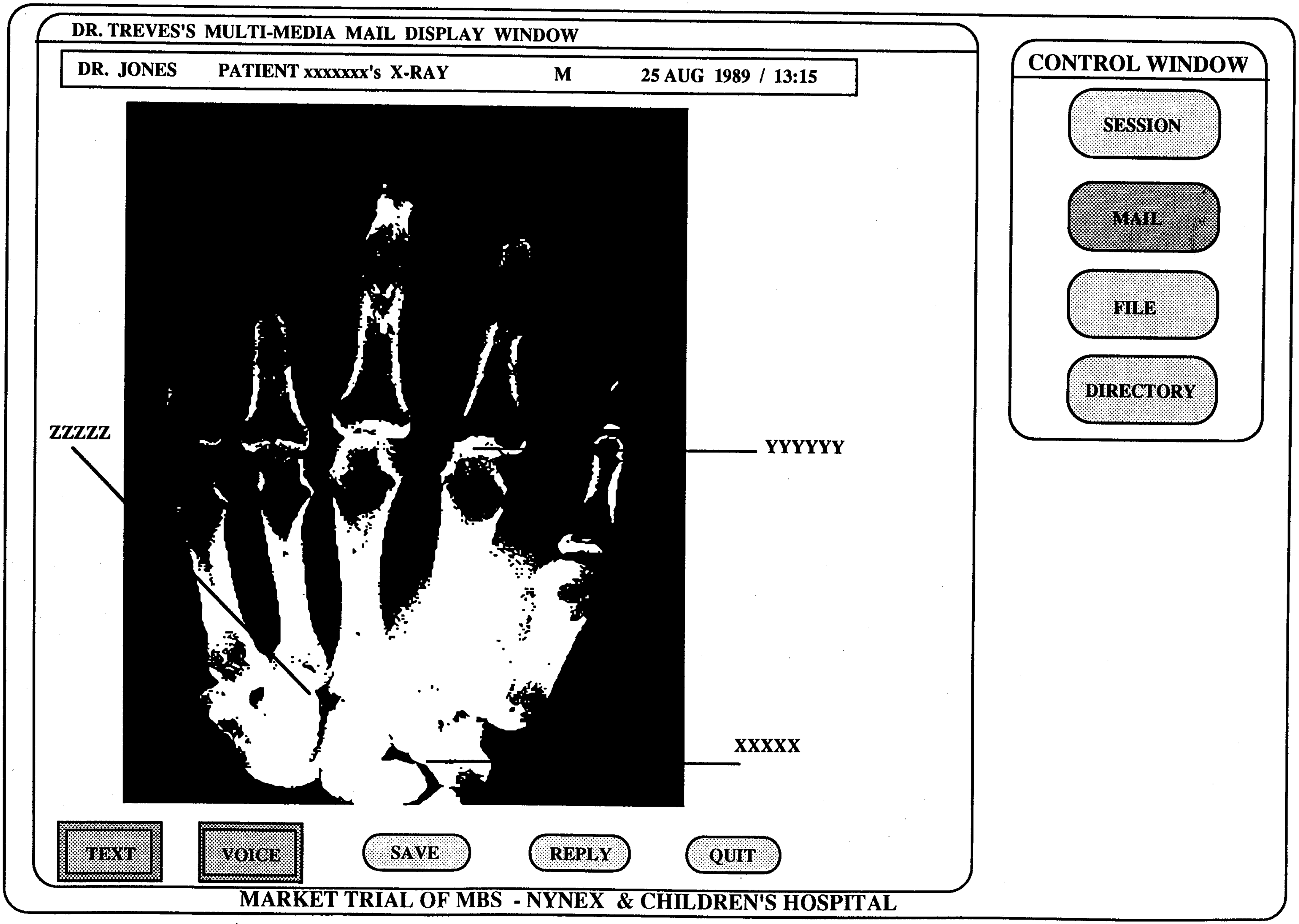
MBS INITIAL SCREEN WHEN MAIL IS SELECTED AND DR. TREVES WANTS TO READ INCOMING MAIL



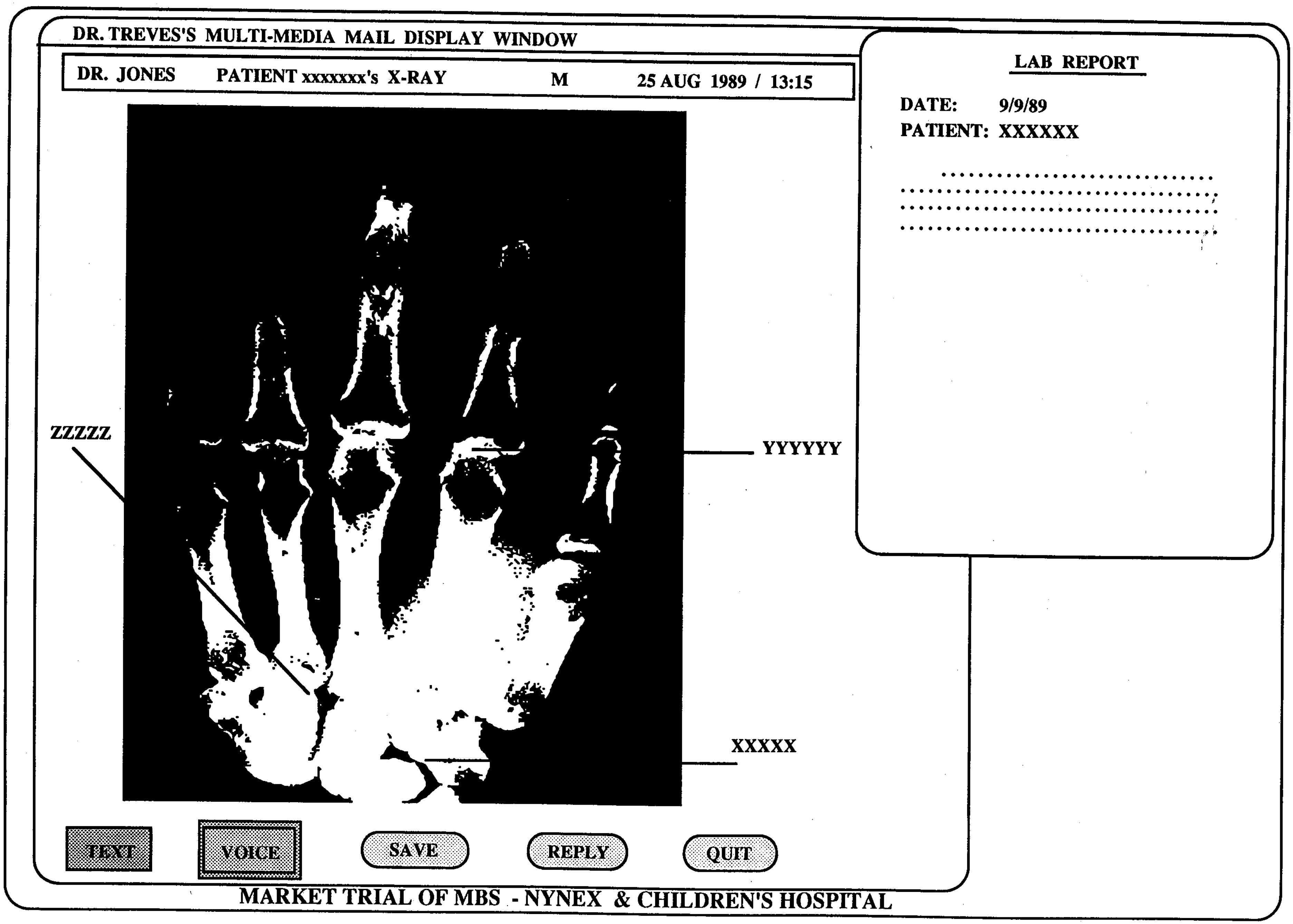
MBS SCREEN WHEN DR. TREVES IS VIEWING THE MAILBOX SUMMARY



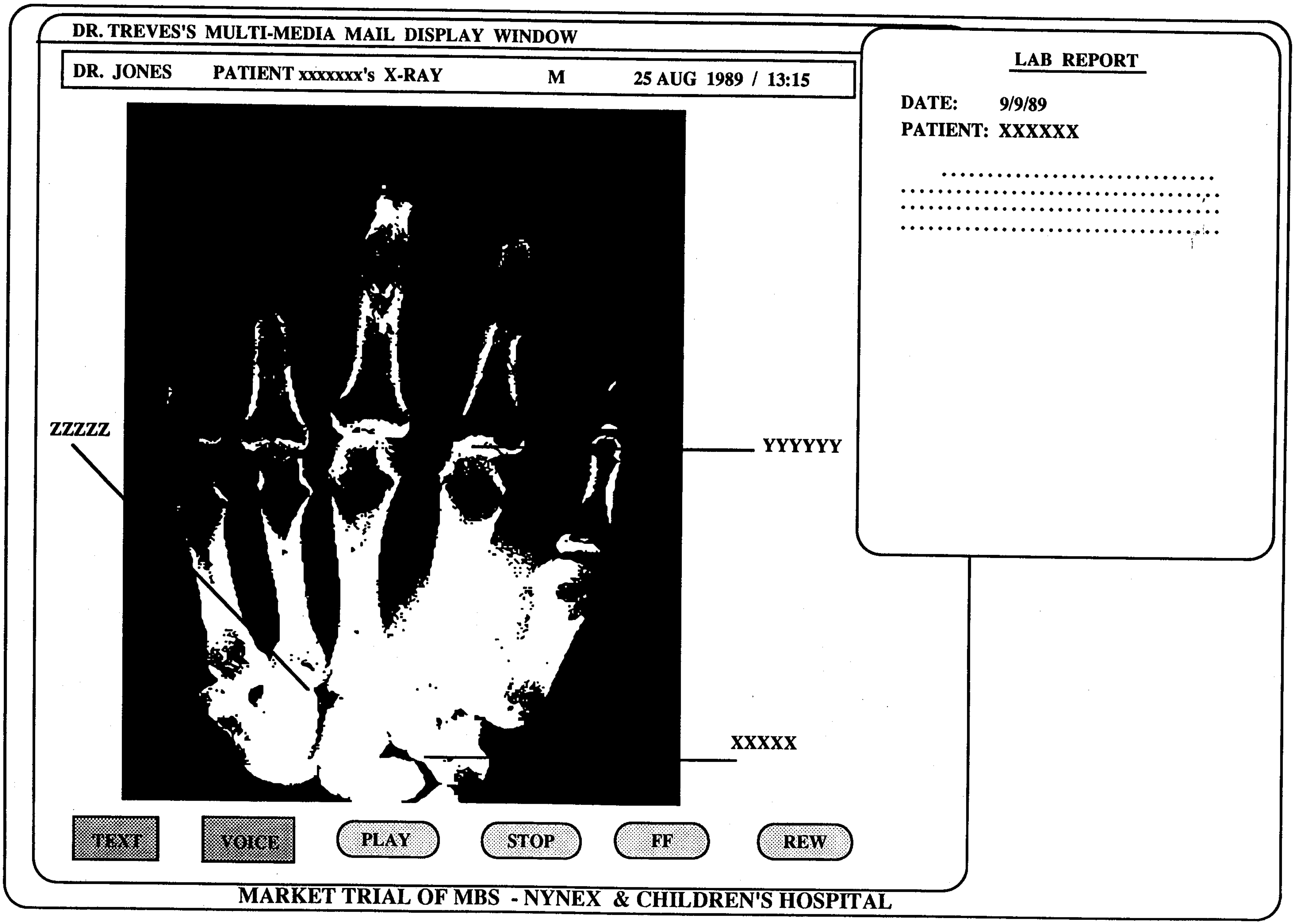
MBS SCREEN WHEN DR. TREVES IS DISPLAYING A CHOSEN MAIL MESSAGE



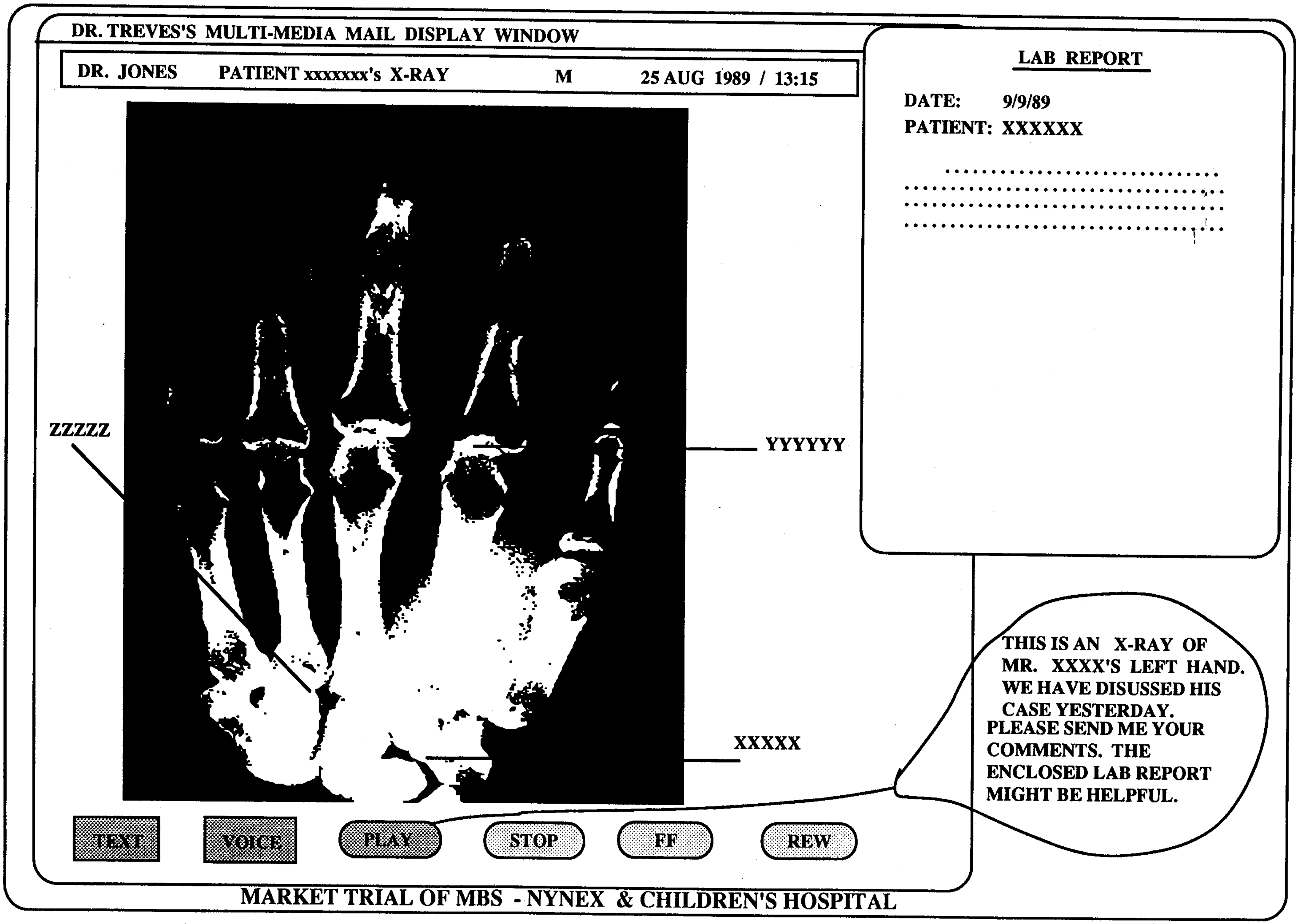
MBS SCREEN WHEN DR. TREVES IS DISPLAYING TEXT PORTION OF CHOSEN MAIL MESSAGE



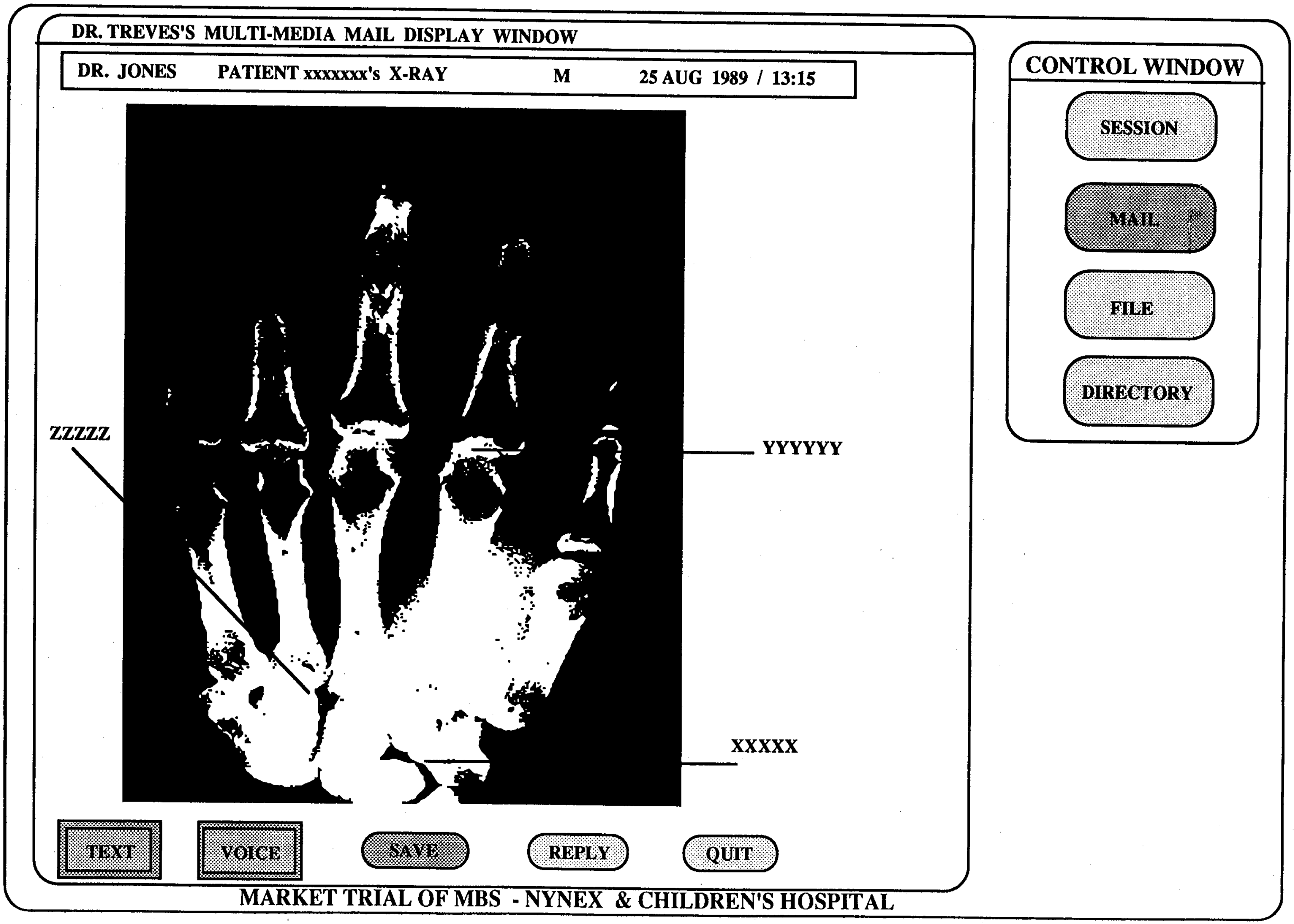
MBS SCREEN WHEN DR. TREVES WANTS TO LISTEN TO VOICE PORTION OF MAIL MESSAGE



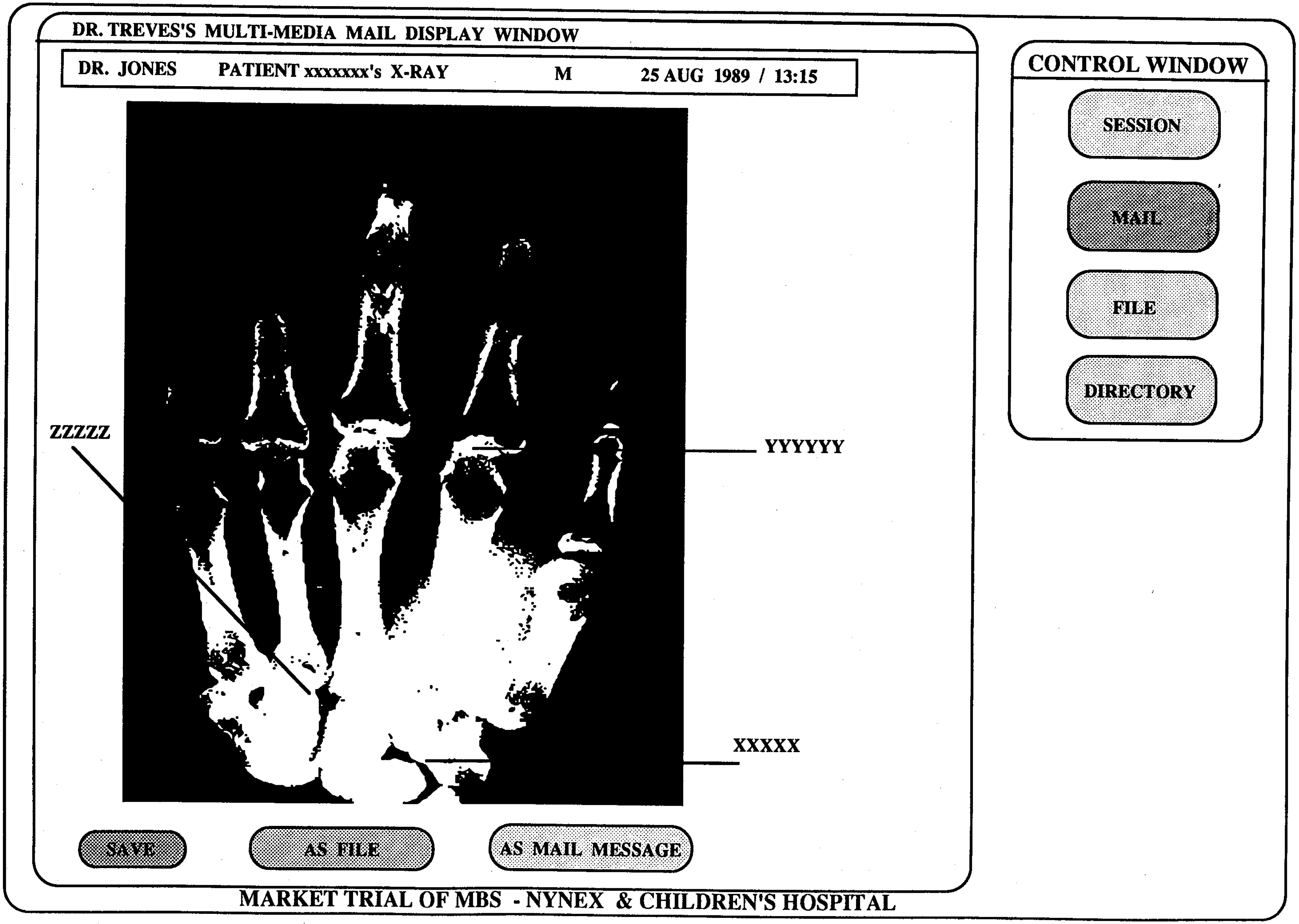
MBS SCREEN WHEN DR. TREVES IS LISTENING TO VOICE PORTION OF CHOSEN MAIL MESSAGE



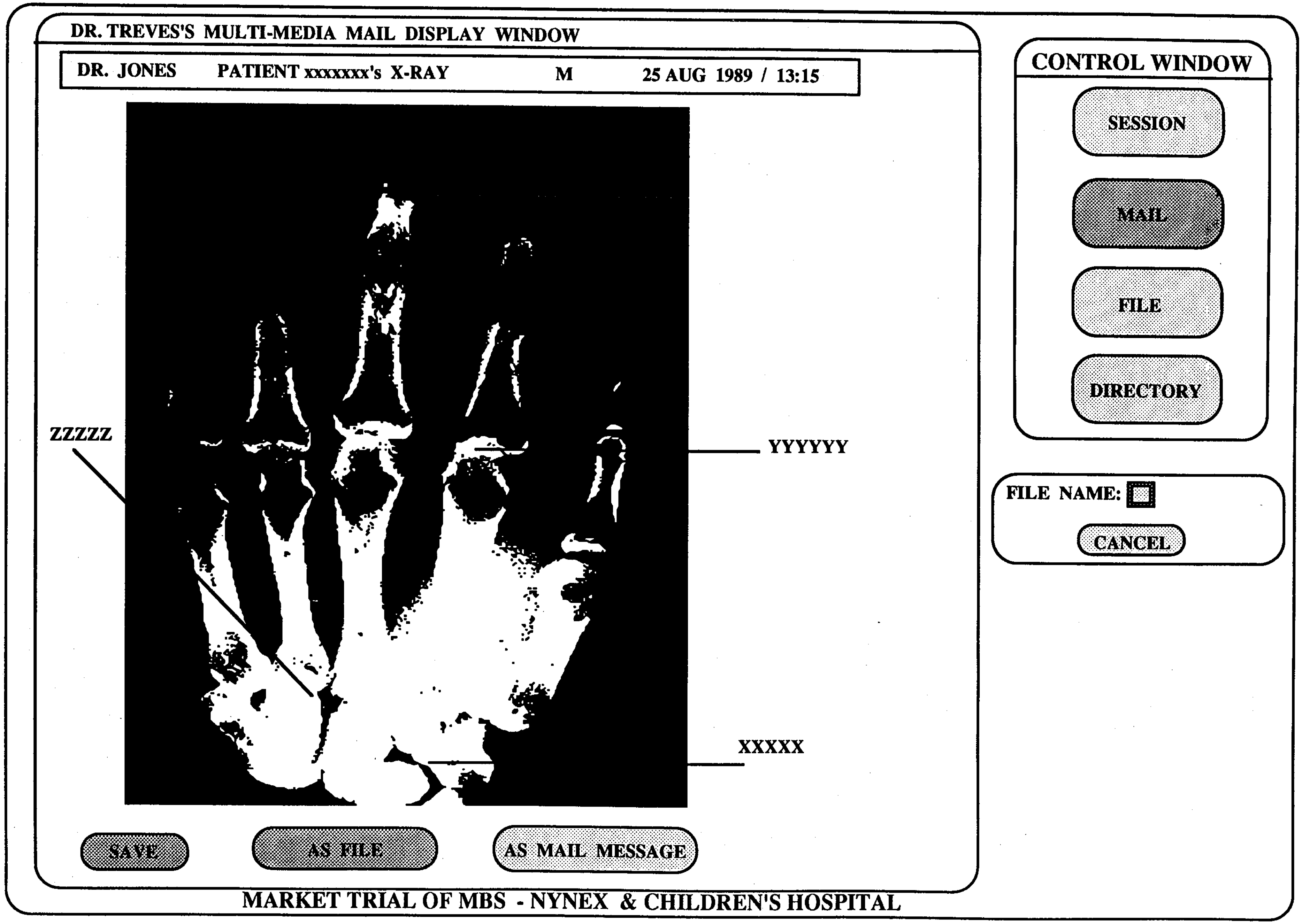
MBS SCREEN WHEN DR. TREVES WANTS TO SAVE THE MAIL MESSAGE



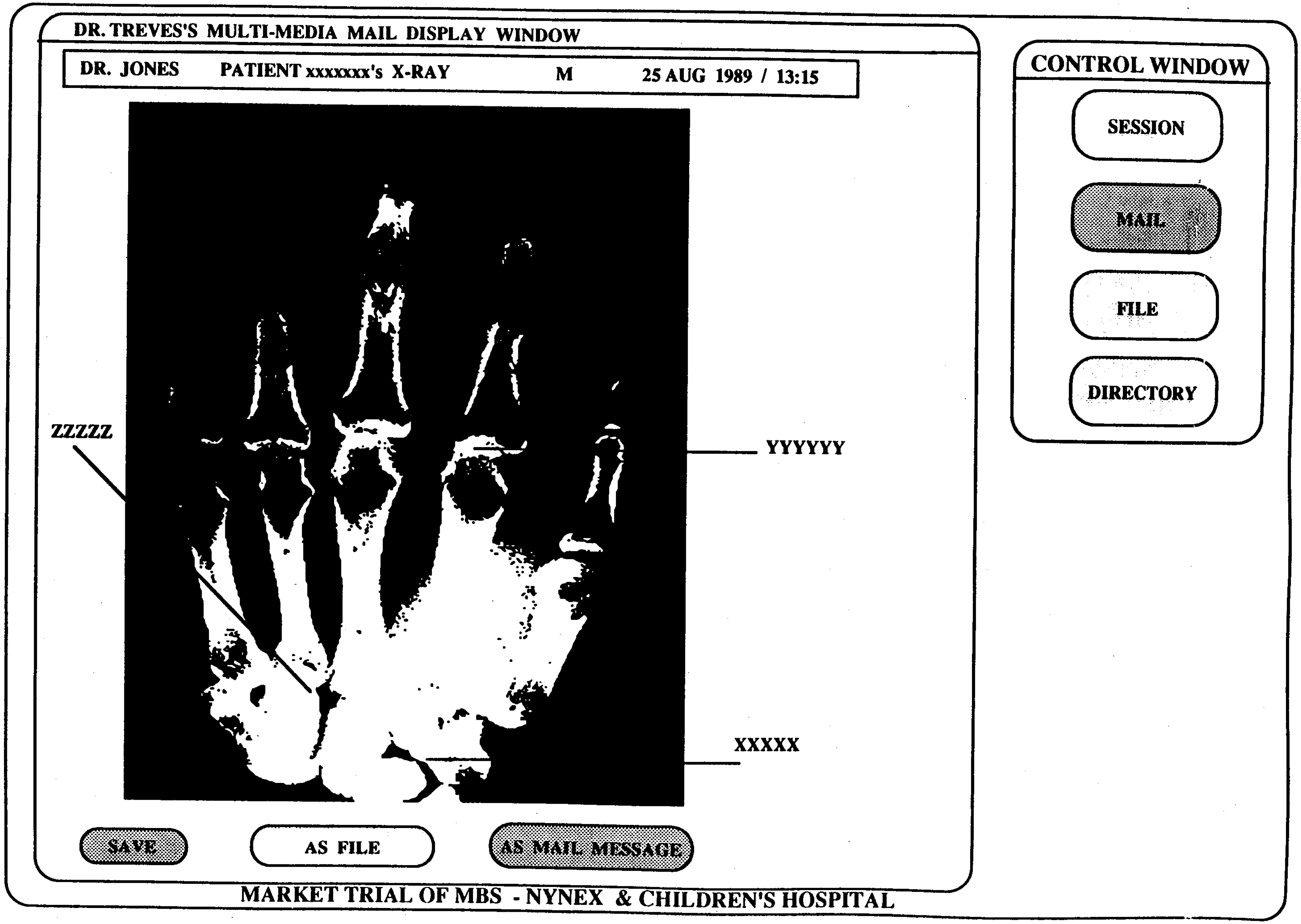
MBS SCREEN WHEN MAIL IS SELECTED AND DR. TREVES WANTS TO SAVE THE MAIL MESSAGE



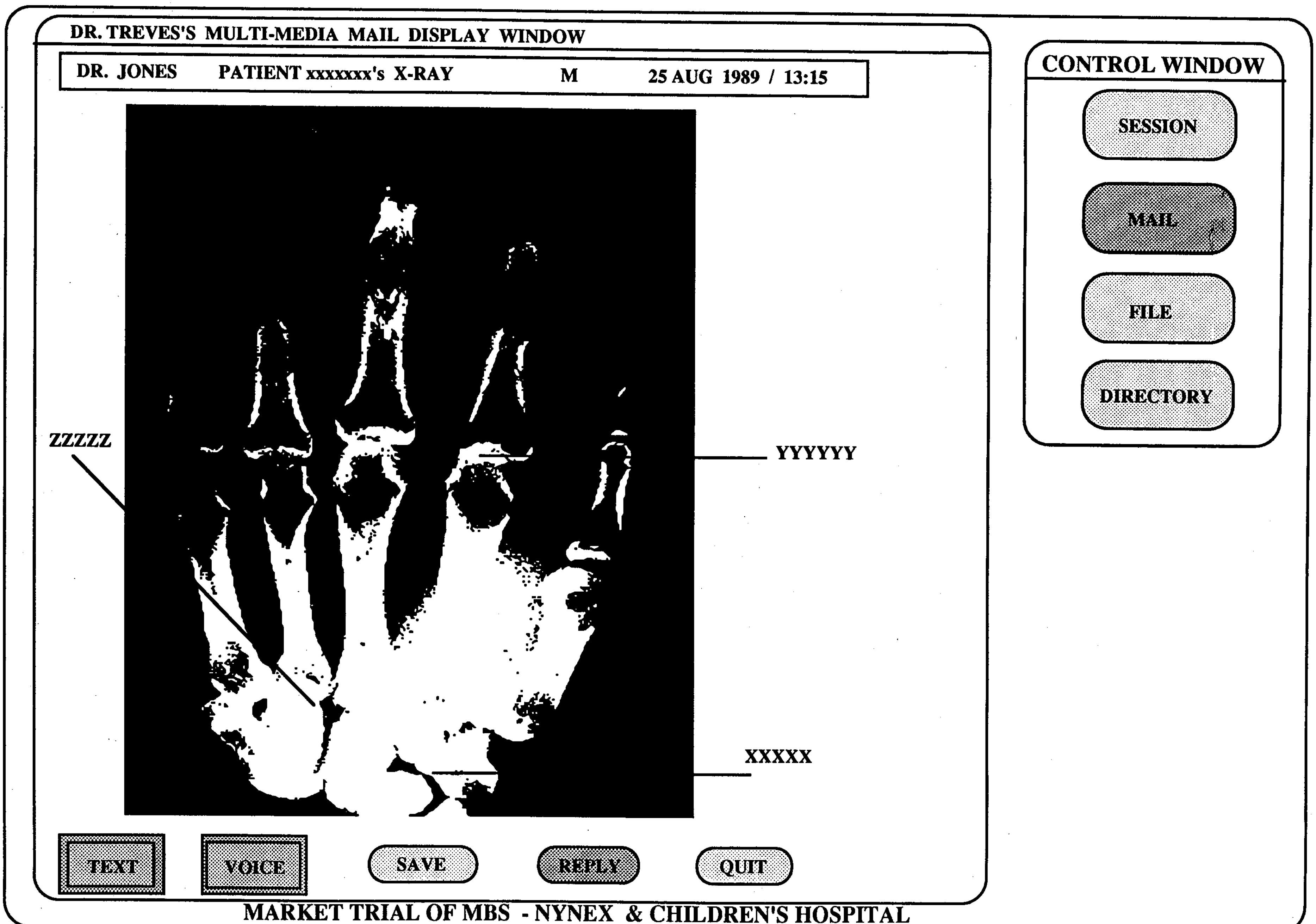
MBS SCREEN WHEN DR. TREVES WANTS TO SAVE THE MAIL MESSAGE AS A FILE



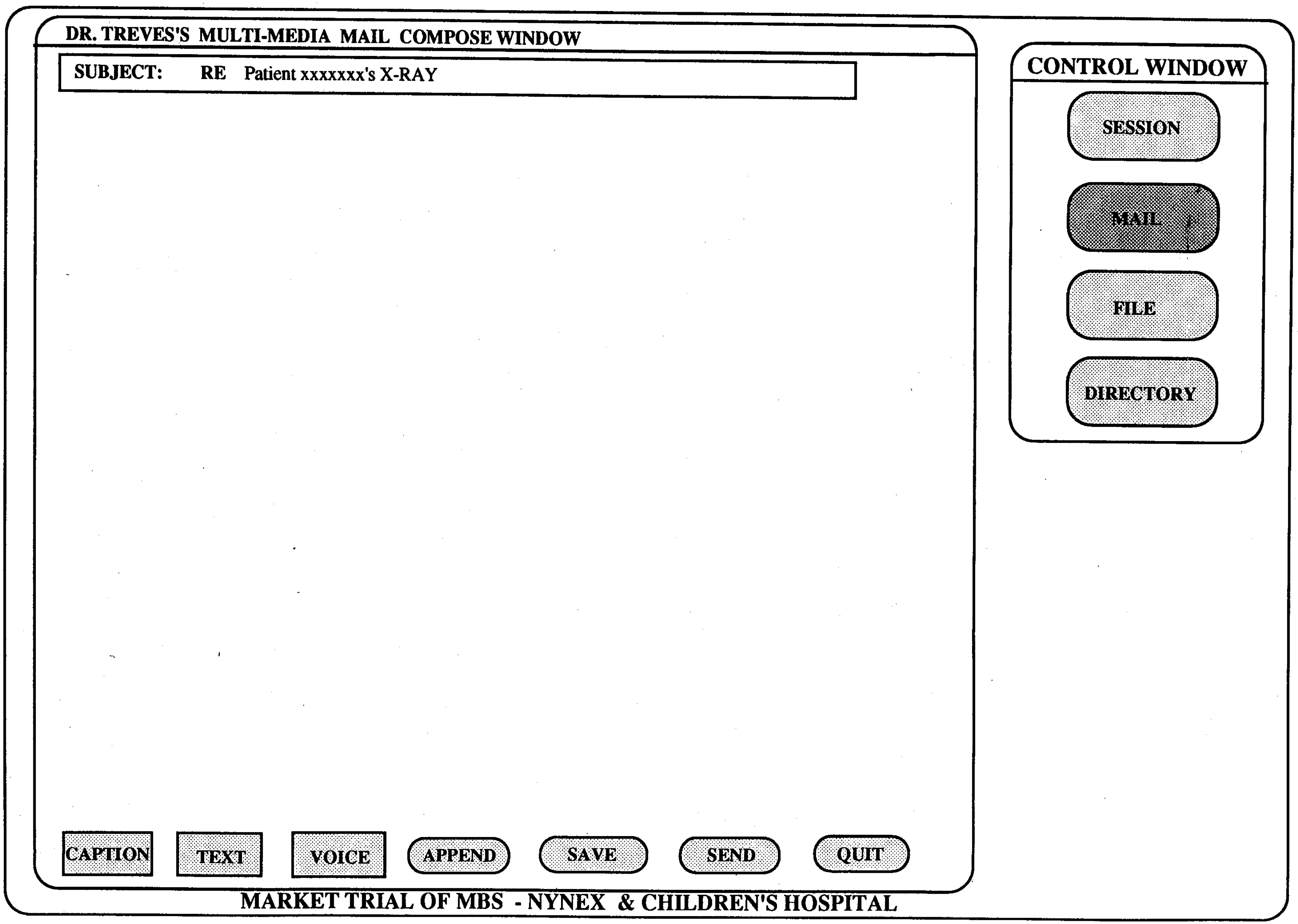
MBS SCREEN WHEN DR. TREVES WANTS TO SAVE THE MESSAGE AS MAIL



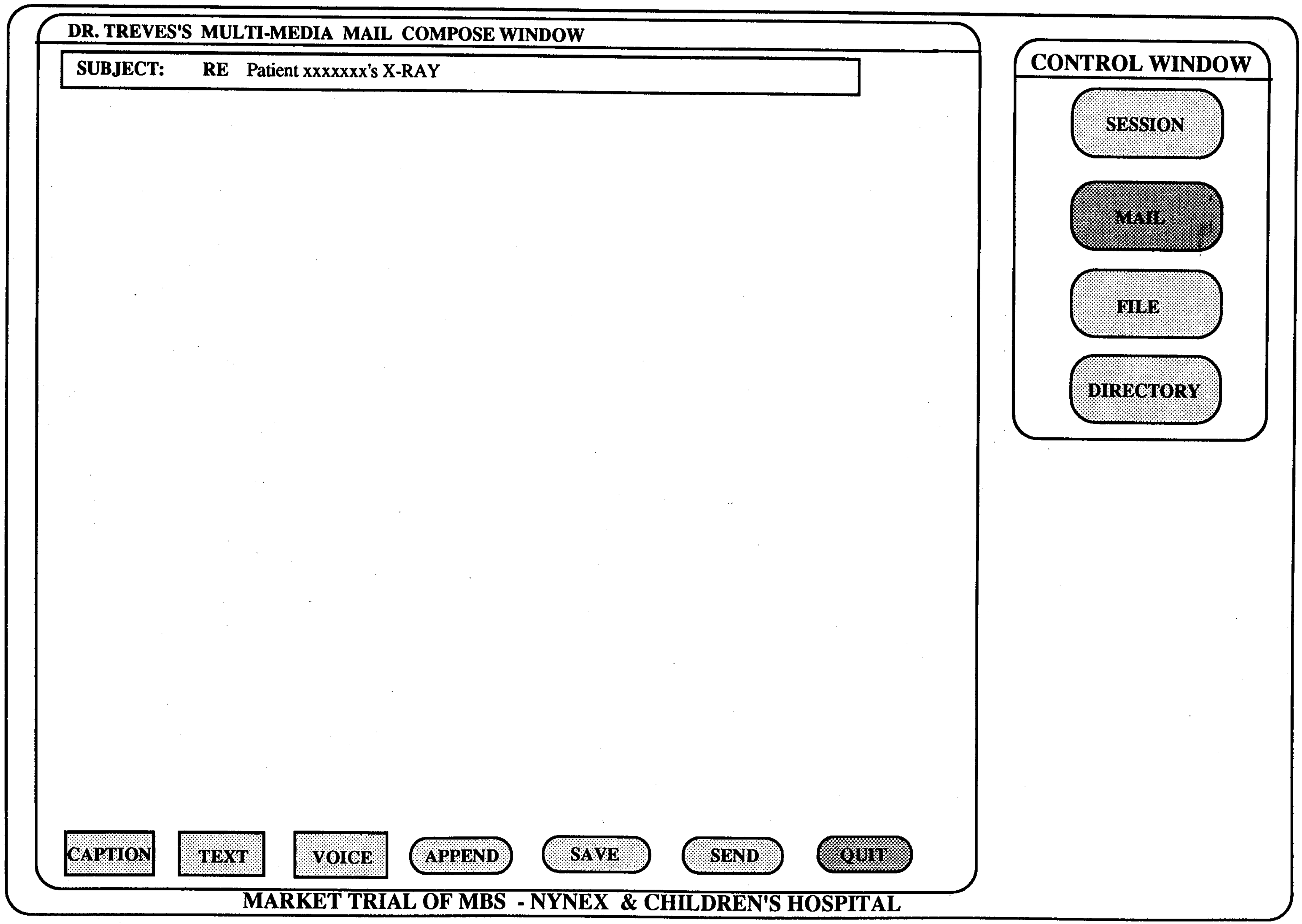
MBS SCREEN WHEN DR. TREVES WANTS TO REPLY TO THE CHOSEN MAIL MESSAGE



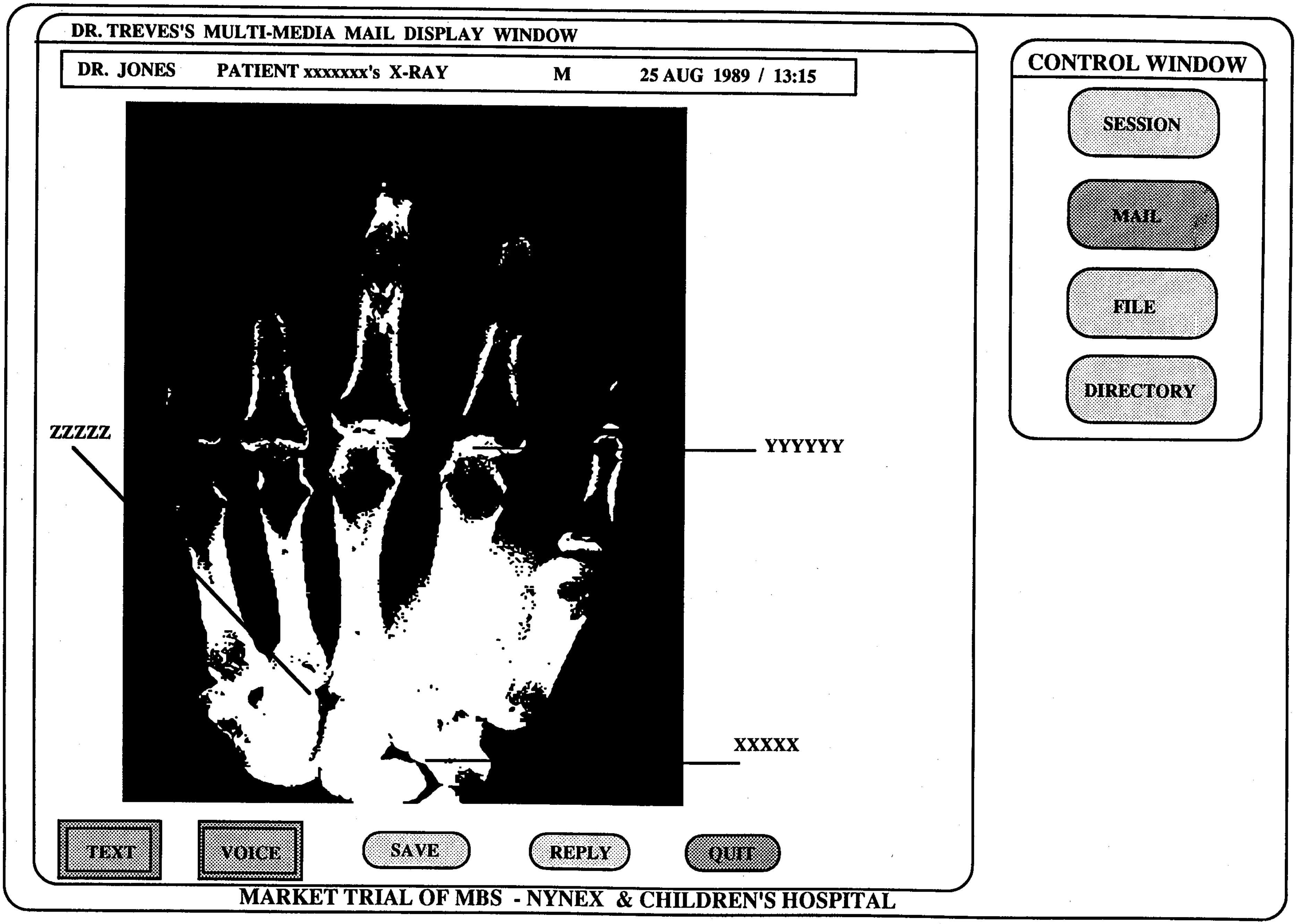
MBS SCREEN WHEN DR. TREVES WANTS TO COMPOSE THE REPLY



MBS SCREEN WHEN DR. TREVES IS DONE COMPOSING HIS REPLY



MBS SCREEN WHEN DR. TREVES IS DONE READING THE CHOSEN MESSAGE



MBS SCREEN WHEN DR. TREVES IS DONE VIEWING THE MAILBOX SUMMARY

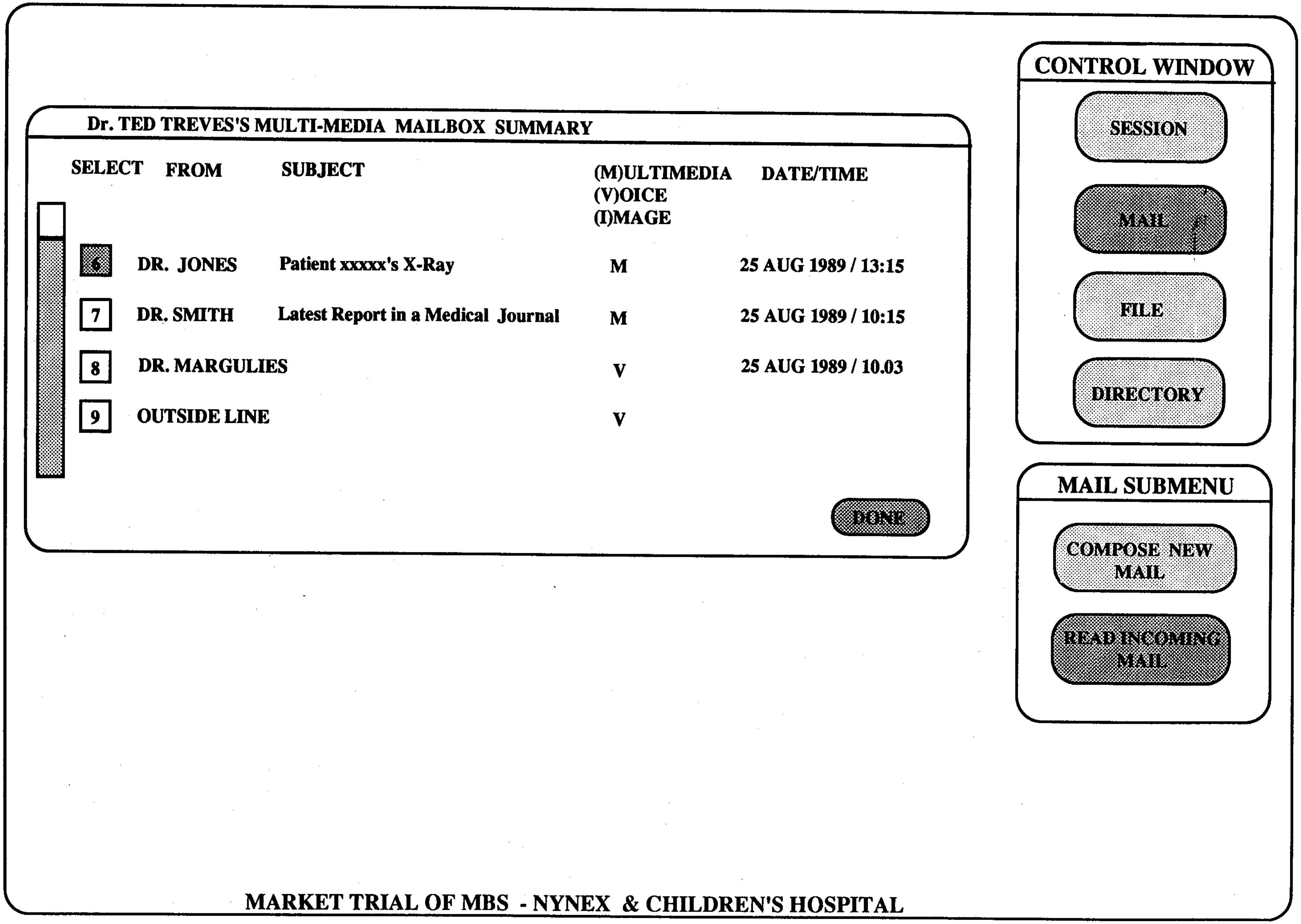


Figure 3.x is s detailed description of the state diagram for the mail service. It represents the states as detailed by the screen presentation formats and includes both detailed input and output elements.

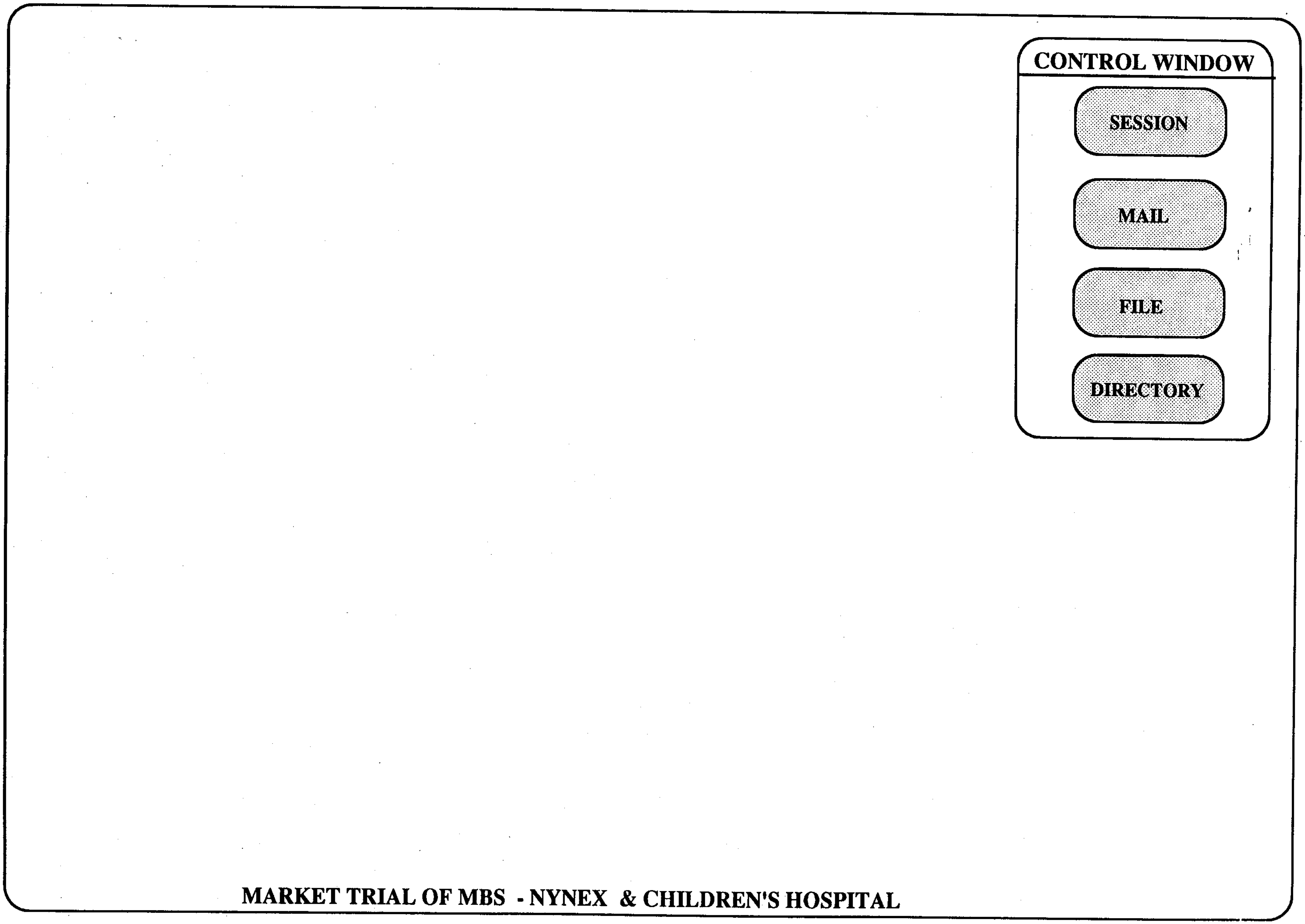
Figure 3.x Mail State Model

3.2.3.3 File

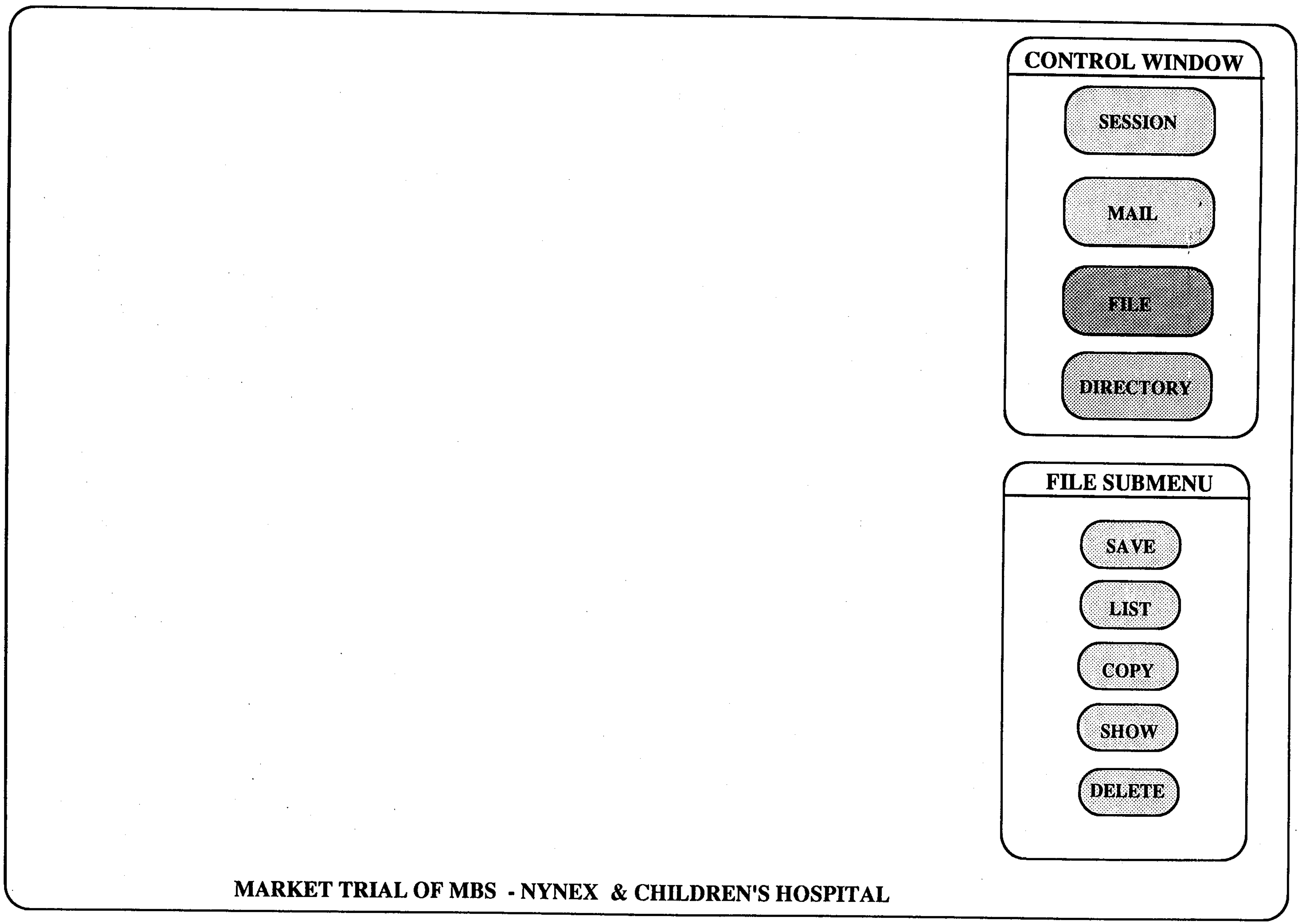
File services is a third example that details the structures of the end user interface. These are shown in Figure 3.x.

Figure 3.x File Screen Layout

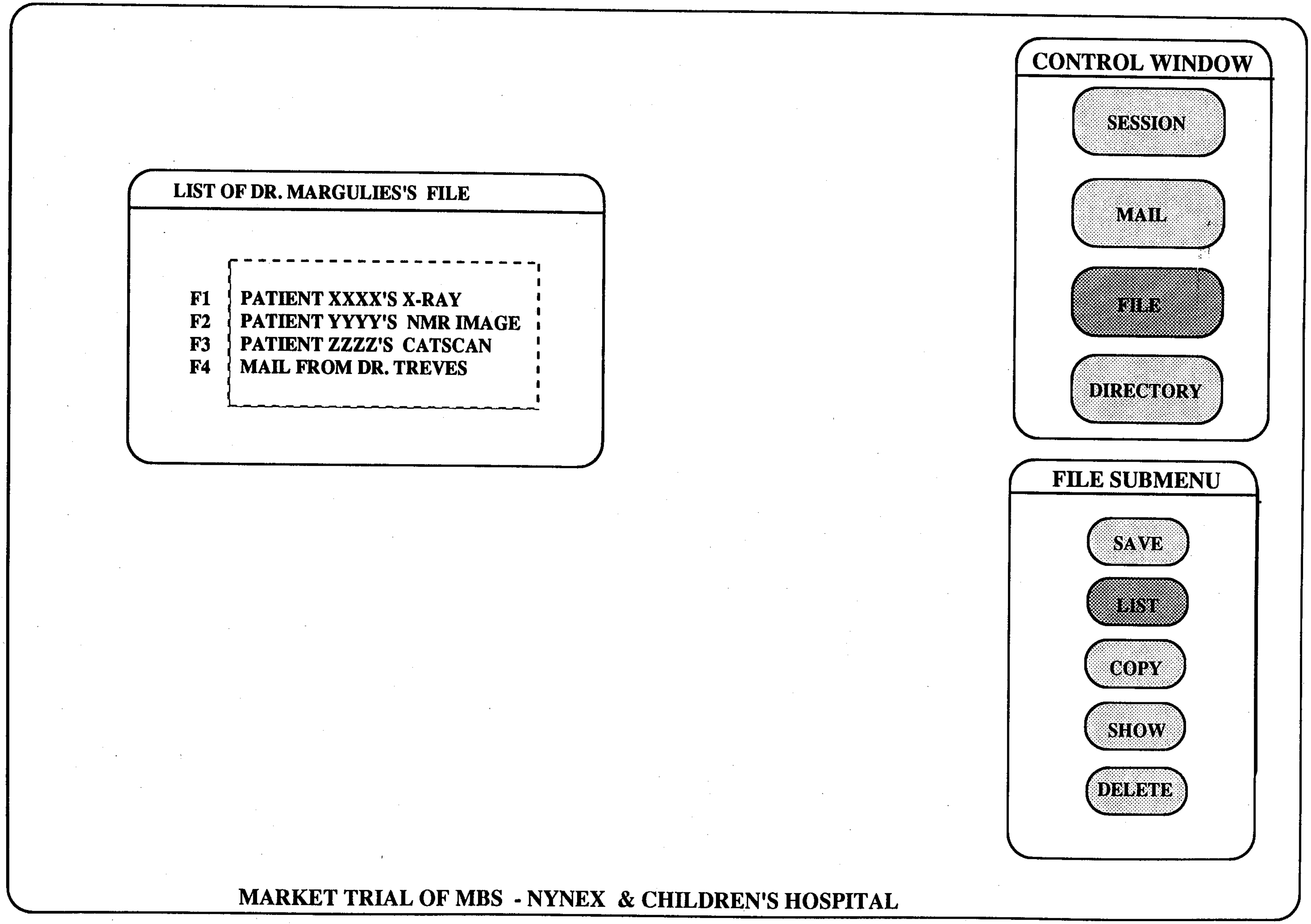
MBS INITIAL SCREEN WHEN WORKSTATION PROGRAM IS LOADED



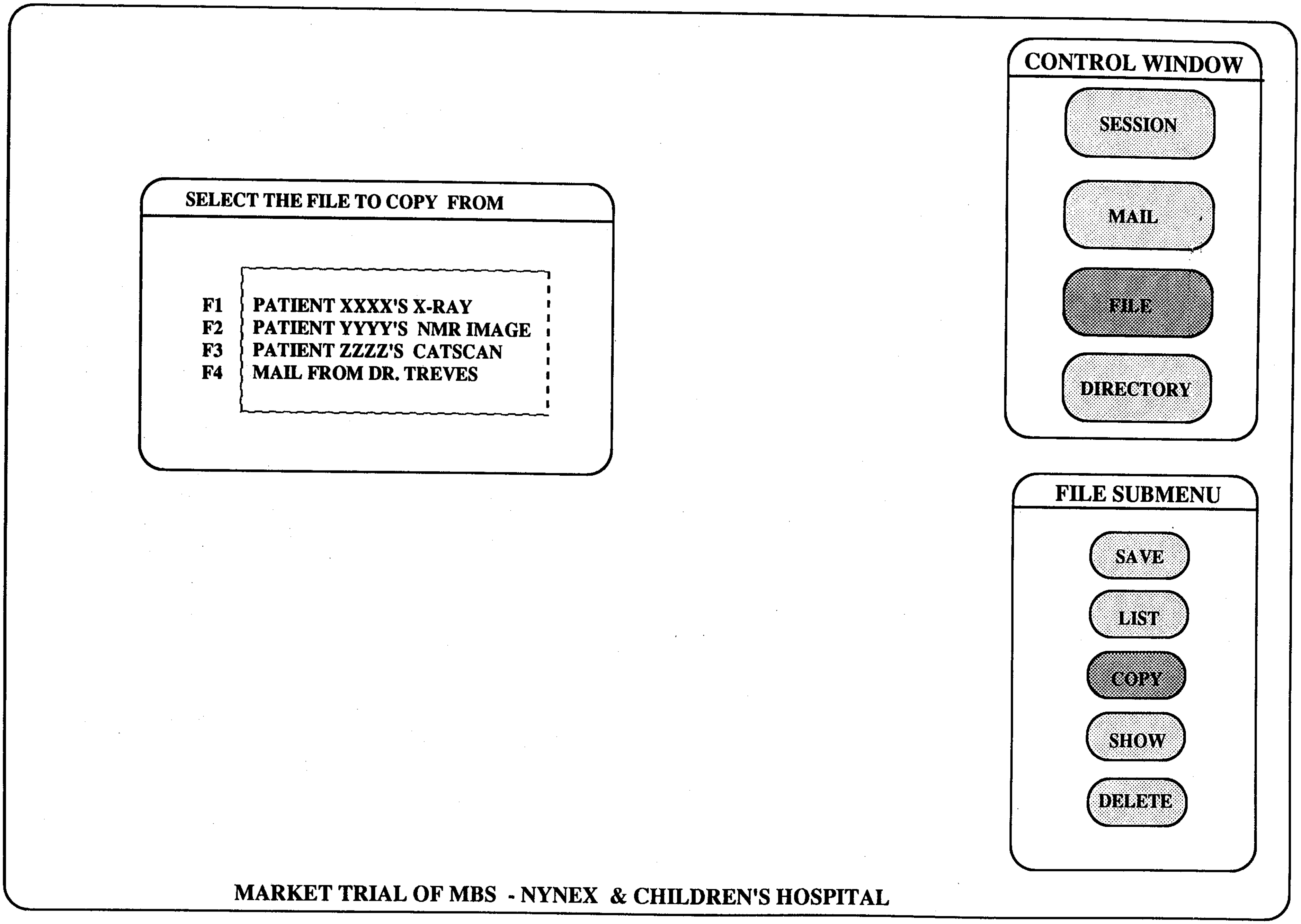
MBS SCREEN WHEN FILE ICON IS CHOSEN



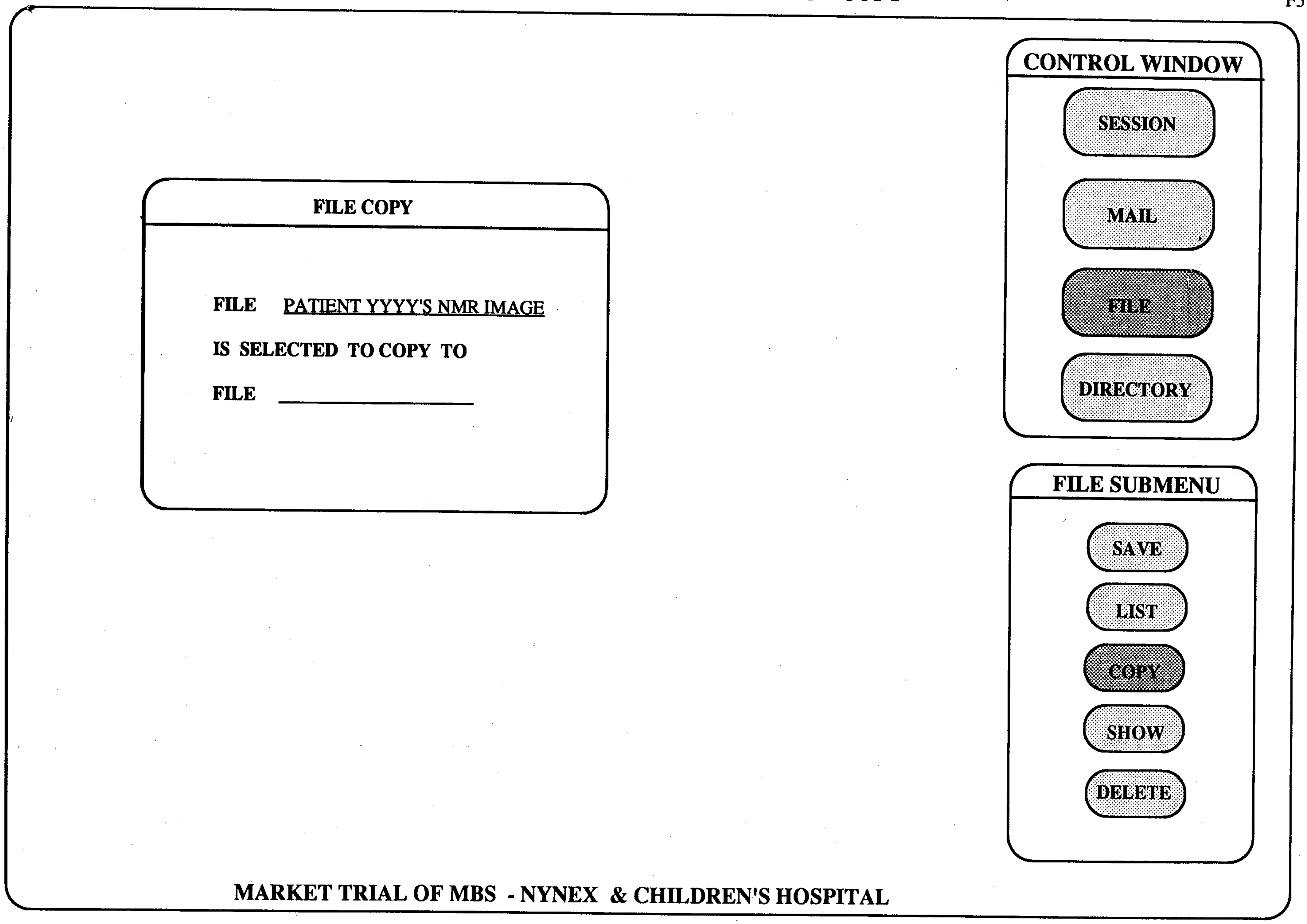
MBS SCREEN WHEN LIST ICON IS CHOSEN



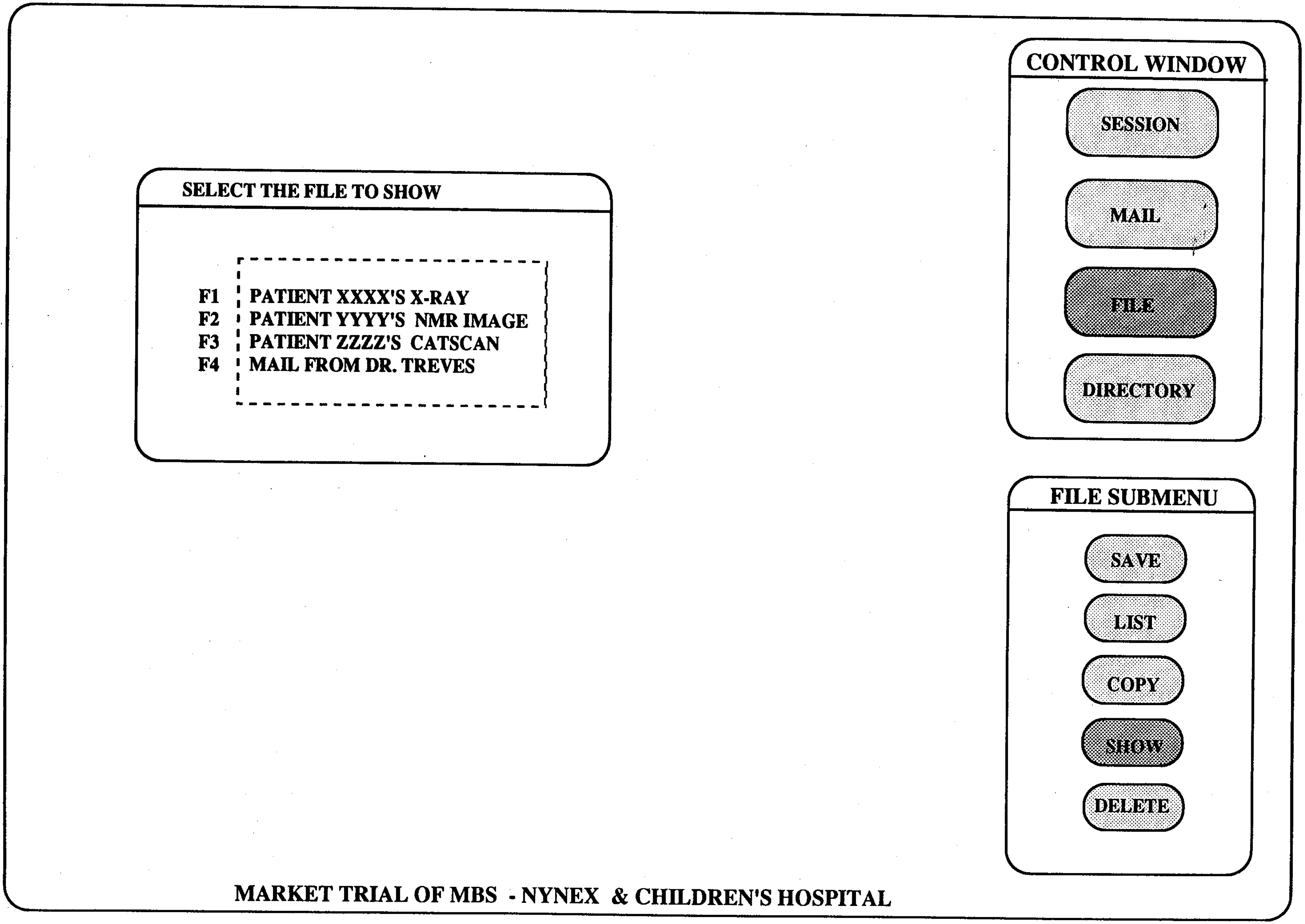
MBS SCREEN WHEN COPY ICON IS CHOSEN



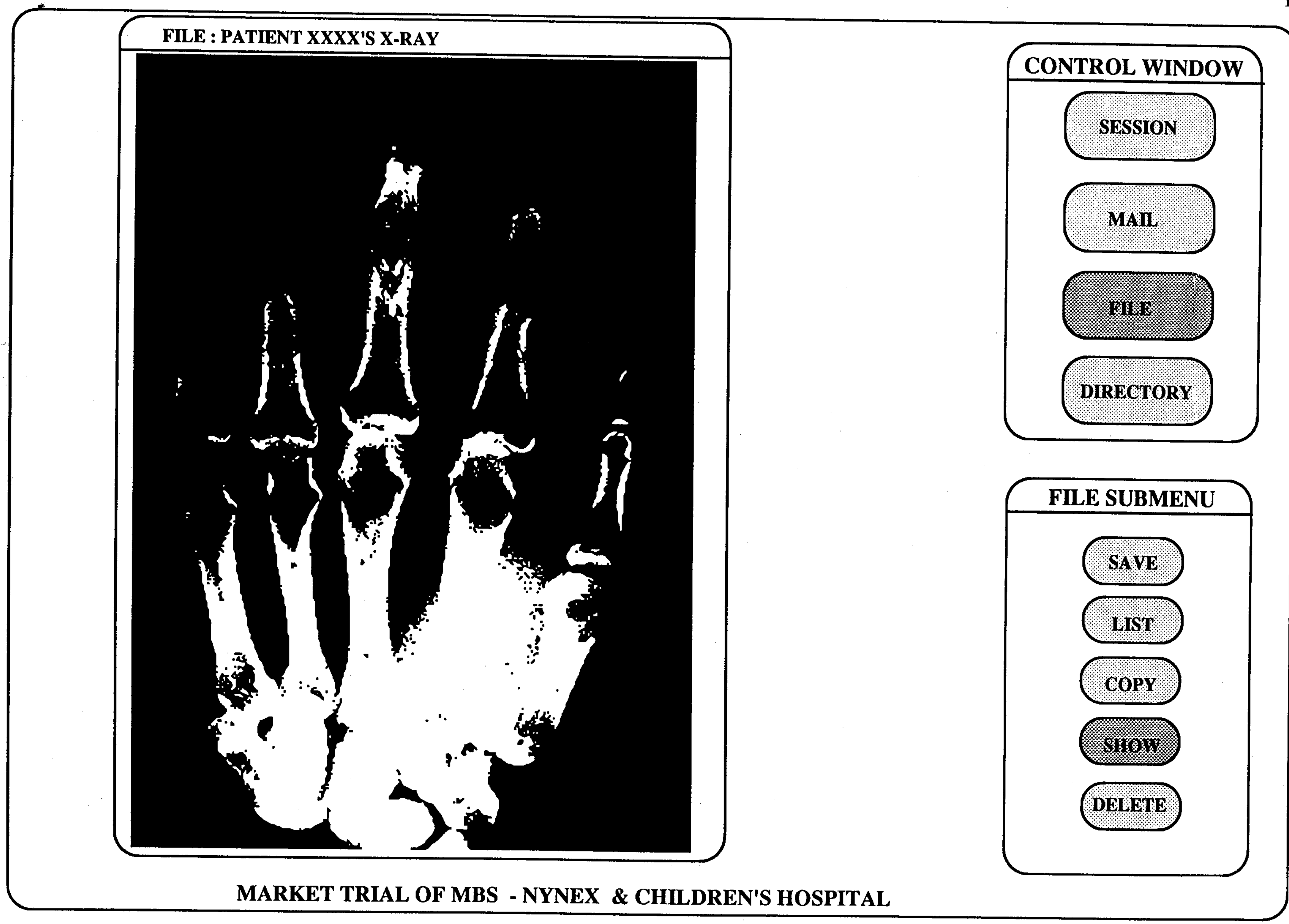
MBS SCREEN WHEN A FILE IS CHOSEN FOR COPY



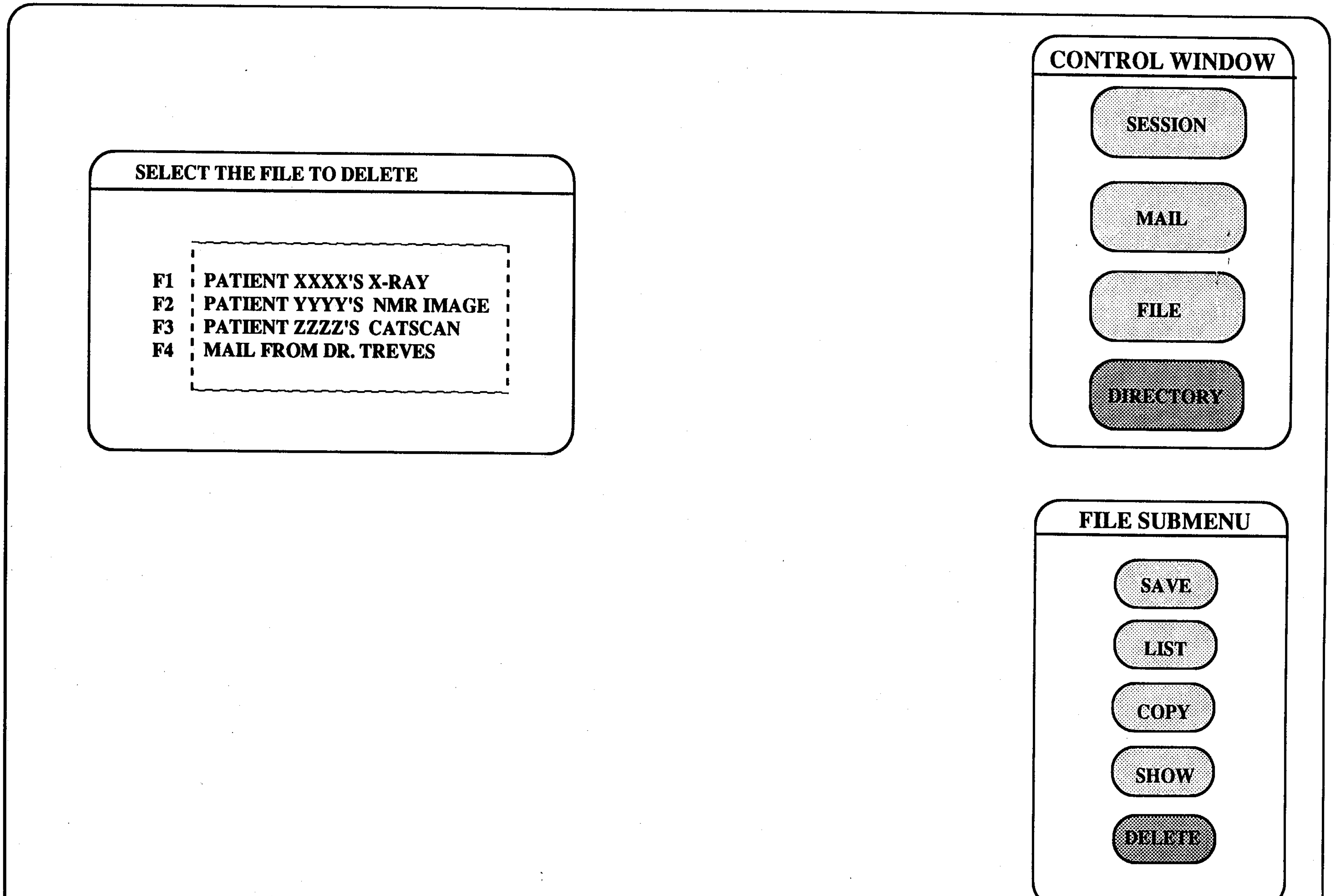
MBS SCREEN WHEN SHOW ICON IS CHOSEN



MBS SCREEN WHEN A FILE IS CHOSEN TO SHOW

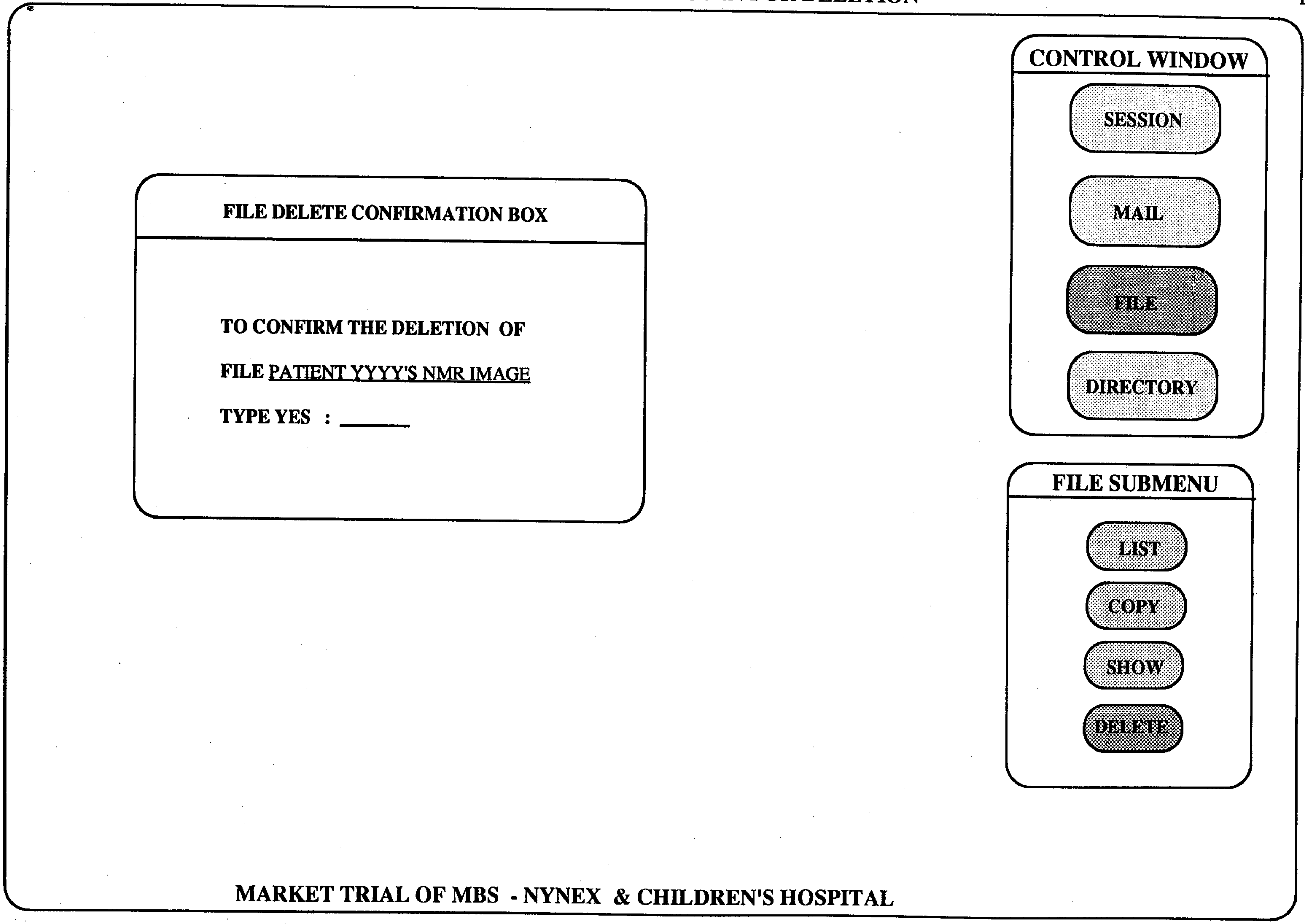


MBS SCREEN WHEN DELETE ICON IS CHOSEN

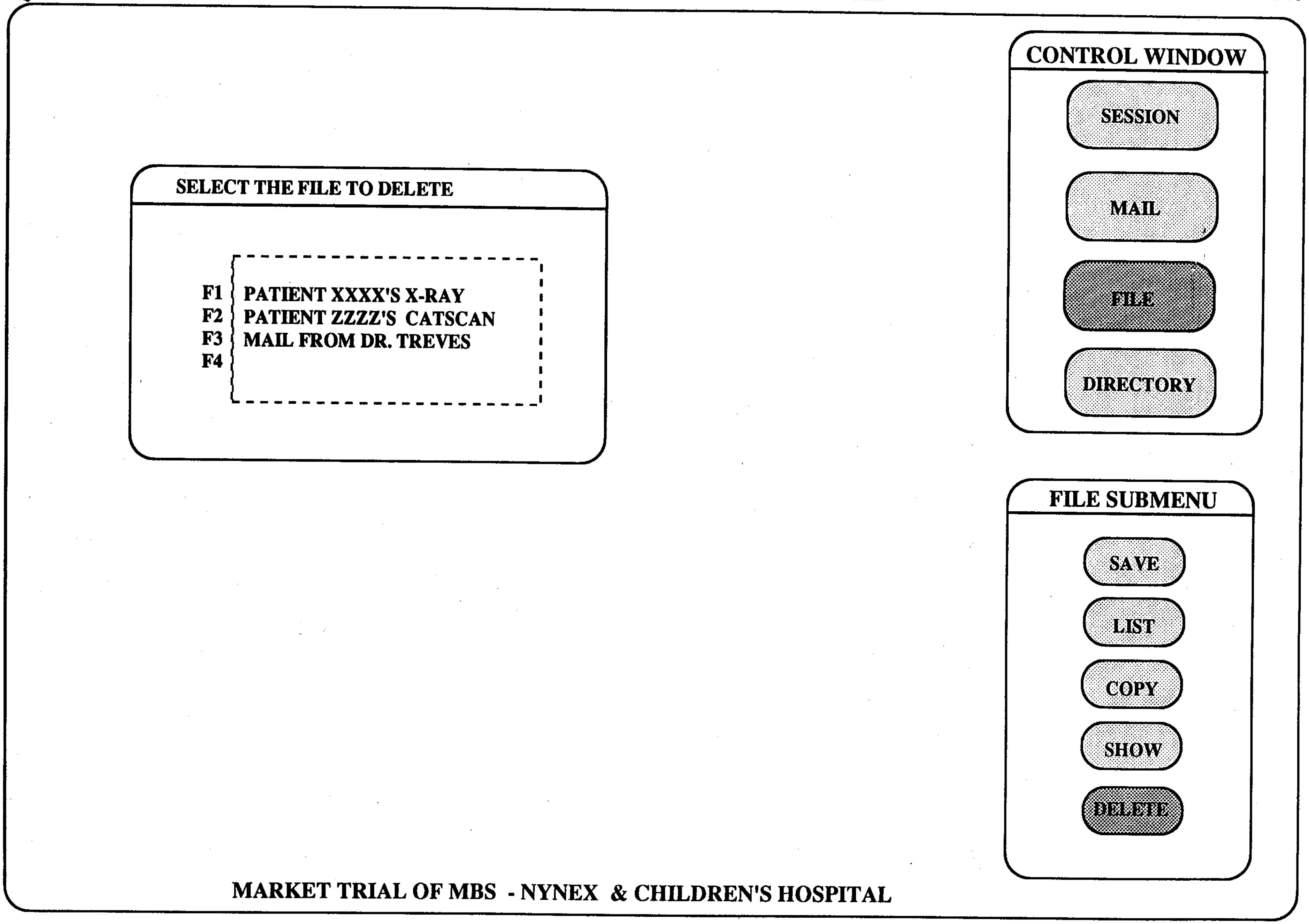


MARKET TRIAL OF MBS - NYNEX & CHILDREN'S HOSPITAL

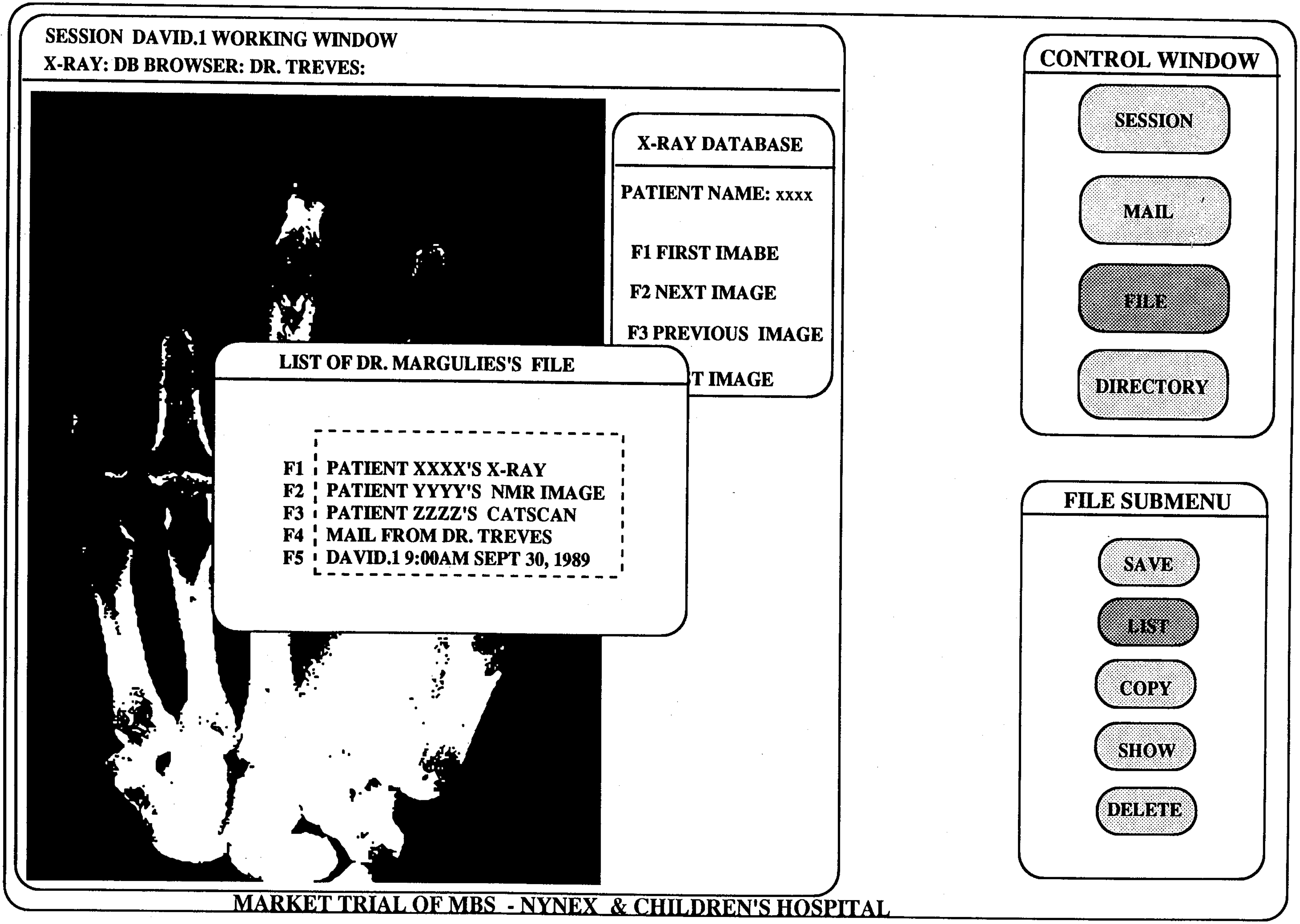
MBS SCREEN WHEN FILE IS CHOSEN FOR DELETION



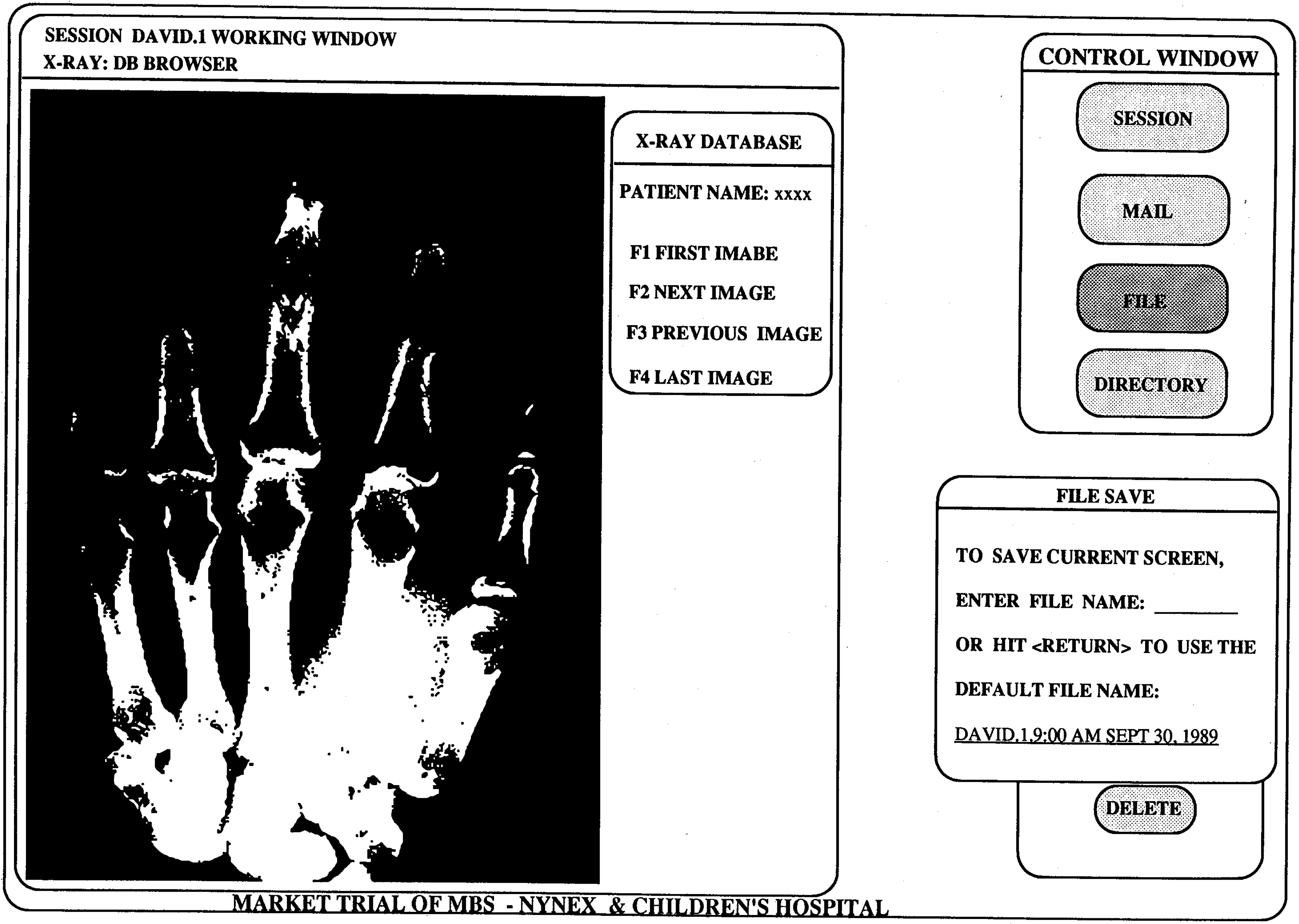
MBS SCREEN WHEN DELETION IS CONFIRMED



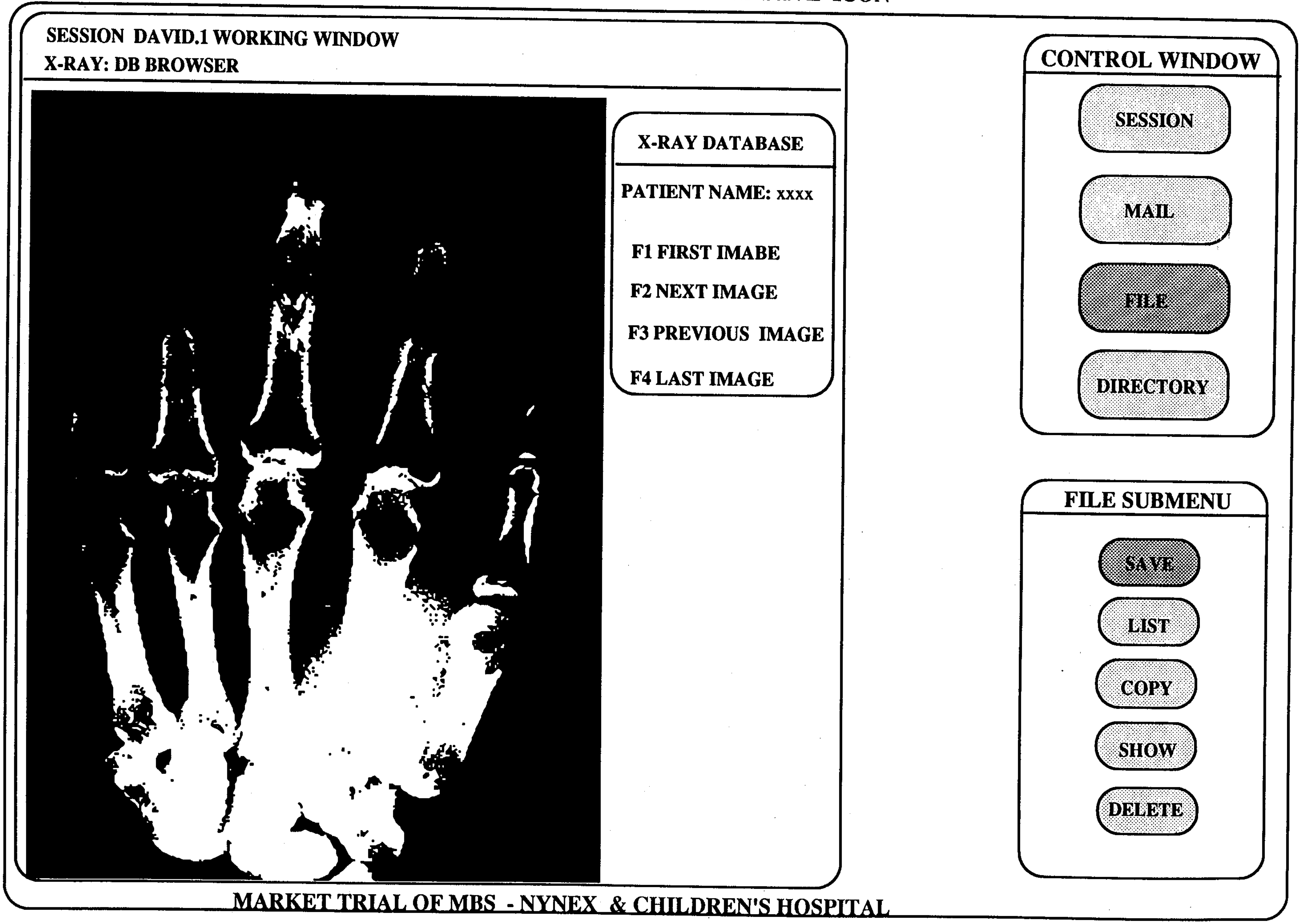
MBS SCREEN OF FILE LIST WHEN THE DEFAULT FILE NAME WAS USED FOR FILE SAVE



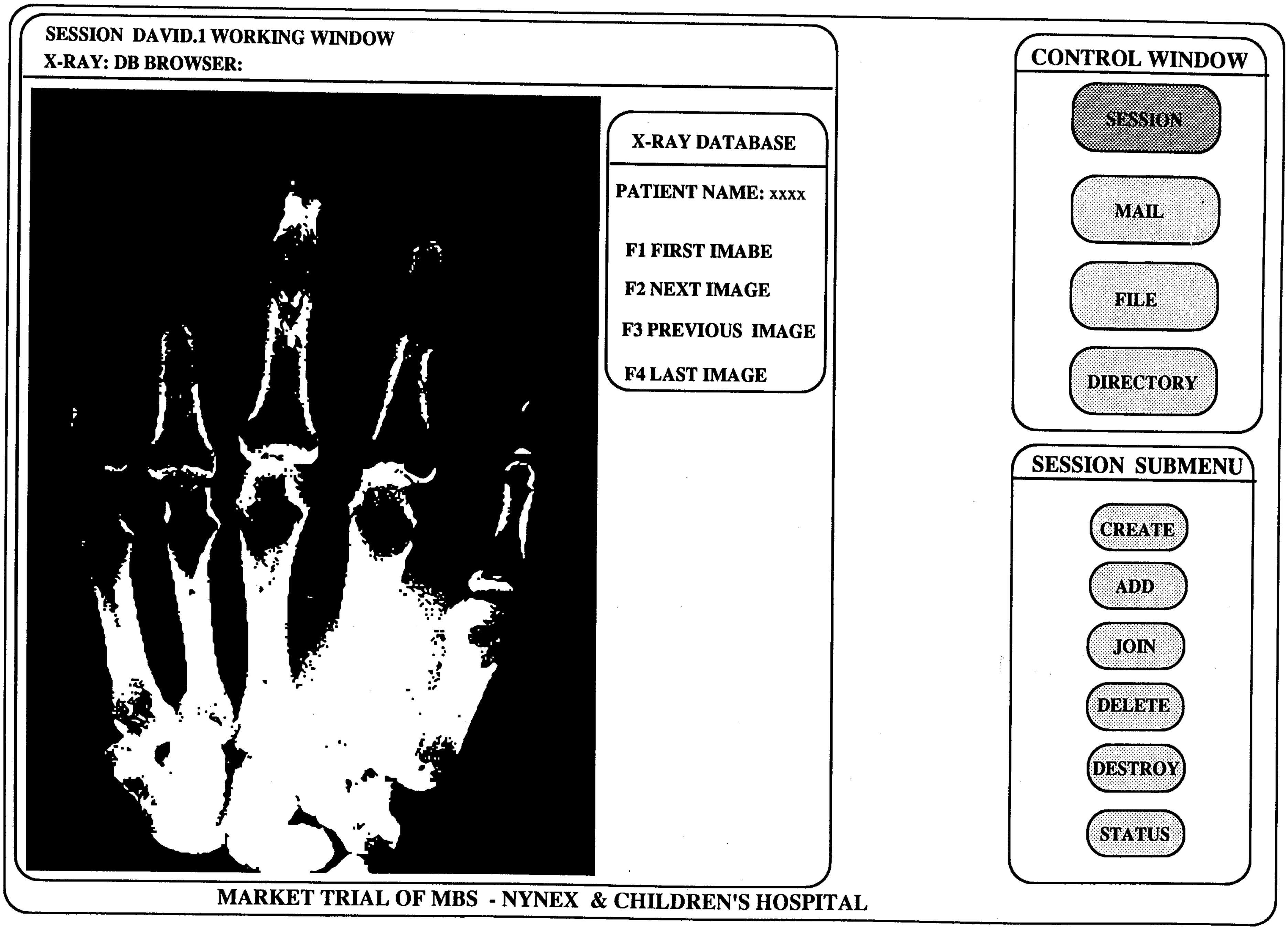
MBS SCREEN WHEN SAVE ICON IS CHOSEN



MBS SCREEN FOR CHOOSING FILE SAVE ICON



MBS SCREEN WHEN A SESSON IS GOING ON WITH DB BROWSER



MBS SCREEN WHEN FILE ICON IS CHOSEN TO SAVE THE SESSION SCREEN

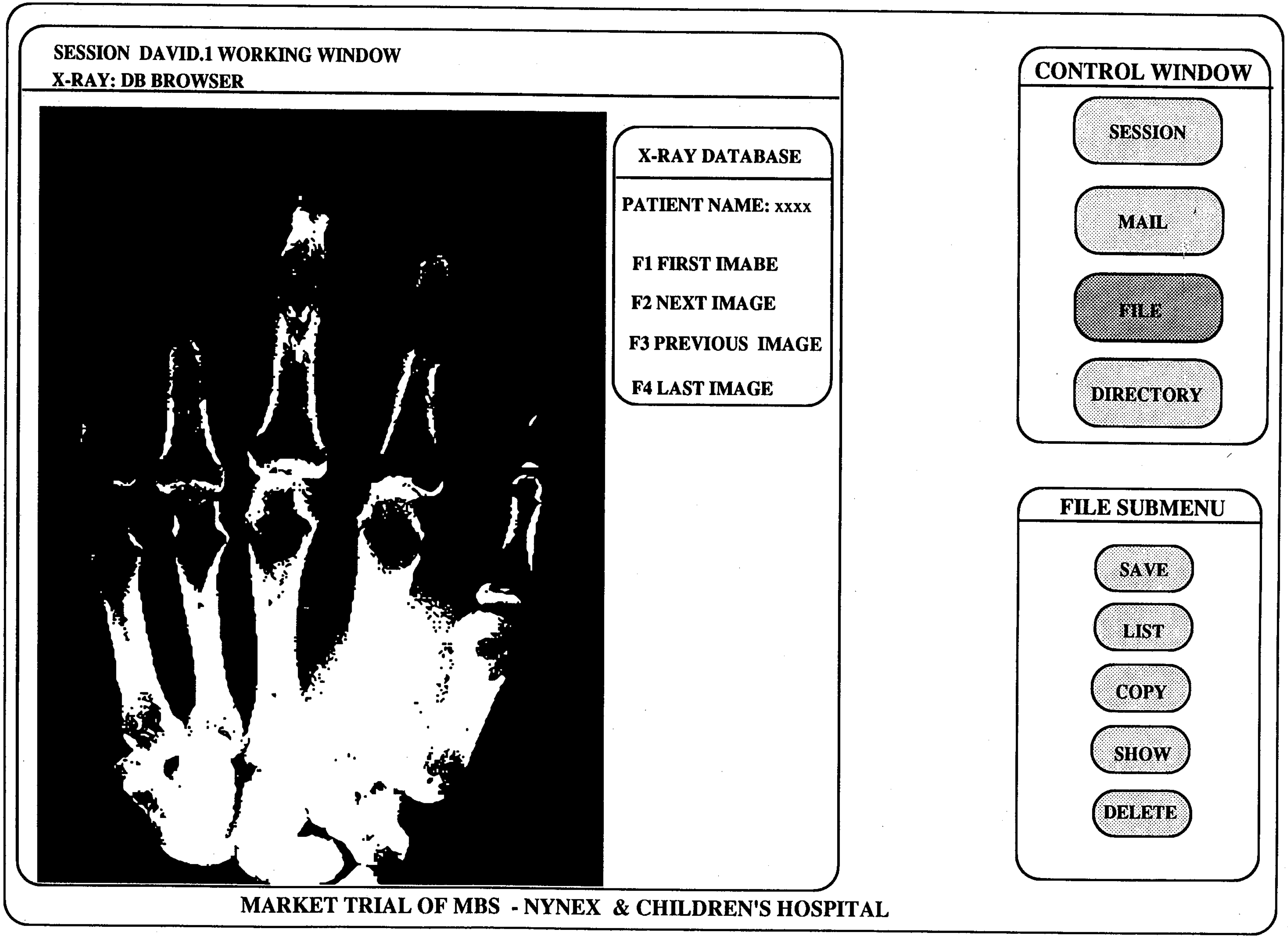


Figure 3.x depicts the state diagram for the file service.

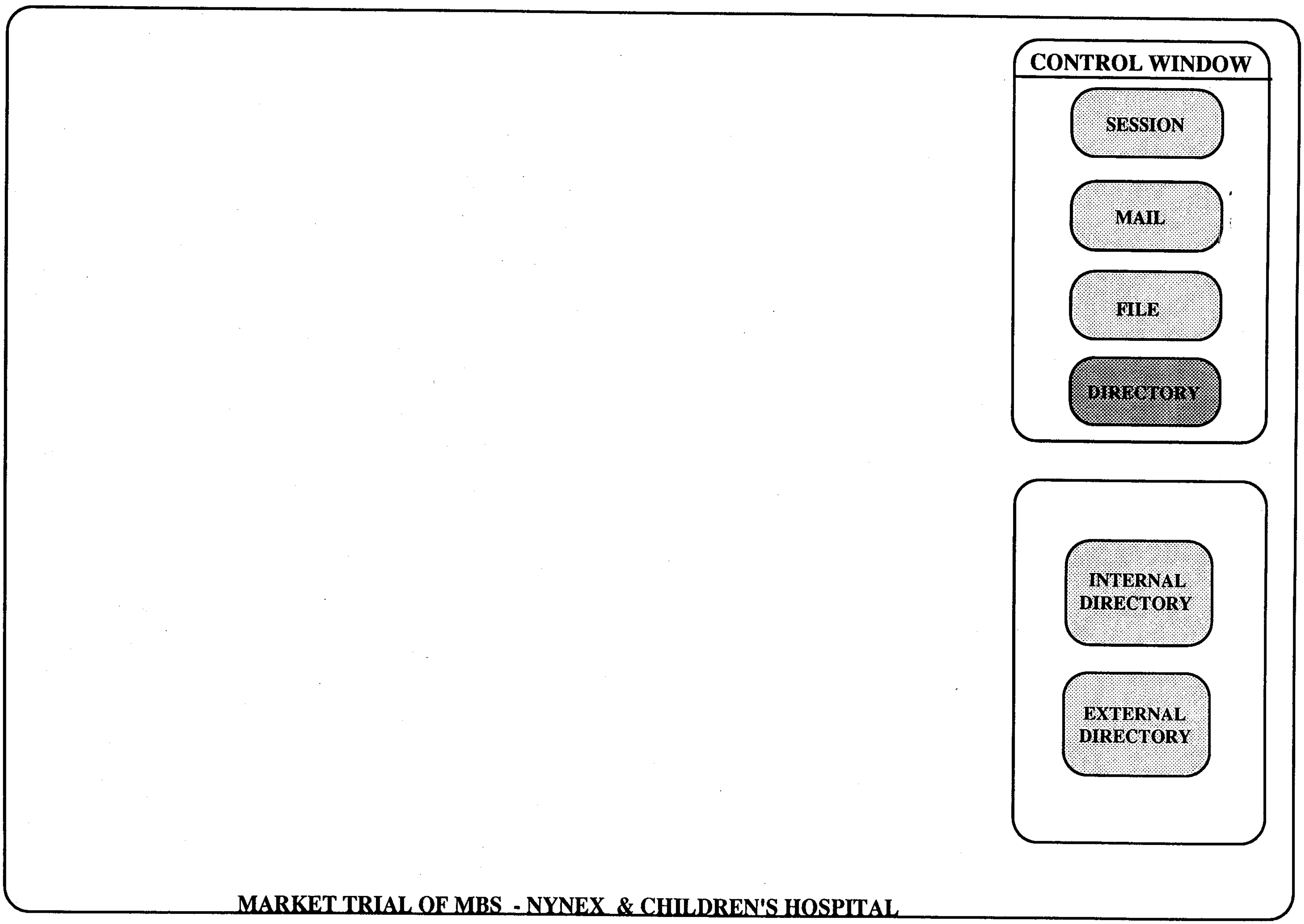
Figure 3.x File State Model

3.2.3.4 Directory

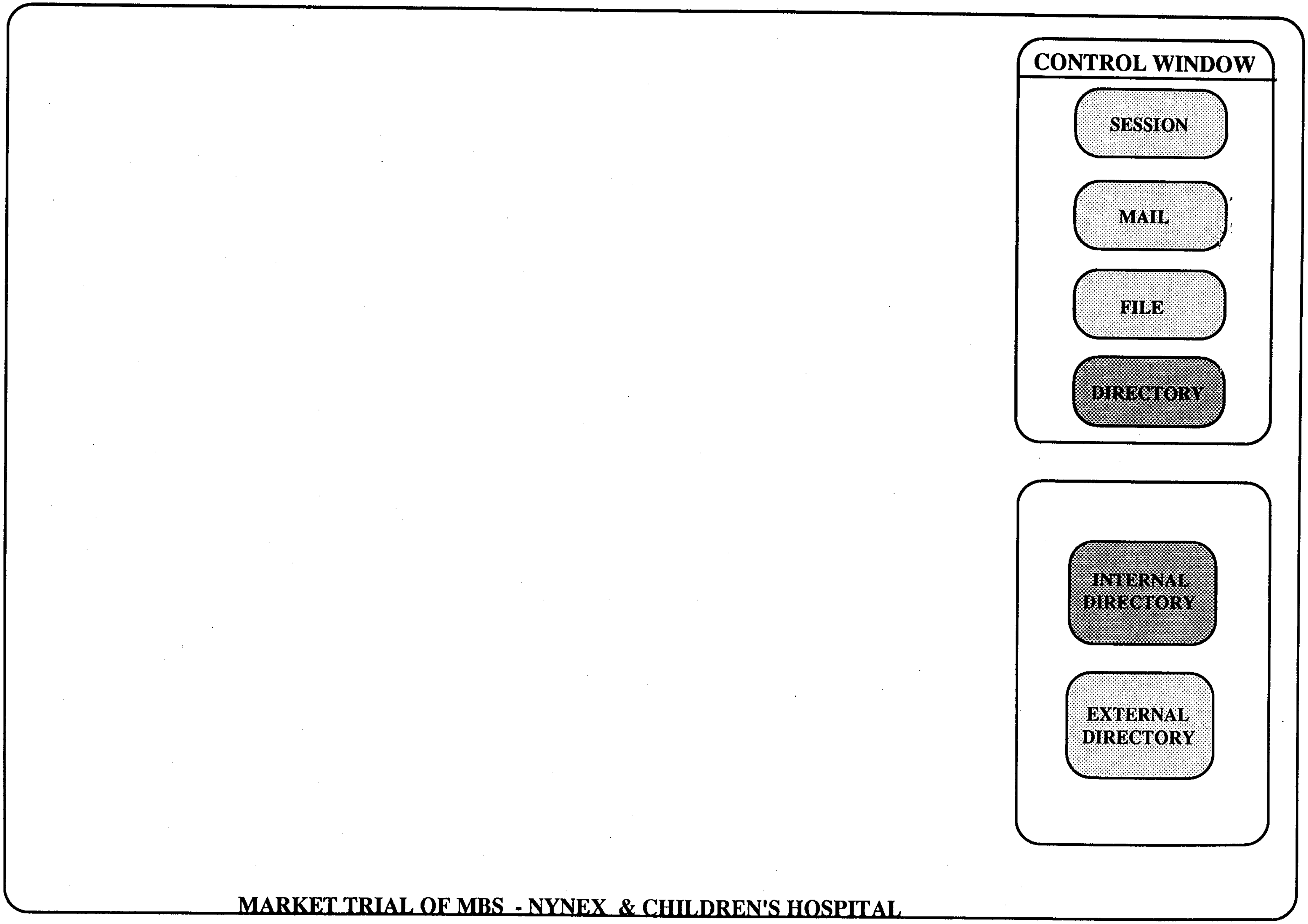
The directory service, as detailed in Figure 3.x is the final service that we shall be developing in this text.

Figure 3.x Directory Screen Layout

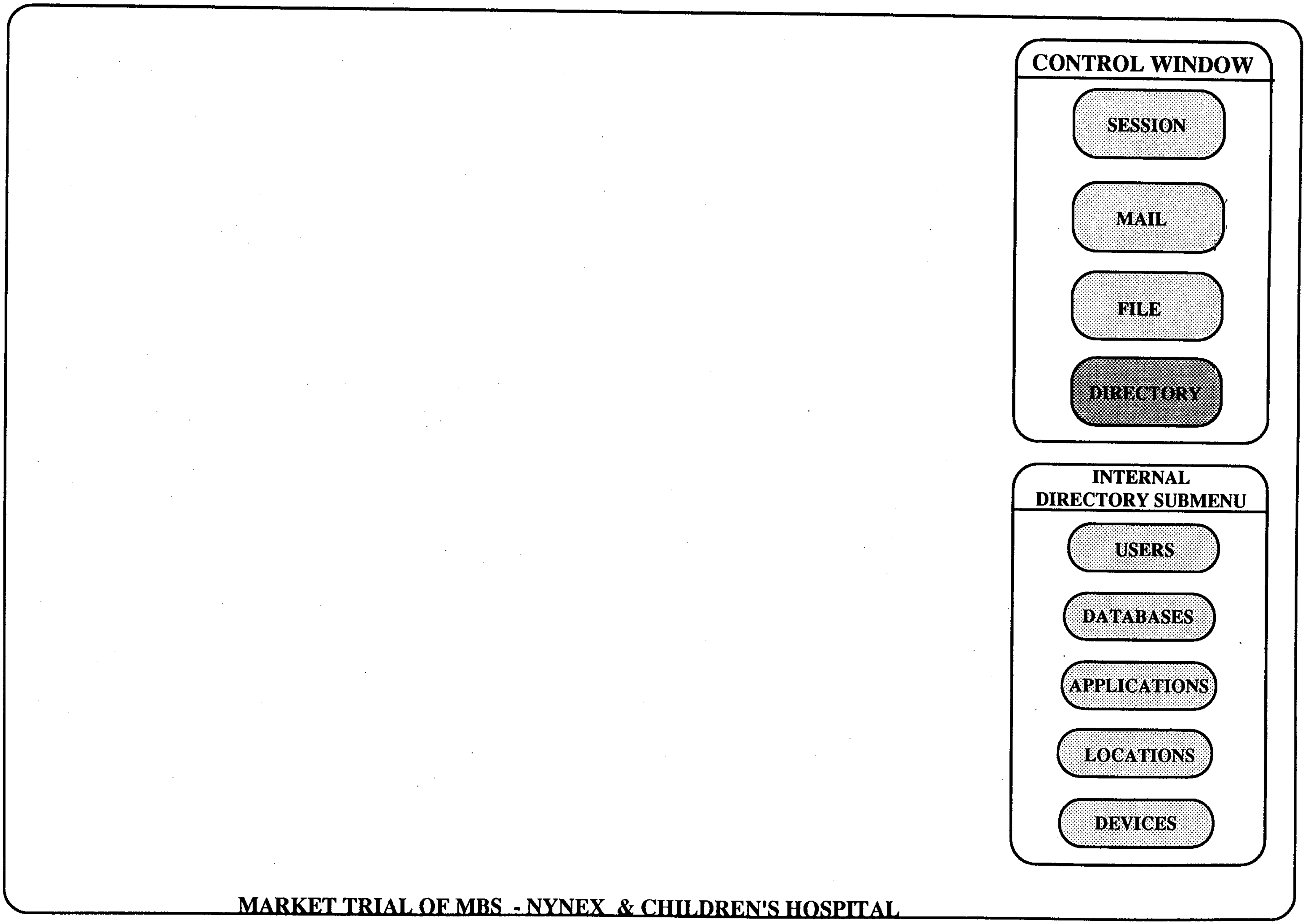
MBS SCREEN WHEN DIRECTORY SERVICE IS SELECTED



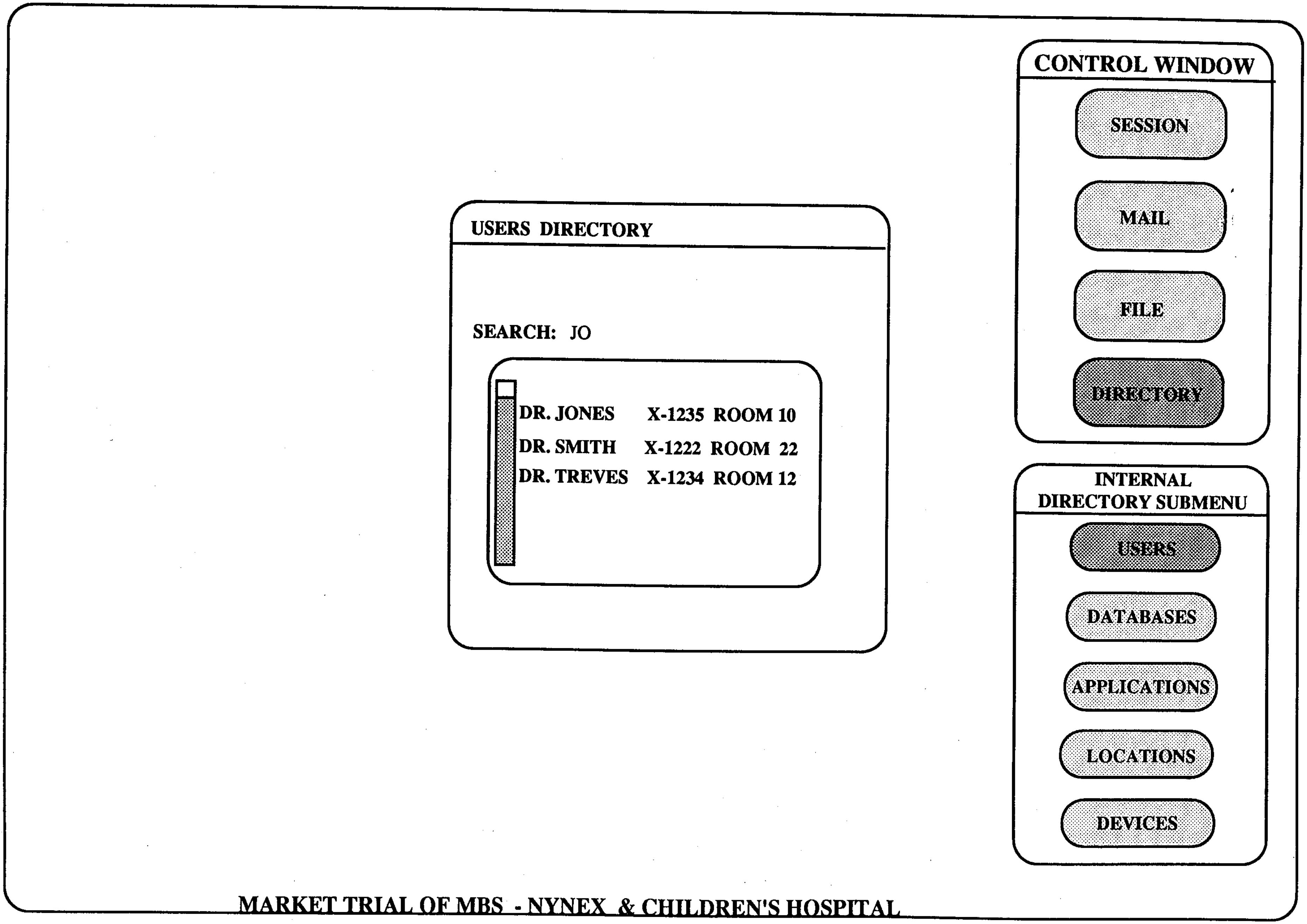
MBS SCREEN WHEN INTERNAL DIRECTORY SERVICE IS SELECTED



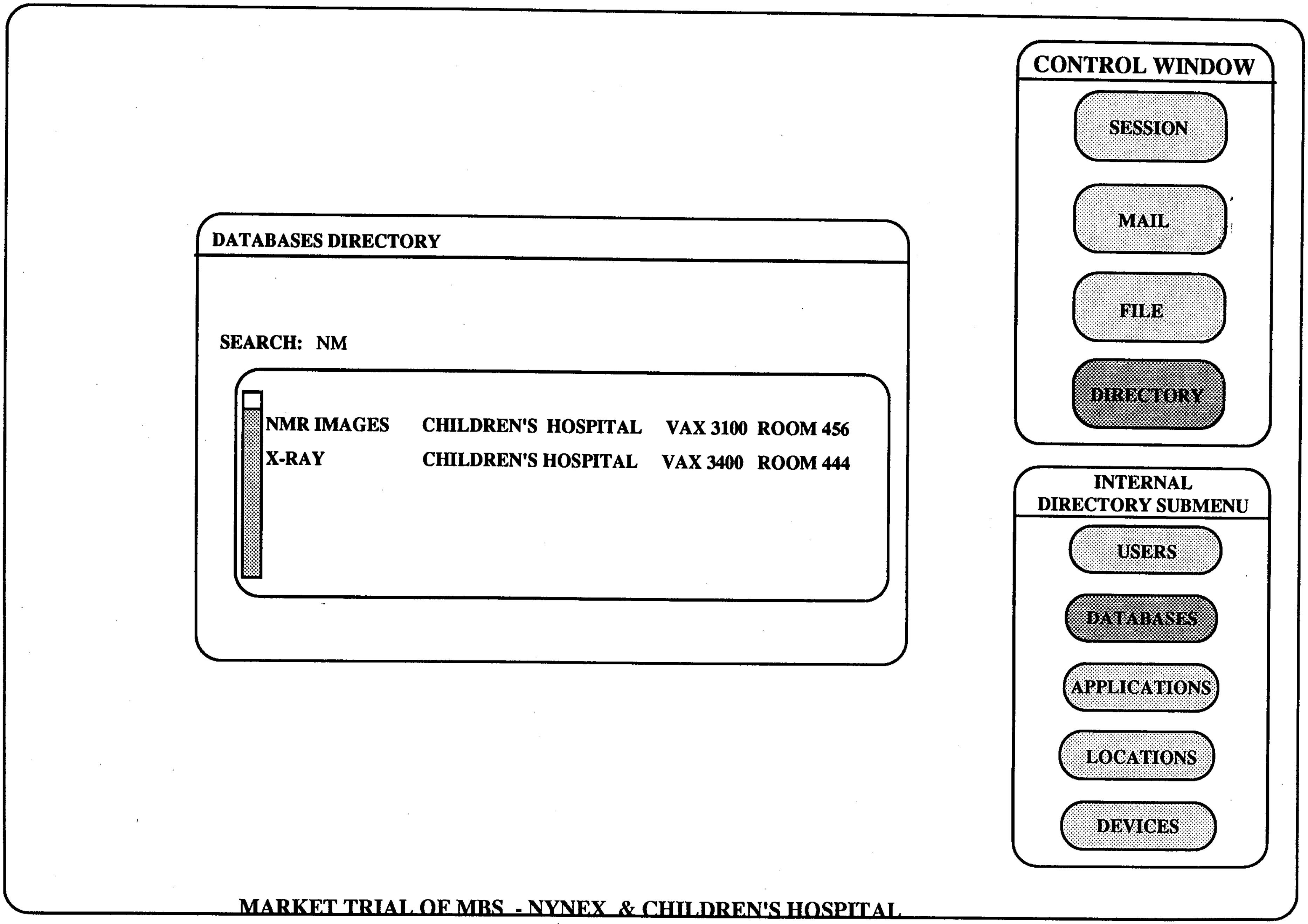
MBS SCREEN WHEN INTERNAL DIRECTORY SERVICE IS SELECTED



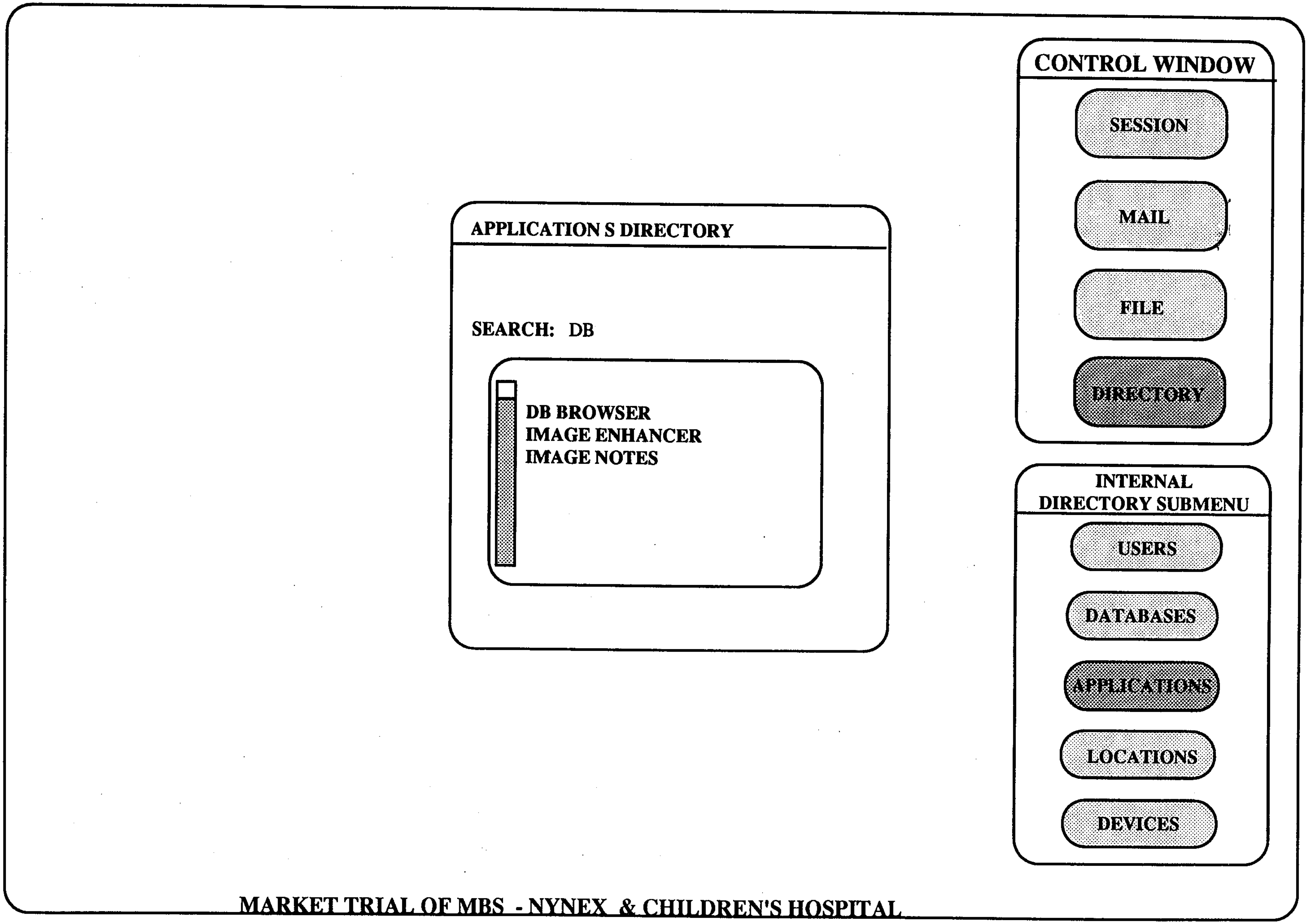
MBS SCREEN WHEN USERS DIRECTORY IS SELECTED



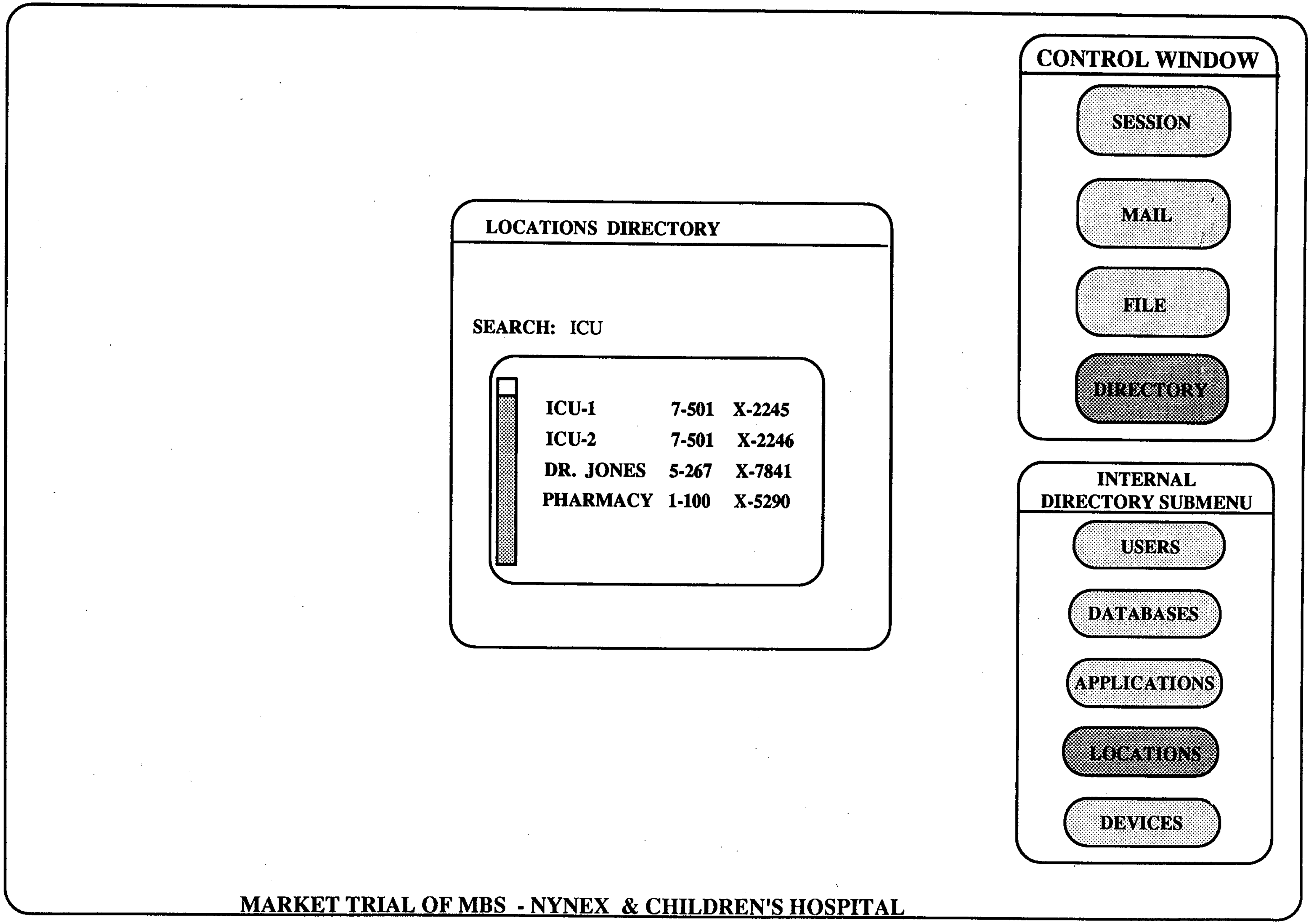
MBS SCREEN WHEN DATABASES DIRECTORY IS SELECTED



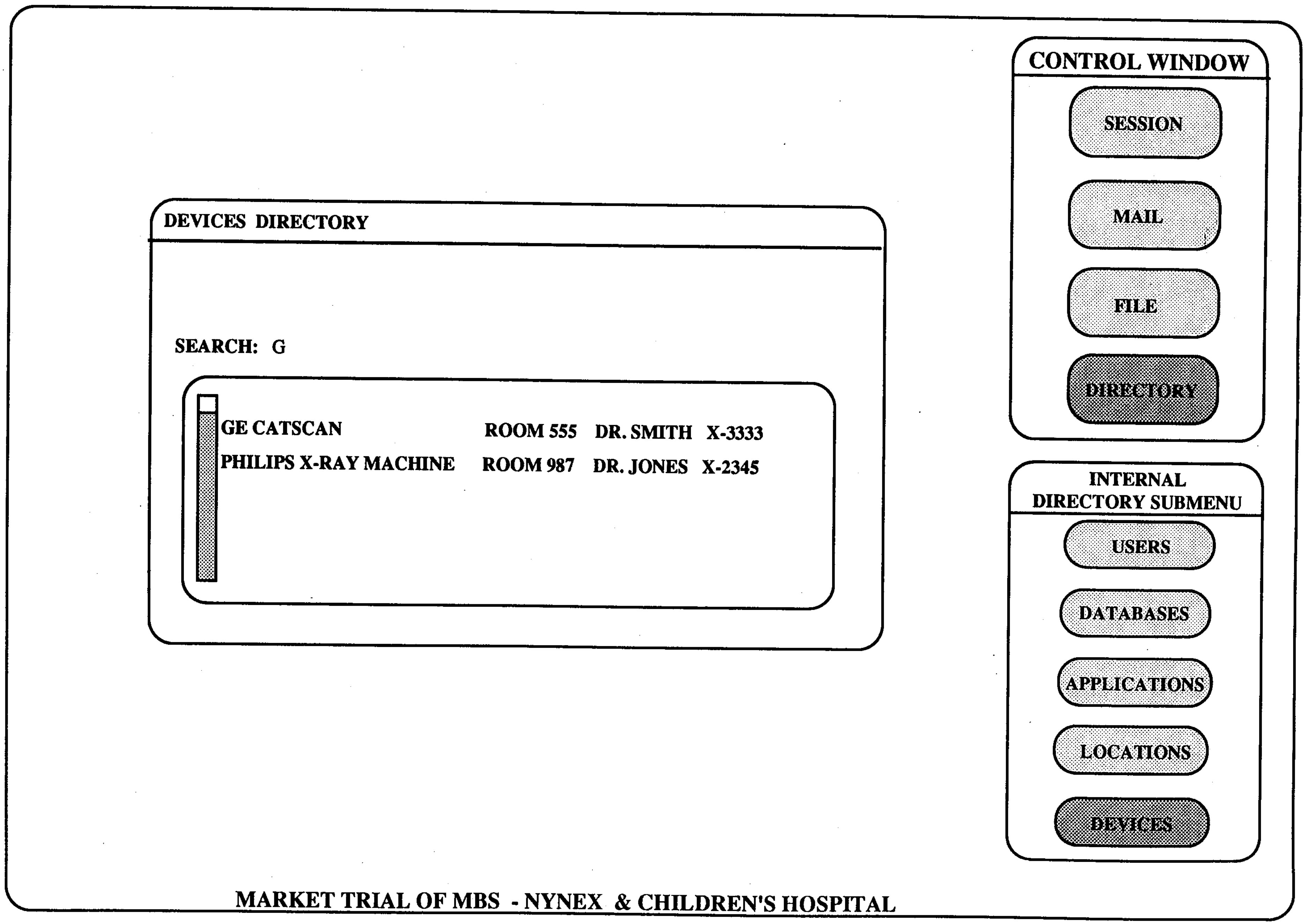
MBS SCREEN WHEN APPLICATIONS DIRECTORY IS SELECTED



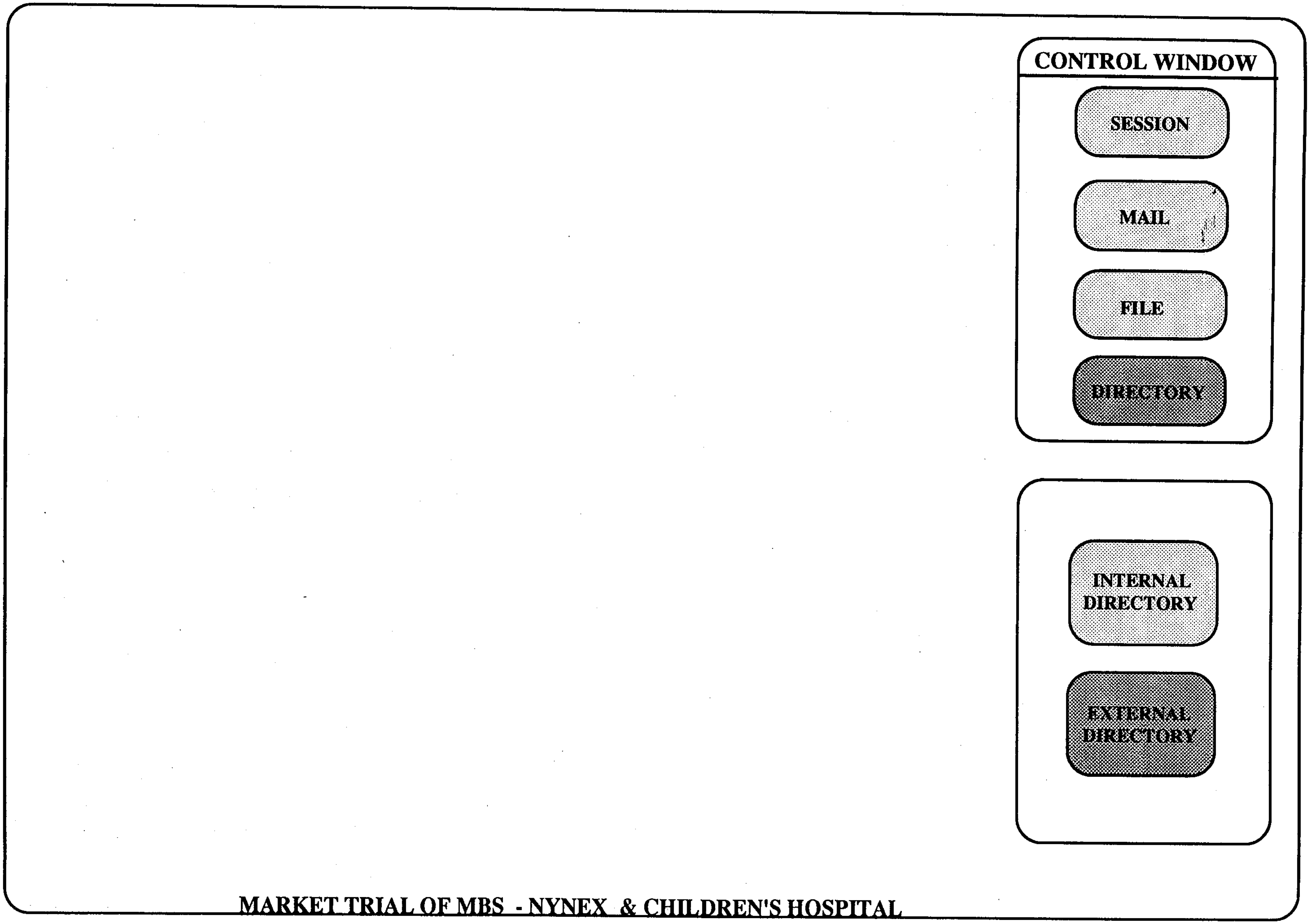
MBS SCREEN WHEN LOCATIONS DIRECTORY IS SELECTED



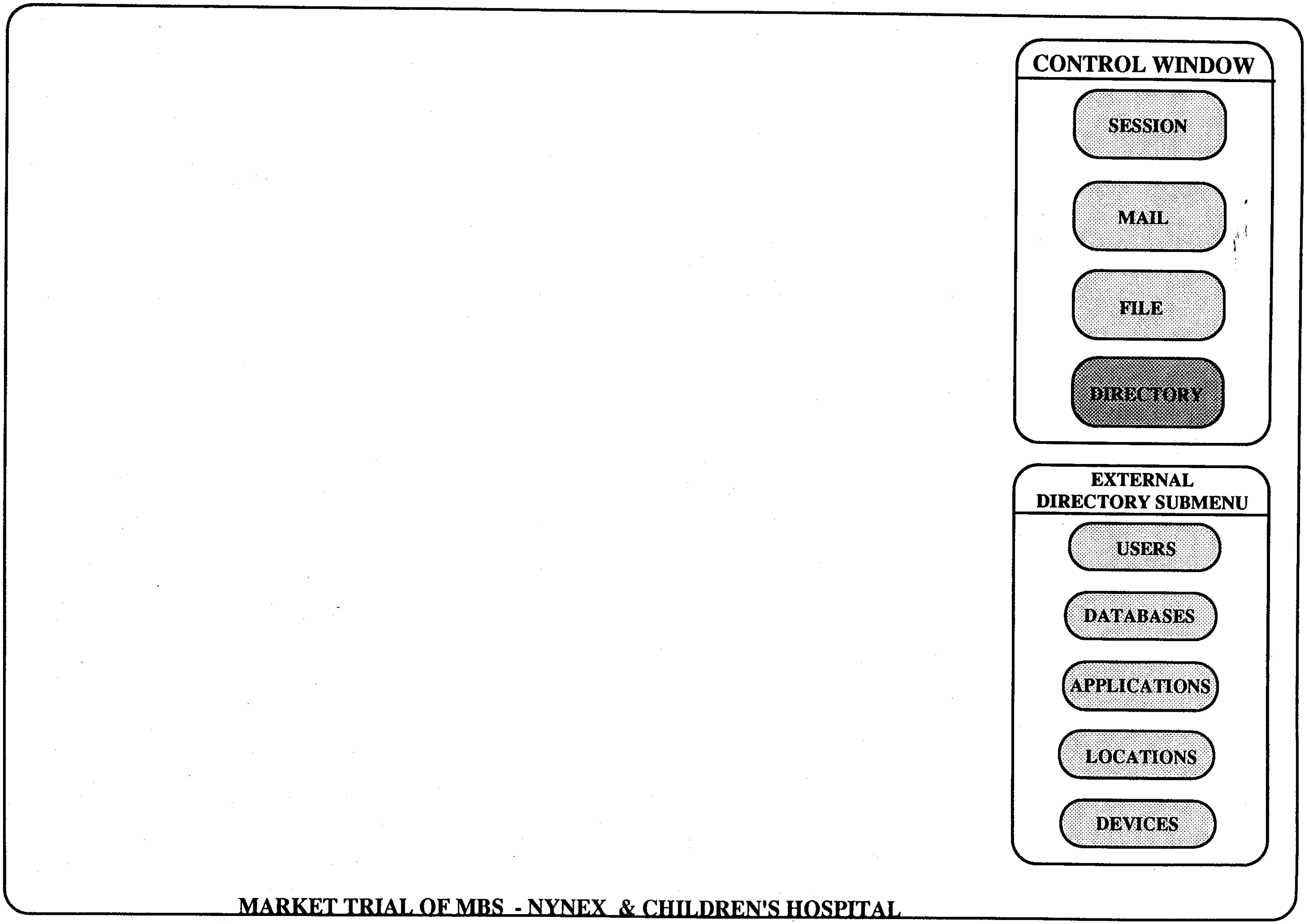
MBS SCREEN WHEN DEVICES DIRECTORY IS SELECTED



MBS SCREEN WHEN EXTERNAL DIRECTORY SERVICE IS SELECTED



MBS SCREEN WHEN EXTERNAL DIRECTORY SERVICE IS SELECTED



The state diagram for the directory service is shown in Figure 3.x.

Figure 3.x Directory State Model

3.3 Interface Modeling

The end user interface is the construct that allows the human to interact with the multimedia environment and to effect changes on that environment. The end user may be in one of several levels of intellectual development. At one extreme the end user may be a highly sophisticated user of the system with significant training in his specialty. At the other extreme is the more casual user who desires to use the system independent of any significant training. The issue of interface modeling is to address the needs of both types of users.

Interface modeling is a methodology used to design and analyze the end user interface using interfaces states. It does not deal directly with the detailed human factors issues of screen layout and colors shapes forma and other factors. All of these others come to play a significant role in the operations of the end user interfaces but they are secondary in the role of modeling the interface in a communications type environment.

3.3.1 Interface State Descriptions

We shall be dealing with the interface in terms of the state diagram of the interface state machine. To develop that methodology we will first define and develop the state description elements. These elements are:

o States

o Presentation

o Inputs

o Outputs

o Process

o Read Data

o Write Data

o Interface

3.3.2 State Machine Analysis

We shall now take the state diagram construct developed in the last section and apply it to several specific applications. Specifically we shall apply it directly to the four services that we have discussed in the last major section; mail session, file and directory. These services are the type that would be of use to the typical end user of the communications service.

In the first model of the system, we shall develop what is called the state dynamic model. This model is depicted in Figure 3.x. In this figure we have shown the following elements:

o Presentation: This represents one screen that presents the information to the end user. The presentation element is the heart of the state dynamics model.

o Input Action: This is the set of input actions that are made in response to the elements that are presented on the presentation screen.

o Read Data Base: This action is one half of what is called a database transaction. It reads a database element that is used as an integral part of the overall end user effort.

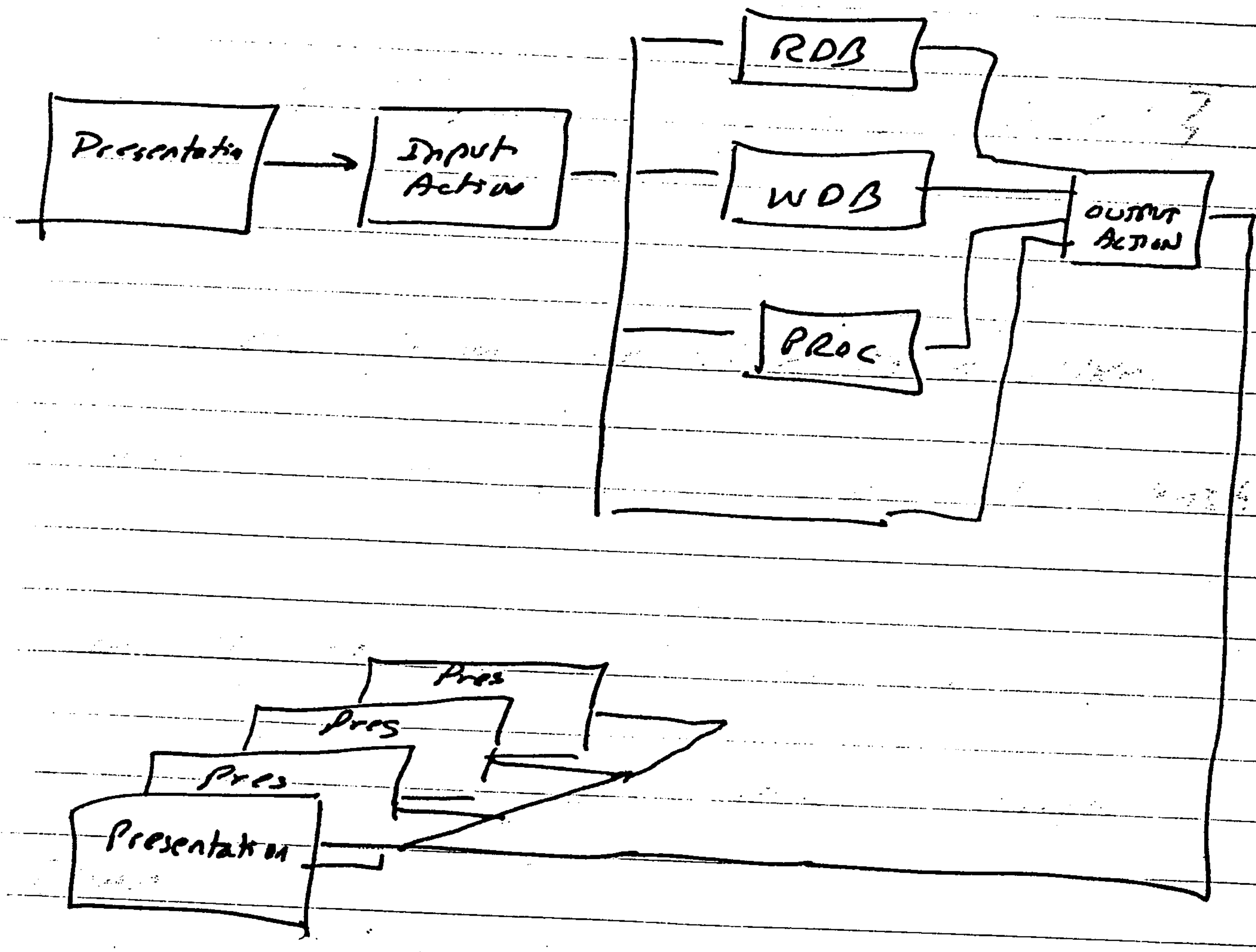
o Write Data Base; This is the second part of the total database transaction and represents the second possible response to and end user input action.

o Process: This is the actions that are taken internal to the system that combines the result from the presentation, that actions of the database transaction and the internal mechanism that are pert of the user interface.

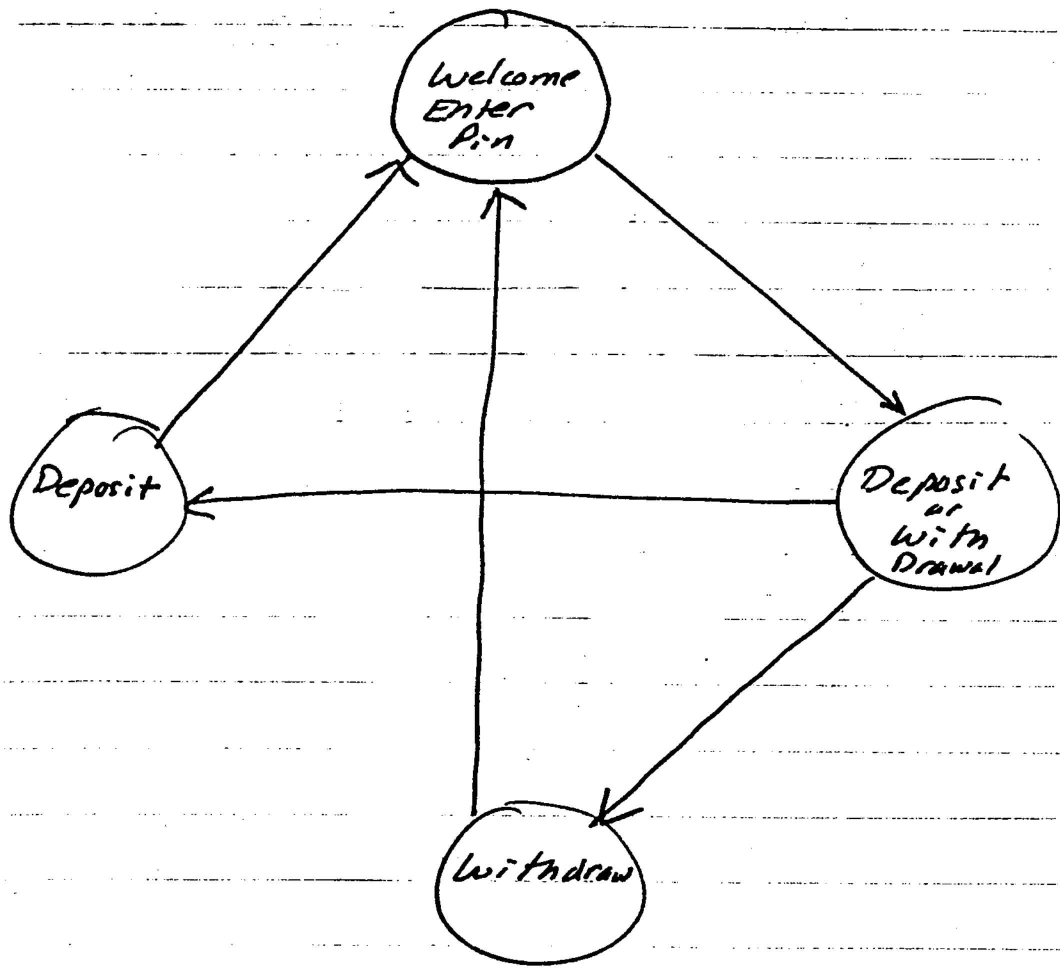
o Output Action: This is the action that results from the process and the database transaction elements or it may even result directly from the input action and the presentation. The output action then leads to movement to another presentation.

Figure 3.x State Dynamics Model

State Dynamics Model



State Transition Using Presentation Formats



Simple ATM example

Consider the example of a simple user interface that is part of the withdrawal of funds from an ATM. This is shown in Figure 3.x. We have four presentation elements or screen that are presented to the end user for s decision to be made. The screens are:

o Screen 1: Indicates that the user must log onto the system with some PIN number or other security code mechanism.

o Screen 2: Asks the user if they desire to deposit money or withdraw funds.

o Screen 3: This is the withdrawal screen and has all the information necessary to withdraw funds.

o Screen 4: This is the deposit screen and it also contains all the necessary information.

Figure 3.x ATM Transaction Using Presentation Formats

We can see in Figure 3.x that the four screens can be viewed as the four states of the system. Movement occurs between the stets. The movement requires that the other elements of the model be evoked, namely the database elements, and the input and output functions, each associated with a particular screen. For example, let us focus on screen 3 which is the withdrawal screen. It has the following elements:

o Input Action: It requires the entering the amount to be withdrawn.

o Process: It calculate the amount in the account and determines the resulting balance to be used for the update of the account database.

o Read Database: This reads the original amount for authorization.

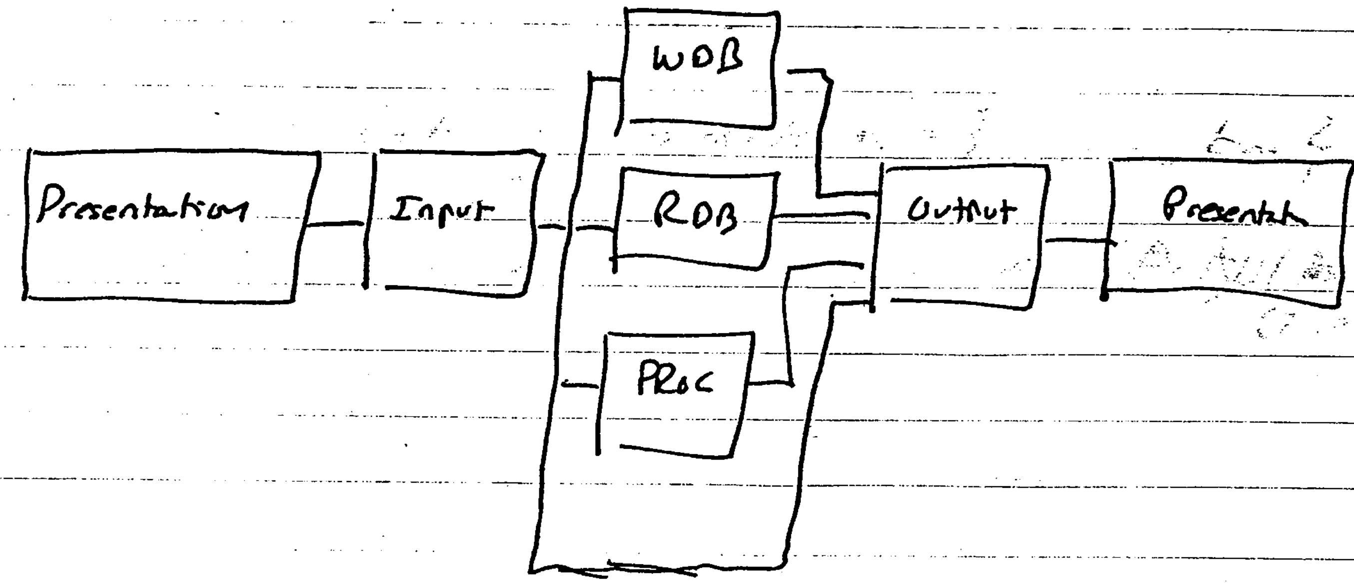
o Write Database : This write to the database the remaining amount.

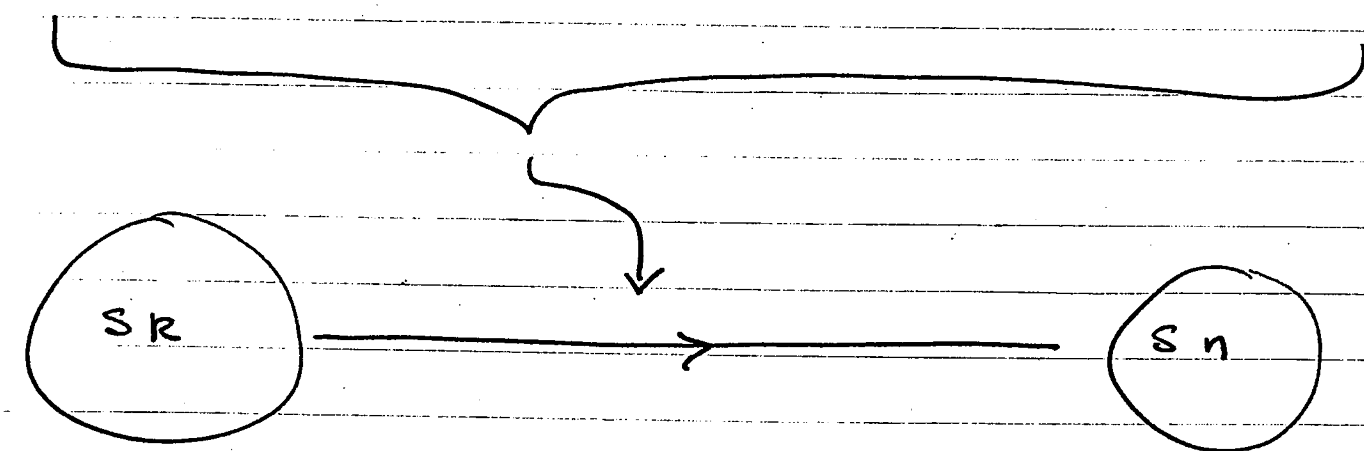
o Output Action: This entails the dispersal of the funds requested. The system then automatically moves on to the next screen.

Using the state dynamics model, we can see that the states are effectively the presentation elements, and that the transitions between are comprised of all,of the other elements as shown in Figure 3.x. In this figure we show the relationship to the state and the transition path. This transition path is a block representing the transition process that we have developed in the state dynamics model.

Figure 3.x State Dynamic Model and Petri Net Element

Using state dynamics model:

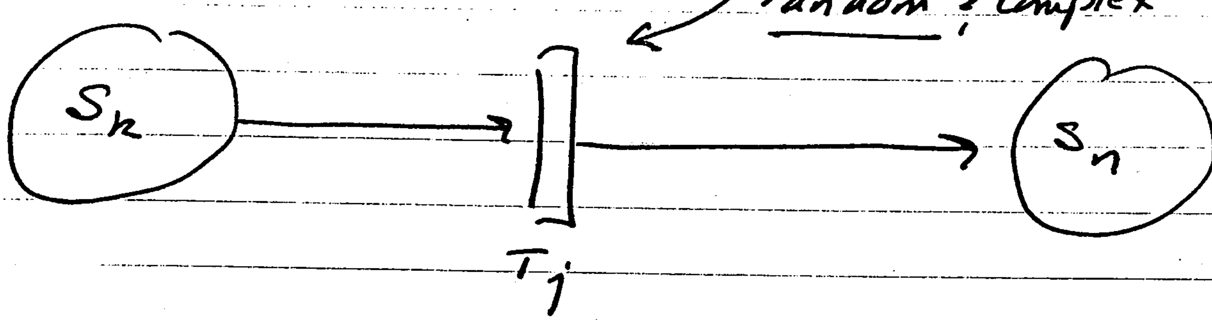




Transition
Path

⇓ Petri Nets

random & complex



In this figure, we show that the basis element are the states and the transitions. These two elements can be combined into a general theoretical structure called the Petri Net. We shall rely upon the work of Marsan et al to develop this theory. This we shall also extend with the work of Stotts and Furata which combines this with the Hypertext model.

We can now begin to define the Petri Net.

Definition: The Petri Net, PN, is a tuple consisting of the following elements:

P = A set of places = {P1,....,Pn}

T = A set of Transitions = {T1,....,Tm}

A = A set of directed arcs; (PXT)X(TXP)

and :

PN = {P,T,A}

In a PN there are two sub sets call pre sets and post set. These are the elements that make up the arcs. We define these as follows:

preT = {P: (P,T)   F}

and

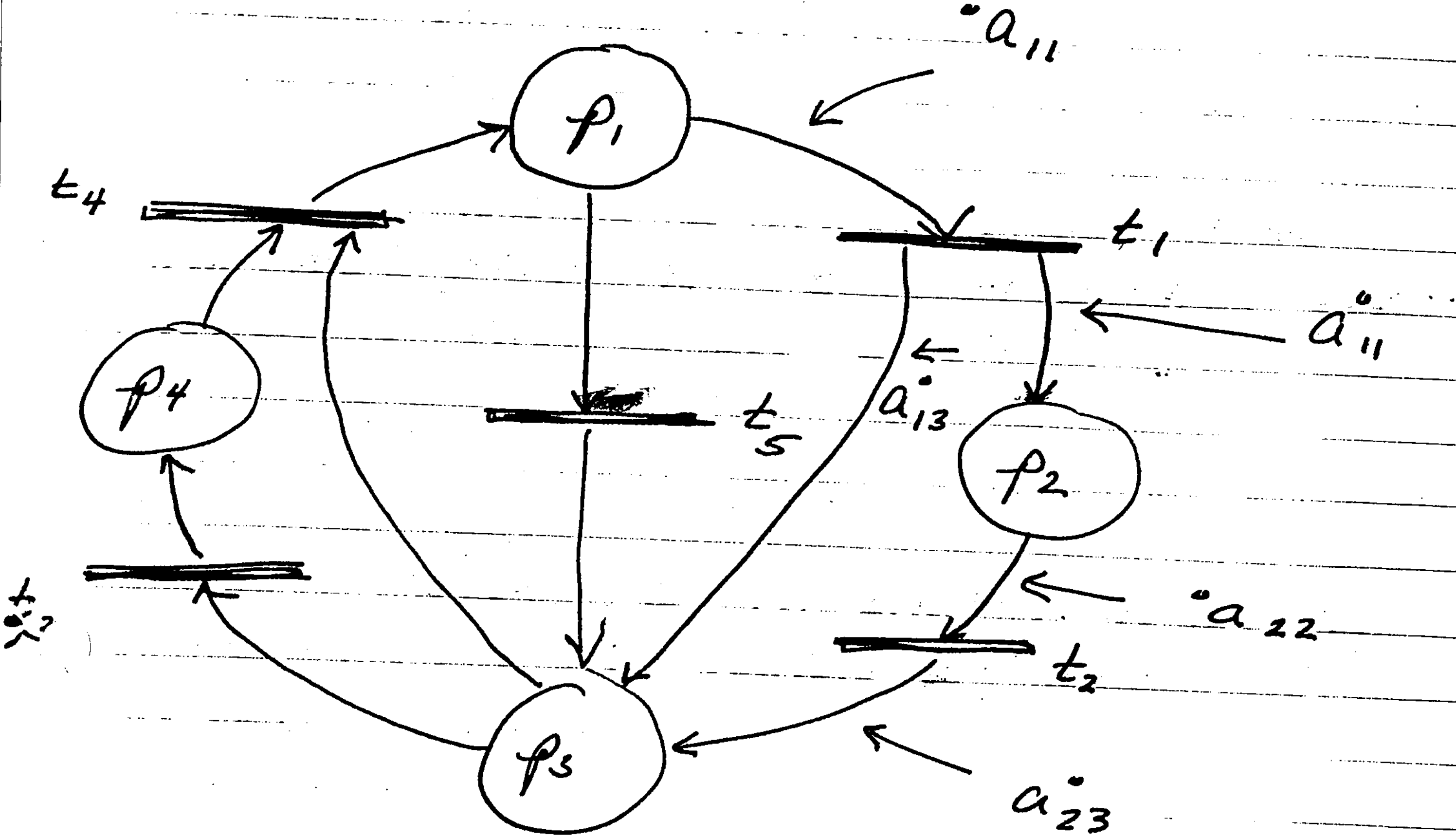
postT = {P: (P,T)   F}

where F is the set of all acceptable P,T pairs.

Consider a sample Petri Net as shown in Figure 3.x.

Figure 3.x Sample Petri Net Example

Petri Nets



$^\bullet a_{11}$
p_1
t_4
t_1
a_{11}°
p_4
t_5
$a_{13}^\bullet$
p_2
t_3
$^\bullet a_{22}$
t_2
p_3
$a_{23}^\bullet$

In this example there are four places, similar to states or presentations in the original model description. There are five transitions. These transitions represent the accumulation of all the steps that are required to be accomplished to get from one place to another. The transitions control the movement from one place to another, whereas the places are less active players in the PN model.

We have also identified the pre and post sets by the use of the terms aij, where we use a pre and post notation attached.

We can now introduce the concept of a marking, which is a way in which movement can bet determined around the PN> A marking is simple a set of integers that are assigned to a set of places, and an algorithm that details how those integers are propagated around the network. Let us define a marking as follows;

Definition: A marking is an n tuple, {b1,.....,bn}, where bk is a binary integer, 0,1, and;

M = {b1,....,bn}

where;

M: P-> {b1,....,bn}

Further we define M0 as the initial marking and Mf the final marking.

Firing of a PN consists of taking one token in each preT and adding it to the token in postT.

we can now further define the actual execution algorithm for a PN.

Definition: The execution algorithm for a PN consists of the following steps:

(i) Any Ti is enabled when ALL of its input places contain one or more tokens.

(ii) A transition Ti, that is enabled, can fire and when it fires it removes one token from each input Pk and places a single token in each output Pj.

(iii) Once the firing occurs, the tokens are repositioned and a new marking occurs. The marking sequence may be given by M(0), M(1), ..., M(n),...

Figure 3.x depicts a example of a more detailed ATM case, wherein we have added a total of eight states. We start with a single token in the start place. Figure 3.x depicts the transition of tokens. It should be noted that there are two non-determinitsic token transitions that can occur. These are at the deposit/withdraw place and at the Another Transaction? place. In this case we shall leave them deterministic. In latter parts of this chapter we shall sow how we handle this probabilistically.

Figure 3.x ATM Token Example

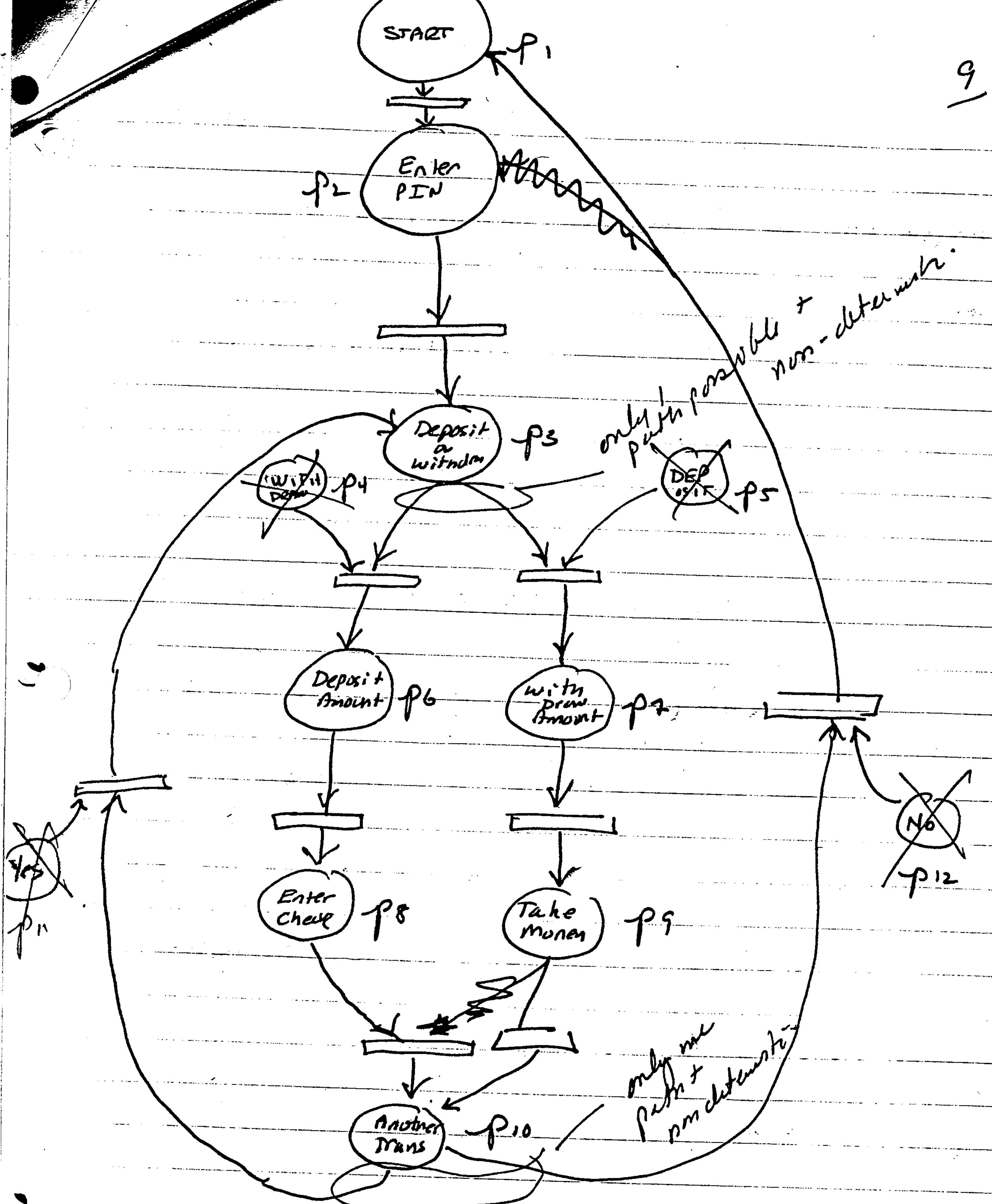

START
P1
Enter PIN
P2
Deposit a withdraw
P3
WITH DRAW
P4
DEPOSIT
P5
only 1 path possible + non-determinist.
Deposit Amount
P6
with draw Amount
P7
Enter Cheque
P8
Take Money
P9
Another Trans
P10
Yes
P11
No
P12
only one path + non determinist

Figure 3.x ATM Token State Evolution

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12
M_o	1	0	0	1	0	0	0	0	0	0	0	1
	0	1	0	1	0	0	0	0	0	0	0	1
	0	0	1	1	0	0	0	0	0	0	0	1
	0	0	0	0	0	1	0	0	0	0	0	1
	0	0	0	0	0	0	0	1	0	0	0	1
	0	0	0	0	0	0	0	0	0	1	0	1
	1	0	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0

Unless have M_o n equivalent.

In the transition of these states, we see that M(0) is the initial state and that initial state may be anything that we wish to start with. From an initial M(0), we generate the sequence of markings, M(1),....,M(n),... These are generated from a generic state transition function;

$$M(1) = G(1,0)\ M(0)$$

or in general we have;

$$M(k+1) = G(k+1,k)\ M(k)$$

The transition function can be calculated based on the transition elements and the arcs. We leave this detail to the problems at the end of the chapter.

Having defined the concept of the PN we can now extend this to the concept of a Hypertext. This is developed in the context of the work of Stotts and Furuta. It is as follows;

Definition: A hypertext is an n-tuple that is as follows;

$$H = \{PN, C, W, B, Pl, Pd\}$$

where;

PN = any Petri Net

C = a set of document contents. This may include any set of text, graphics etc that may make up the hypertext document.

W = a set of windows. These windows may be ordered or otherwise.

B = a set of buttons. These are any actions that may create a response on the system.

Pl = a logical projection of the document.

Pd = a display projection of the document.

Using the concept of the PN Hypertext we can determine several important solutions to problems concerning Hypertext. These problems are:

o Display Complexity: Using the PN graph and the Hypertext adjunct, it is possible to determine the number of hypertext windows that are simultaneously need to display the information. This is the number of marked places in the PN associated with the Hypertext.

o Path Size and Synchronization: Using the PN formulation, we can determine the length of the path needed and the level of concurrent path synchronization.

o Reachability of Places: This is performed in developing the set R(M(0)), the reachability states of the initial state.

There are several extensions to the Petri Net concepts that we shall develop. These will become essential as we develop the model for the sizing of the source characteristics.

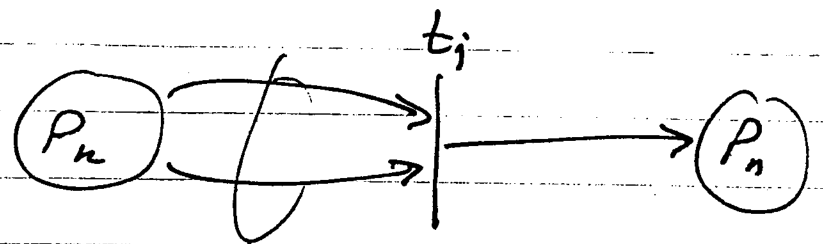
Definition: A PN place Pn has multiple arcs as shown if Figure 3.x. The threshold Tj associated with Pk fires if and only if there are two tokens in Pk. In addition, only one token is passed onto Pn.

Figure 3.x PN Multiple Arcs

The display projection Pd is Collection associated with physical screen projections.

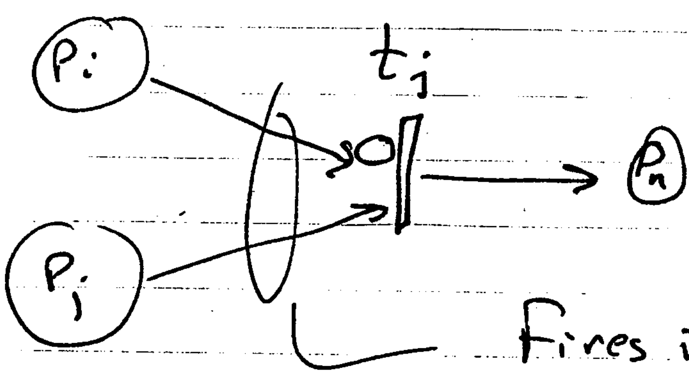
PN Extensions

1. Multiple Arcs



Requires 2 tokens to fire

2. Inhibitor Arcs



Fires if all contain token but the inhibitor node.

Definition: An arc is called an inhibited arc, if and only if all the arcs contain markings, except for the inhibitor arc. This example is shown in Figure 3.x.

Figure 3.x Inhibitor Arcs

Definition: If we let R(M0) be the reachability set of PN, with initial state M0, then the dead markings are those sets in R(M0) that go nowhere. Namely;

M(DS) is a dead set iff;

$$MDS(k+1) = G(k+1,k)\ MDS(k)$$

Definition: PN is called a safe net if the number of tokens <= 1 for all Pj and for all R(M0). We further define PN as STRICTLY CONSERVATIVE (SC) if for all M in R(M(0));

o sum of tokens is constant

If PN is SC, then if C(x) is the cardinality of the set x, we have:

$$\sum_{i=1}^{n} C(preA) = \sum_{i=1}^{n} C(postA)$$

We now introduce two new concepts. The first is the timed PN and the second is the stochastic PN. The timed PN introduces the concept of timing for the PN concept. The stochastic PN allows for the generation of PN transitions that are stochastic in nature. Specifically we have introduced the concept of random timing that is the basic element in determining the ultimate source characteristics.

Definition: A timed Petri Net, TPN, is an n-tuple:

TPN = {P,T,A,M(0),D}

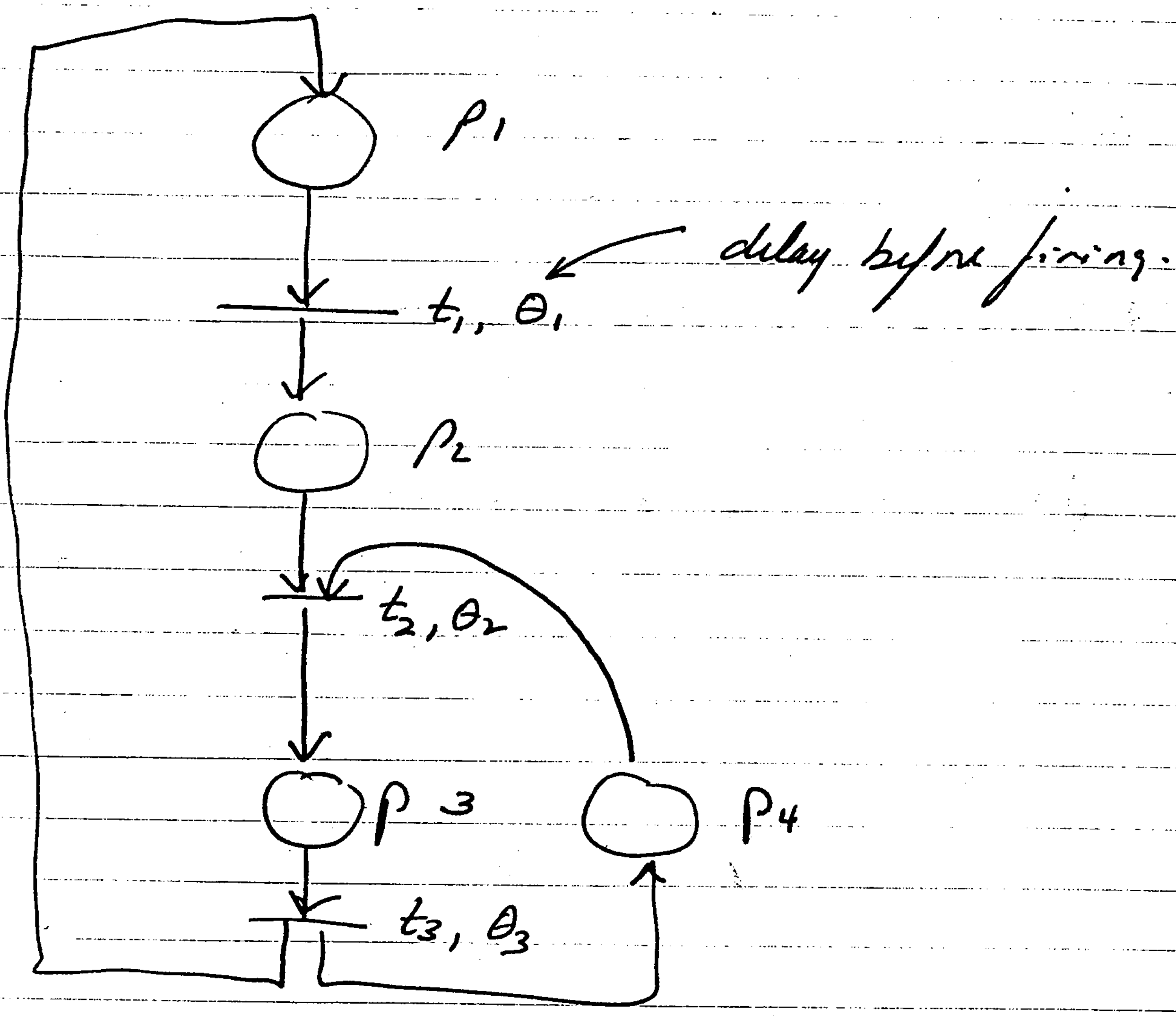
where D is a set of delays at each of the transitions before firing. Specifically, if we look at the example in Figure 3.x, we see that at transition Tk, we have a delay Dk that is in place before the transition fires.

Figure 3.x Timed Petri Net

Timed Petri Nets:

$TPN = \{ P, T, A, M_0, \Theta \}$

$\Theta = \{ \theta_1 \ldots \theta_n \}$ set of delays $\bar{c}$ PN



Let us expand on the concept of the stochastic PN. In this case we expand on the TPN by applying the random variable to the timing at the transitions.

Definition: A stochastic Petri Net, SPN, is an n tuple that is:

$$SPN = \{P, T, A, M(0), L\}$$

where L is a set of random delays at the individual transitions.

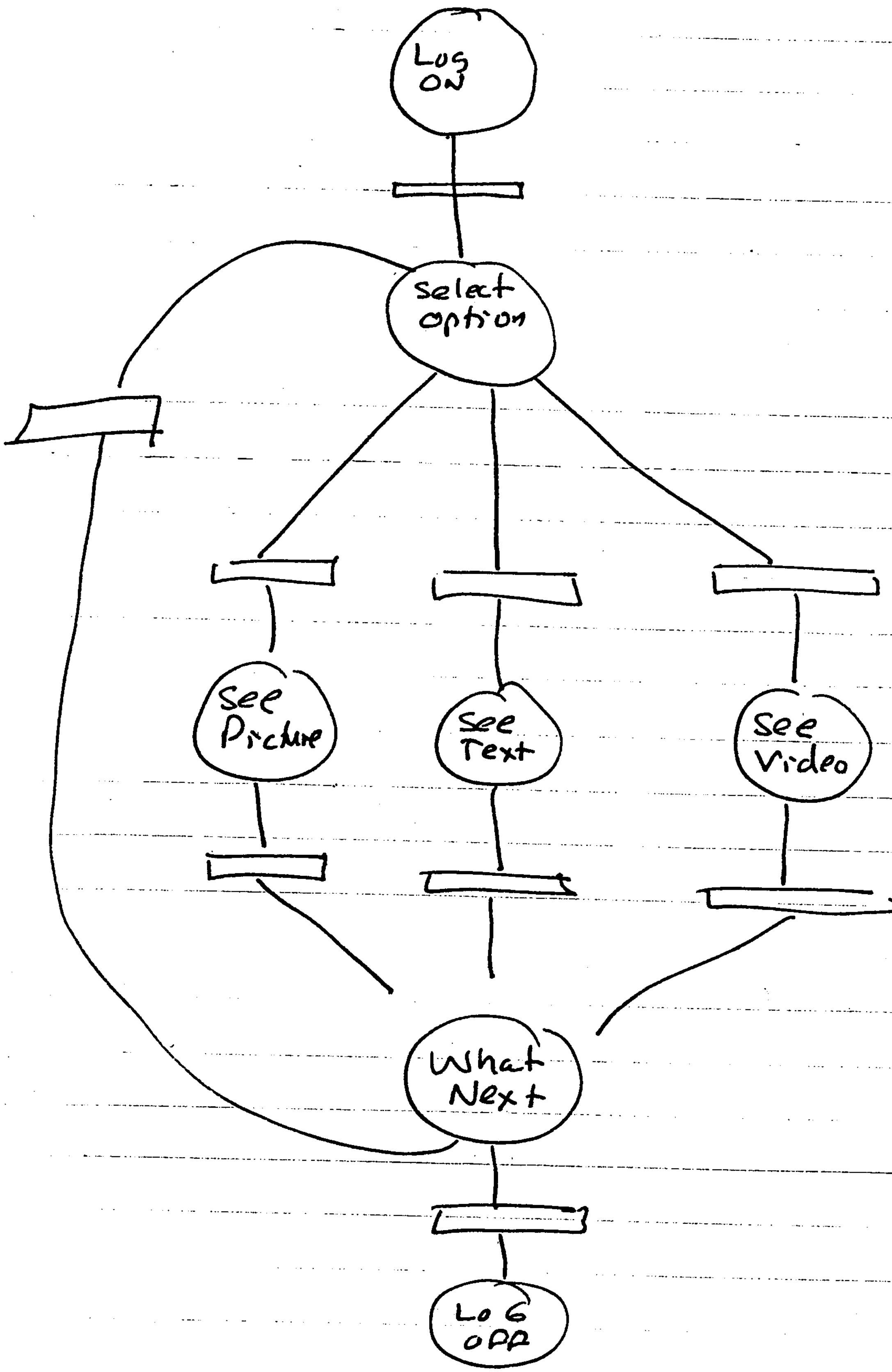
We can then define the probabilities associated with the system going from one state to another, or even the probability of being in any state. In particular it can be shown that the SPN is isomorphic to a markov chain.

Finally, we introduce the structure that we begin this section with. Specifically, we introduce the concept of random transition path selection, as well as random delays at transitions. We also introduce the input and output of data at the time of each transition. We first introduce a simple example to develop this concept.

Example: Consider the sample system that depicts pictures, text, or video. At two nodes we have options that could occur, one allow for the choice of the three images, the second is to decide if the next step is logging off or of starting over again. This is shown in Figure 3.x.

Figure 3.x Image Selection Process: SPN

Source Characterization



What we see in this SPN is that there are non-deterministic choices for these two transition nodes. To alleviate these non-deterministic problems, we introduce an random selection choice at each node. Thus we can define the probability of choosing video as q1 and that of text as q2 and that of video as q3. We note that the sum of the q's is unity as expected.

In addition, when we decide to see a picture, when the transition is activated, it generates an output that is sent to another similar machine. The output is a packet of information of certain length. In a similar fashion, there is an input packet that contains the information necessary to regenerate the picture requested.

Definition: A Source Dynamic Model (SDM) is an n-tuple that is characterized as follows:

$$SDM = \{P,T,A,M(0),L,Q,I,O\}$$

where the first part of the SDM is a SPN, and;

Q are the transition node probabilities

I are the inputs that occur at each transition when activated

O are the outputs that occur at each transition when activated.

We shall use this model for the development of the source characterization.

3.3.3 Performance Analysis

The modeling of the multimedia source will require the use of probabilistic techniques that are found in many other areas. In particular we shall show that the end user interface state diagram ca n be modeled effectively using the finite state markov state machine model and that statistics on the overall state averages can be readily drawn form the technique.

We shall be focusing on the development of performance issues as well with the development of the source model. These typical performance issues are :

- o Response Time
- o Average Holding Time
- o Number of Transitions per Unit time
- o Stability of the state protocol
- o Source generation rate

3.4 MM Source Modeling

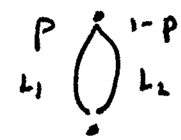
Multimedia source modeling combines the factors developed in the interface model and combines them with the model development that we performed in the last chapter. We can therefore takes a session interaction at one user, and develop the types of input to a communications network that would be anticipated from such as system source. We cane then further expand this source to include a collection of sources to encompass a complete array of users that can provide the cumulative load on the total multimedia communications network.

The source modeling will include the combining of the elements developed in the system state diagrams with the additional factor of image sizing and packaging within the dialogue of the end user session.

3.4.1 Model Elements

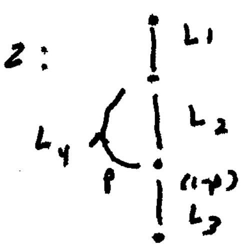
A typical source model is shown in Figure 3.x. It contains the elements that we have developed in the SDM system. It is based on a standard Petri Net models and we have introduced the timed transitions and the stochastic selection of transitions at the places indicated.

Theorem 1:



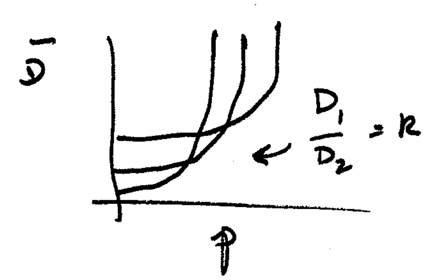
$\Rightarrow$ $L = pL_1 + (1-p)L_2 = \sum p_i L_i$

Theorem 2:



$L_1 + L_2 + L_3(1-p)$
$+ p(L_4+L_2) + p L_3(1-p)$
$+ p^2(2(L_4+L_2)) + p^2 L_3(1-p)$
$+ \cdots$
$+ np^n(L_2+L_4) + p^n L_3(1-p)$
$+ \cdots$
$= L_1 + L_2 + L_3(1-p) + (L_4+L_2)p + \cdots + np^n + \cdots)$
$= \underbrace{L_1 + L_2 + L_3}_{D_1} + \underbrace{\left(\frac{p}{1-p}\right)(L_2+L_4)}_{D_2}$

Discuss

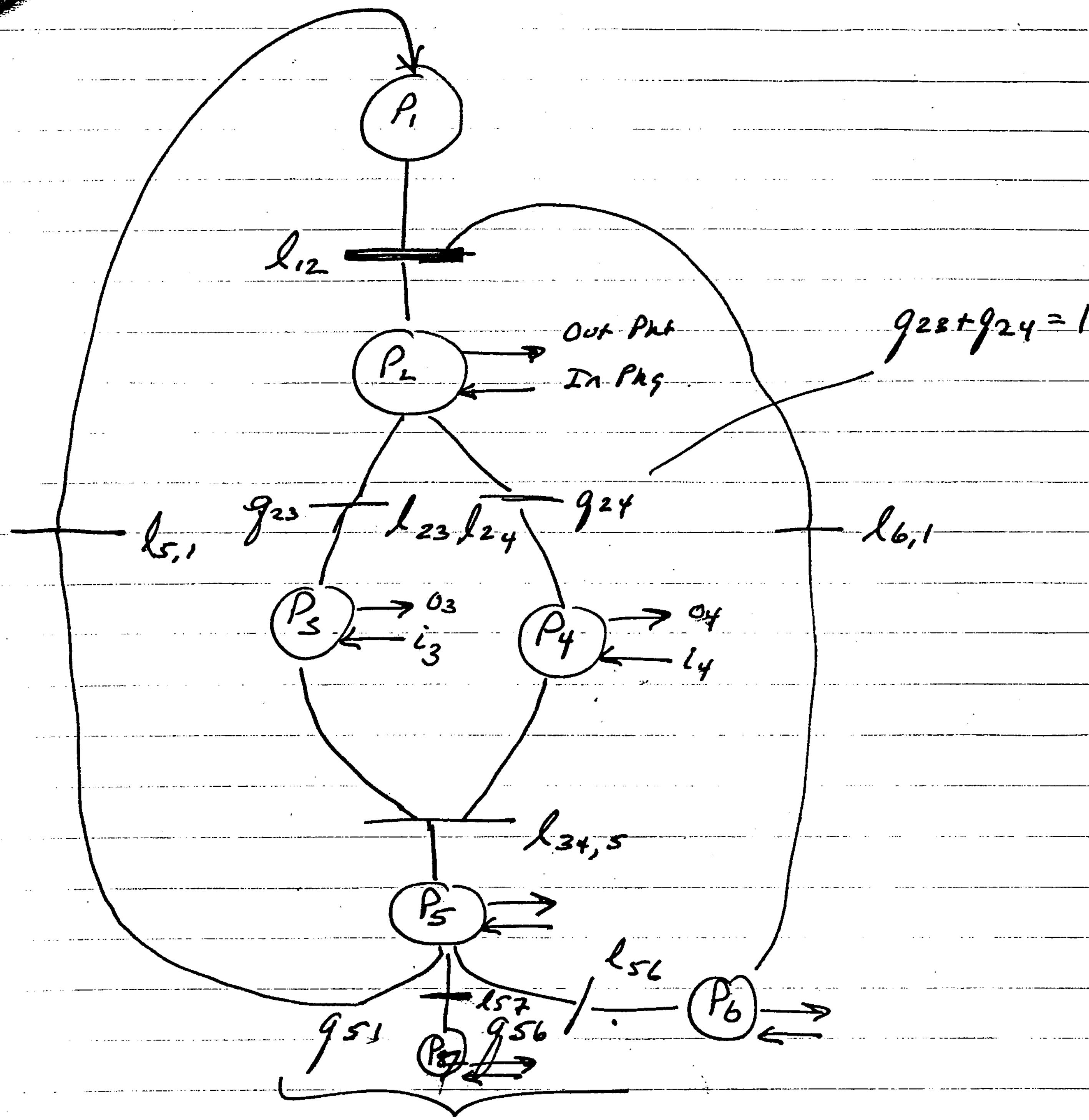


Theorem 3: $\bar{L} = L_{user} + L_{comm} + L_{proc} + L_{data} \Rightarrow \bar{L} \approx \bar{L}_{user}$
$> \quad > \quad >$

Discuss Diff.

Theorem 4: (Source Rate Theorem) $\bar{R} \triangleq \frac{\bar{B}}{\bar{L}} = \frac{\bar{B}_1 + \bar{B}_2 + \bar{B}_3 + \frac{p}{1-p}(B_2+B_4)}{L_1+L_2+L_3 + \frac{p}{1-p}(L_2+L_4)} \approx \frac{B_2+B_4}{L_2+L_4}$

Figure 3.x Model Example of SDM



$q_{57}+q_{51}+q_{56}=1$

P_7 = stop state

To develop the source model, we need to do the following steps:

o Define and develop the PN for the characterization of the Hypertext interaction with the end user.

o Assign probability distributions to the firings of the transitions. Typically, exponentially distributed random variable are adequate for this assignment.

o Assign transition probabilities to each set of non-deterministic transition. A non-deterministic transition is one in which an arc from a place to a transition occurs several times from one place. The transition probabilities are for each non-deterministic place, P_j, and we have the transition from P_j to P_k or P_j to P_n. Define the transitions:

$$q_{jk} = P[\text{to } P_k \mid \text{from } P_j]$$

and note that for each j:

$$q_{jk} = 1$$

$k \ H_j$

where H_j is the set of all transition places from P_j.

3.4.2 Model Structures

The source model characterized in the previous section combines the results that we had developed in Chapter 2 and those that we have developed in this chapter. In particular, we can see that the source can be given in terms of a fully stochastic model. Let us begin by modeling the input an output responses for the SDM. These responses are the corresponding input and output traffic on the communications network.

The input and output messages consist of the following possible elements:

- o Images
  - o Stills
  - o Video
  - o Voice
  - o Text
  - o Graphics
- o Commands
  - o Requests
  - o Responses

We can now characterize each of these in terms of bits of information, and furthermore characterize them in terms of data rates, namely bits per second.

Since the SDM is a stochastic Petri net, which is in turn a timed Petri net, we know that responses occur a set of define, albeit random, times. We call these times {L1,...,Ln,..}. Thus we have a sequence of input and output messages. Let us define Mi(t-Lk) as the input message at time Lk and Mo(t-Lj) as the output message at time Lj.

There are two classes of sources that we have developed. They are continuous and transient. A continuous source is one that never stops and continuously generates input and output messages. A transient source is one that does not last indefinitely and has a continuously decreasing probability of of continuing. We can provide a more definitive definition as follows.

Definition: A place Pterm is called a terminal plce if no transitions emanate from Pterm and there are no input or output messages associated with Pterm.

Definition: A SPM is said to be continuous if there is no Pterm.

Definition: A SPM is said to be transient if there exists a Pterm and if for all M(0) Pterm belongs to R(M(0)), and M(k) converges to Pterm as k increases. Specifically;

Let M( ) = {x,x,........,z}

where z marks Pterm and can take 0,1. Let:

M(oo) = {0,0,.....,0,1}

Then M(k) approaches M(oo) with probability approaching 1 as k approaches oo.

We can define the input and output message streams as follows:

mout(t)

= mo,1(t-L1)+mo,2(t-L2)+....+mo,k(t-Lk)+..+mo,n(t-Ln)

min(t)

$= mi,1(t-L1)+mi,2(t-L2)+...+$

where:

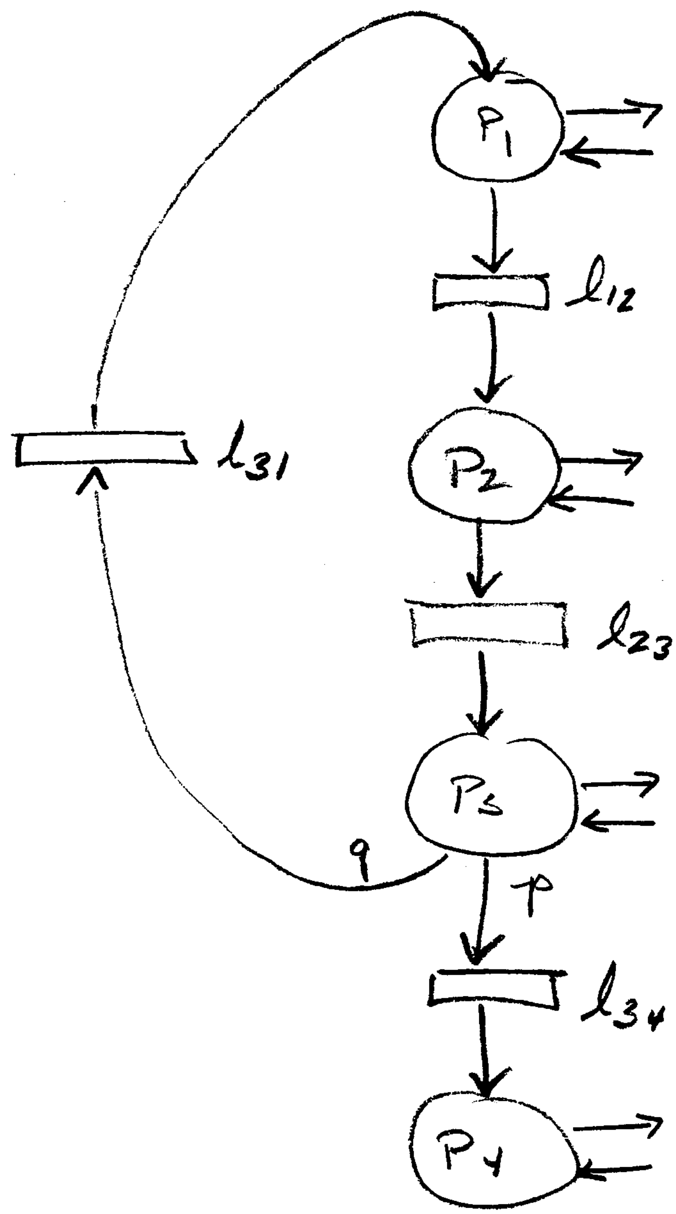
$mo,k(t-Lk) =$ response, request, image etc.

which depends on the probability assigned or directly upon the place transition characterization.

3.4.3 Model Performance Analysis

We can now take the models that we have developed and convert them onto source models for the purpose of systems performance analysis. Consider the simple four place system shown in Figure 3.x. P1 is the initial state and P4 is the final state. We associate the time delays Lk with each transition.

Figure 3.x State Sizing Example



We start by enumerating all of the paths through the network. A path is an ordered sequence of places that define the network. Associate with each path is a probability of occurrence. We assign to the nondeterminisic choice at P3 by assigning the probability of p to going from P3 to P4 and q or (1-p) to going again to P1. With this we can assign the following:

Path	Probability
P1,P2,P3,P4	p
P1,P2,P3,P1,P2,P3,P4	p*q
P1,P2,P3,P1,P2,P3,P1,P2,P3,P4	p*q*q

etc. We can readily add the probabilities and see that it adds to unity. Associated with each path is a duration Dk, where:

Duration	Probability
L1+L2+L3	p
L1+L2+L3+L4+L1+L2+L3	pq

etc. From this we can determine the average duration of a transient SDM.

We define E[L] as the average path length in seconds, where it is the length of each path times the probability of that path.

In a similar fashion, we can generate a model for the total number of input and output bits generated in this network. We do this again by following the progress through each of the paths in the network. We can then define the average number of input and output bits as;

E[I] = average input bits

E[O] = average output bits

Using this and the source duration we have the average input and output rates of the source;

$$Ri = E[I]/E[L]$$

$$Ro = E[O]/E[L]$$

If we have many such sources, then we can generate an average source rate by knowing the total number of sources, determining how frequently they are activated, and then averaging this over the individual sources own statistic. We expand on this in the problems.

3.5 Conclusions

This chapter provide the designer with the second element that is necessary for the development of the source model in a multimedia context. It allows for the characterization and development of a multimedia stochastic source and the specification of that source in terms its average characteristics as well as a full probabilities characterization. We shall see how this concept will be latter integrated into the overall system design and evaluation.

The key issues to be obtained from this discussion are those that relate to both the design and evaluation of the end user interface. Many works have been written on the end user design factors but as we continually see there are all too often designs that do not provide for ease of use and simplicity of access. There is the continual battle to provide the end user with all the flexibility that could be desired while at the same time increasing the complexity of the interface.

CHAPTER 4

Multimedia Storage

The storage of multimedia objects is a major issue not only to the basic archiving of the data but goes to the heart of the issue of multimedia characterization. As we have discussed before, the ability to do more with complex multimedia objects than just raw digitization will represent the major capability to deal with complex abstractions. Thee complex image abstraction concept is the heart of multimedia storage.

This chapter discusses the ways in which complex multimedia elements can be compressed and reduced to simple data elements. It also addresses the more complex problem of how best to abstract elements of images and to combine them in a fully interactive multimedia session.

Storage is one of the elements that are key to the overall design and performance of the multimedia communications environment. Often the concern is the availabliity of communications channels that are fast enoght to bring the images to the end user. The problem in realsity is not the communications channel but the storage elemet. In this chapter, we focus on the individual stroage elements and blend them together into a full multimedia stoage capability. In addition we focus obn the system iussue associated with the syayet perfrmanmce and sizing that isthe driving force in all of the design tradeoffs in this text.

4.1 Storage Architectures

In this section we present an overview of the key characteristics of storage devices and in addition provide the key factors that go into the performance analyisi of the storage systems.

4.1.1 Storage Elements

4.1.2 Storage Access Characteristics

4.1.3 Storage Performance Parameters

The overall performance factor for the operations of a storage system is the access time fro a specific file element. In a multimedia system, the performance factor is the access time for the compound multimedi file element. There are four major elements of the access time. These are:

- o Bus access speed and time
- o Memory unit latency access time
- o Data element seek time to find the data.
- o The data transfer time from the medium to the memory display device.

4.1.4 Multiple Storage Techniques

We have just completed the analysis of system that have single storage media and the perfromance that results. When dealing with multiple media we are dealing in two dimensiona of further integration. These dimensions are:

o Image context integration, integrating voice, video, text, images etc from differing multimedia sources.

o Storage device type integretaion, taking multimedia storage from multiple devices at the same time and creating a compund multimedi storage construct.

4.2 Storage Alternatives

There are many storage media that are available for the temporary storage of the multimedia elements that are used in an operational context. These media are in range fro very fast but not grate in extent, to fairly slow bu of high density. One of the driving factors for the choice of the specific storage medium is the cost per bit of storage and its accompanying access time. In this section, we shall not focus on the cost issues since they are generally so volatile but urge the reader to review the issue whrnever develpin a system design.

4.2.1 High Speed Memory (RAM)

4.2.2 Disk Storage

4.2.3 CD ROM Storage

4.2.4 Tape Storage

4.3 File Formats

Having the specific storage device is one of the elements in the overall stroage design problem. The second, and equally important, is having the proper file formats. There are many types of file formats for the efficient storage and retreival of multimedia data elements. Clearly, it should be obvious that the storage of voice, video and images may require different file formats from both the perspective of placing the information in storage and in ultimately retereiving it.

4.3.1 Text Fiules

4.3.2 Voice Files

4.3.3 Image Files

4.3.4 Video Files

4.4 Multimedia File Retreival Alternatives

Having discussed the overall issues, technology and file structure, we now will develop the overall system view of a multimedia file reteival system. In this section we shall focus on the overall implementation of the architectual aleternatives and develop methods and models for the analysis of performance and sizing of the luttimedia file structures.

4.4.1 Storage Devices

4.4.2 Access Techniques

4.4.3 Performance Measures

4.4.4 Access Optimization

Having developed the measures of overal, system performance, there are several isssues that relate to the optimization of the file system architectures. We have noted that severalissues awere key to the design and perfromance of a multimedia system. These issues are:

o Access time per element

o Multimedia access time and compound access integration time

o Consisten file namin and formtiing of data.

4.5 Conclusions

In this chapter we have focused on the element of the multimedia communications system that stores and retreives the data elements. The key issues are those that relate to the technological alternatives and the concerns relating to the performance of the system as an overall communications system. The issues of stroage capacity, the layout of the specific files and the analysis of the access time to retreive files are key to the development of an effcicient overall system. We shall use these result as an element of the overall system disghn latter inthe text.

CHAPTER 5

Communications Environment

The ability to interconnect many users with multimedia applications has developed over the years into a structured and layered methodology. This chapter presents to the reader an overview of the communications environment, focusing on broadband systems, and expands to coverage of the overall communications lawyering protocols.

The key factor that drives the communications environment for the multimedia user is the need for a connection based service. Connection based services assure the user that there is a synchronous service provided that allows for the end to end integrity of the image.

In this chapter, we discuss the communications element from the aspect of what is needed to effect the overall sessions service and to implement the complete multimedia communications environment. We take an approach that is the inverse from that of the standard communications study, starting with the higher layers of functionality first and moving downward towards the transport function as the latter element.

We view communications as a means to an end, but a means that has a cost in terms of delays, capacity limitations and errors. As indicated, we begin by assuming that the communications medium has all of the elements that are necessary for the errorless interconnectivity of all the users in the session. It is the assumption of all of the players being in the same room at the same time. As we relax these assumptions, we get farther and farther into the limitations of a real communications environment.

5.1 Requirements and Architecture

The issues of the communications environment revolve around the questions of what communications do and how much is needed to effect the necessary task. The multimedia environment places many more requirements on a communications system than is placed by any other usage. In the multimedia environment we are dealing with very high data rate requirements that demand high throughput and this moves quickly into the 100 Mbps to Gbps range of data speeds. In addition we are dealing with an environment that demands that requires that there be minimal delays and that the messages are synchronized. Also we demand that the users be allowed to develop the session structure that we have been developing and that the sessions be transparent to the communications network.

In the multimedia environment, we are dealing with a world of communications that is quite unlike that of the computer to computer communications world. In the computer only world, the end user, namely the computer, can be designed through its software to adjust for the vagaries of the communications channel. In fact, the development of the layered architecture that we discuss in the next section was a direct result that the communications channel was unreliable from the point of view of throughput, errors and delay. This then required layer upon layer of protocols to adjust for the channel. As we evolve new channels, we can envision correcting many of the errors of the past and providing a much more enabling environment.

5.1.1 MM Communications Environment

The multimedia communications environment is complex in that it demands that any set of users of the system be allowed to be interconnected in any fashion and further that they share the resources of any of the elements of the overall system, either separately or in concert. We have seen that in a multimedia environment that we combine together the forms of communications such as voice, video, image, text, and standard record files, and that we can create many virtual users, such users being real people, applications programs or database and other devices. The ultimate embodiment of a multimedia communications environment is the session which reflects the ability of all of these users to share complex multimedia data objects in a seamless fashion.

When we look at the communications environment of the real world we see that it is often lacking in its ability to provide the resources necessary to effect this capability. The standard communications system is a voiced based network that provides data rates typically of 56 Kbps or less to the end user. Even with the introduction of the proposed ISDN network, this moves up to 64 Kbps, which is still too little too late for the multi media environment.

There is, however, an available set of higher data rates in the larger network, rates up to 45 Mbps. However, these are not provided on a shared switched basis, only on a dedicated basis. The question may be, why do we need shared and switched. The answer is that we really do not want to pay for dedicated fiber from one location to all other locations on any session connection that we may desire to connect. The shared switched network works well for voice communications, since that is the essence of the current telecommunications network.

Figure 5.x depicts a view of the current communications network. The network depicted is that of the shared switched public network which is generally a hierarchical network, consisting of local switches and a higher switch hierarchy.

Figure 5.x Current Telecommunications Network

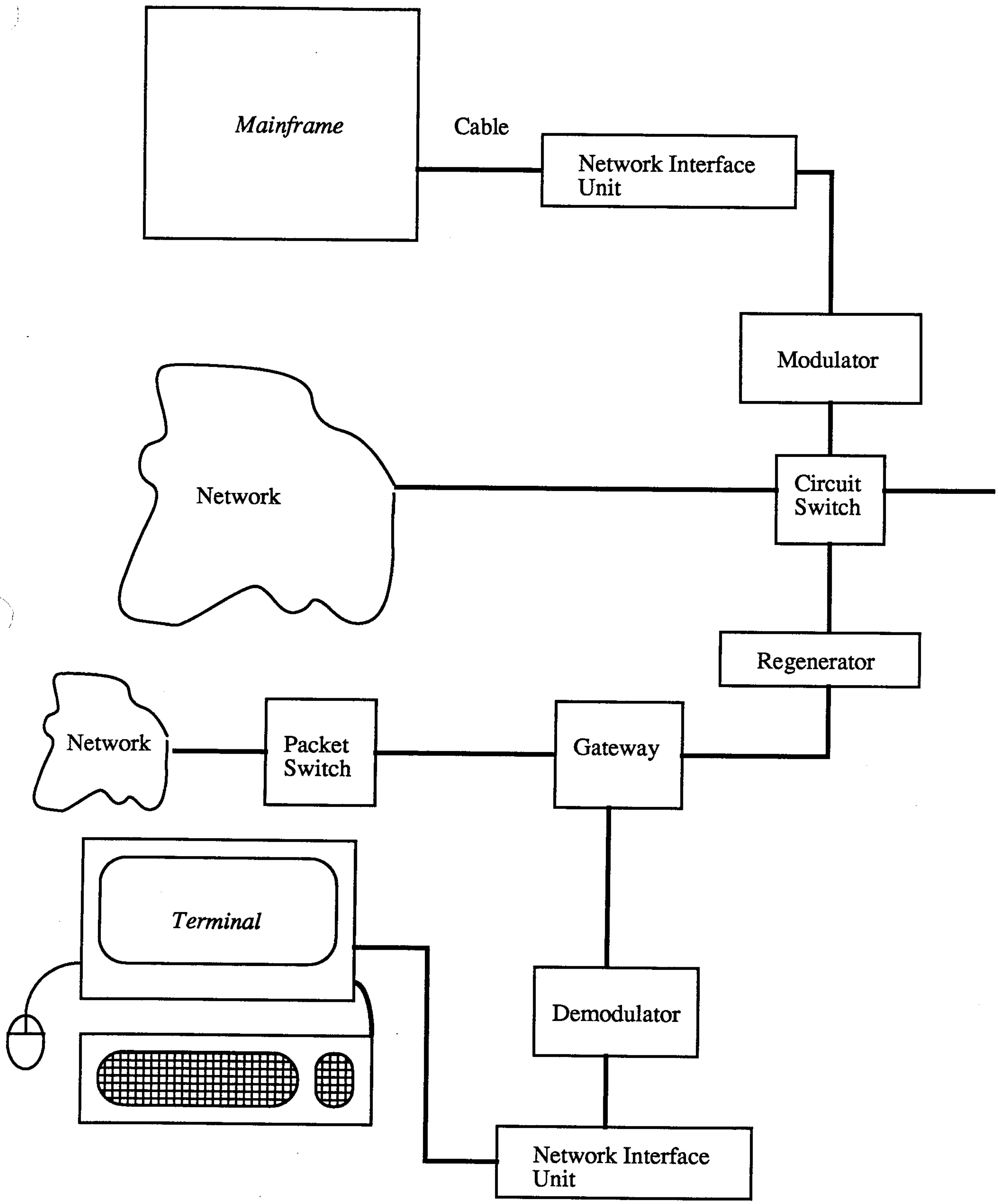


Figure 5.3 : Network Sources and Delays - Examples of Sources (in *italics*) and Delays.

The environment to support multimedia communications needs is generally quite limited at the present time. For example, at the time of the writing of this book (1989), there are only 45 45Mbps circuits that have been sold under tariff in New York City. In contrast, the city is the media capital of the world, with the highest concentration of media intensives users anywhere. Prime amongst these users is the printing, publishing and advertising industries. Millions of images and hundreds of thousand of hours of video are created and moved from one location to another around the city on a daily basis. In fact, the most common form of communications is the use of bicycle messenger, each moving at significant speed across the city streets. In fact, these bicycles are moving in the Gbps range!

Thus despite the needs for this multimedia communications market, there is yet an established infrastructure. This is generally because of the lack of the needs of the multimedia communications environment. These requirements are as follows;

o Effective and Available I/O Devices

o Shared Switched Networks

o Layered Architectures

o Protocols and Standards

o Connection Based Services

o Video, Voice, Data Integration

o Local and Long Distance Interconnect

These requirements are shown as satisfied in the network presented in Figure 5.x.

Figure 5.x Broadband Multimedia Network

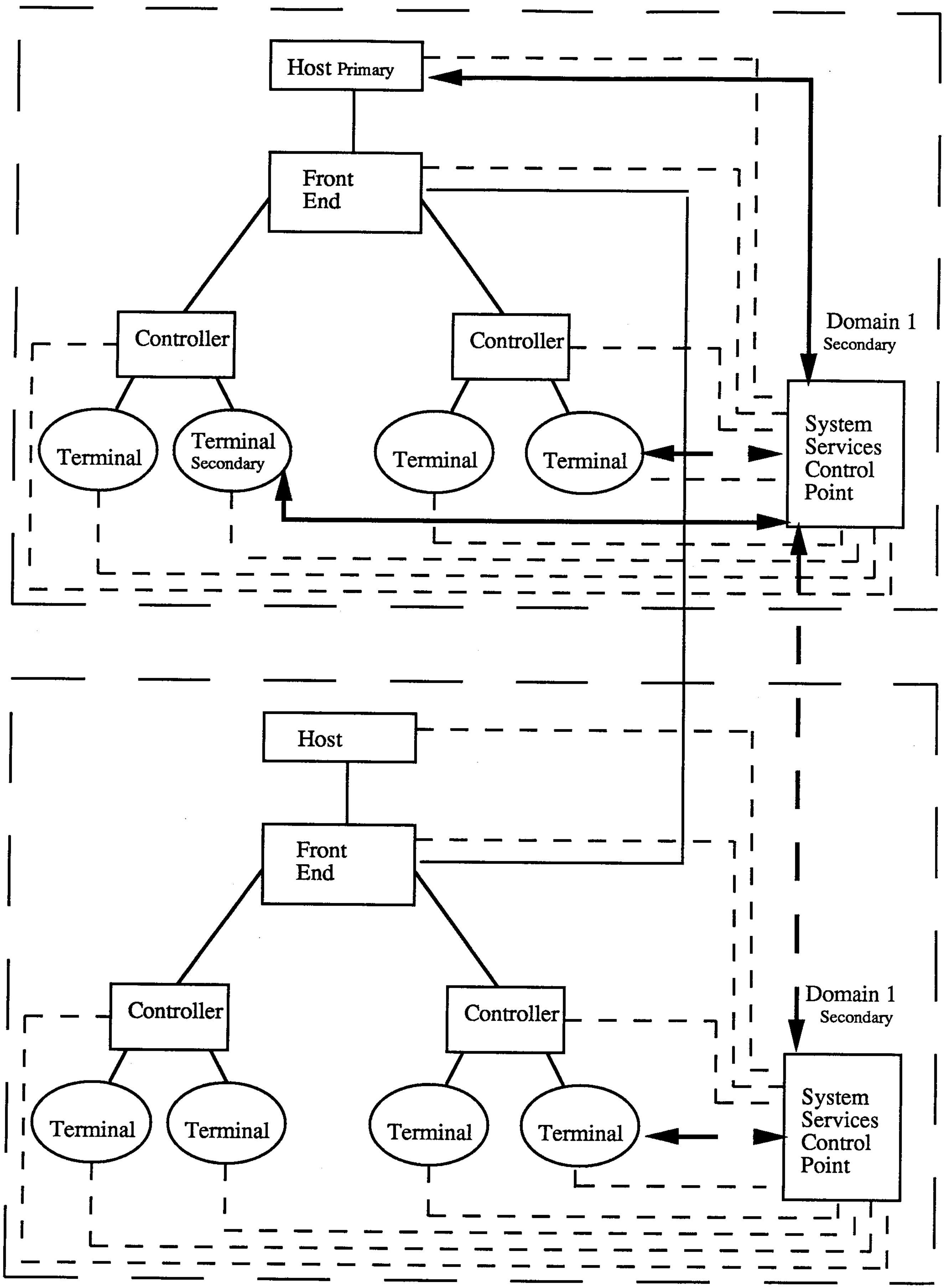


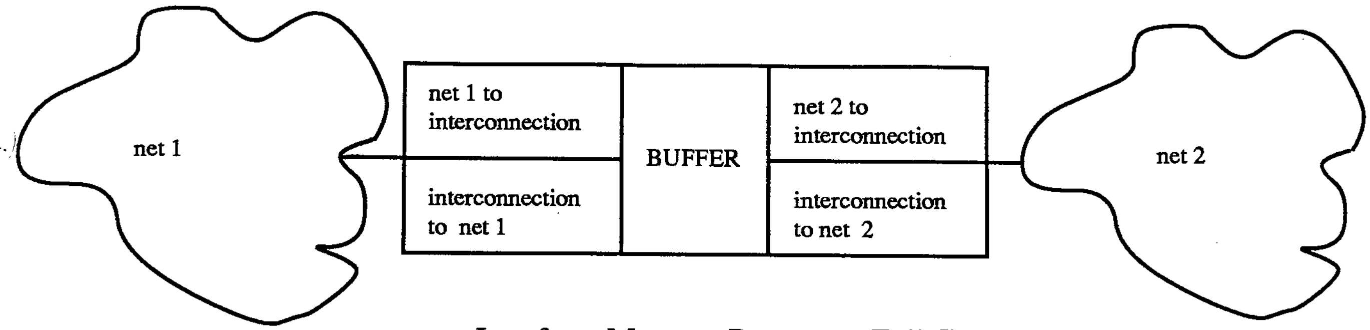
Figure 5.1 : Session Communications - inside a domain ——— and between two domains — - .

5.1.2 Requirements Factors

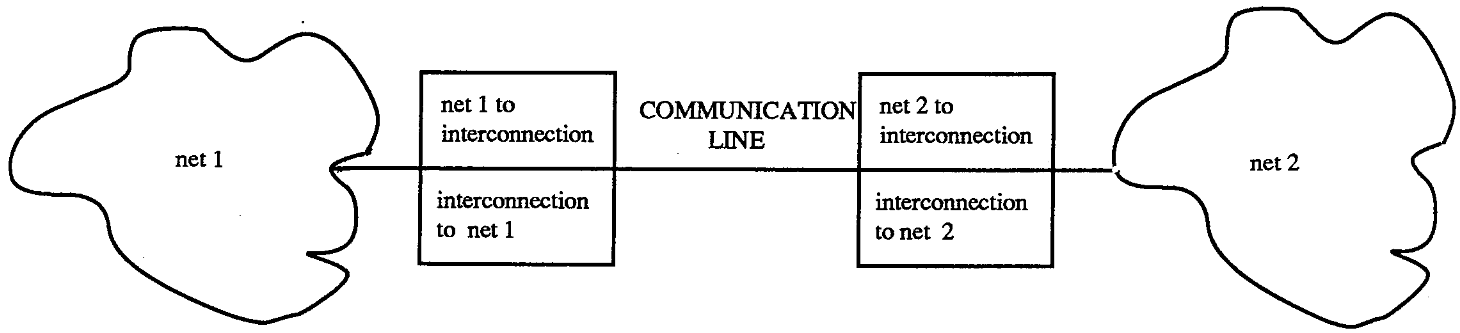
Based upon the discussion of the multimedia environment it is possible to develop a detailed set of requirements for a general multimedia environment. In this section, we focus on some general requirements issues, but note that special network requirements may be necessary for very specific applications.

- Interconnectivity
- Addressibility
- Shared versus Dedicated
- Grade of Service (Delay versus Blocking)
- Interfacing and Interconnecting (Local versus Long Distance)
- Connection versus Connectionless
- Data Rate

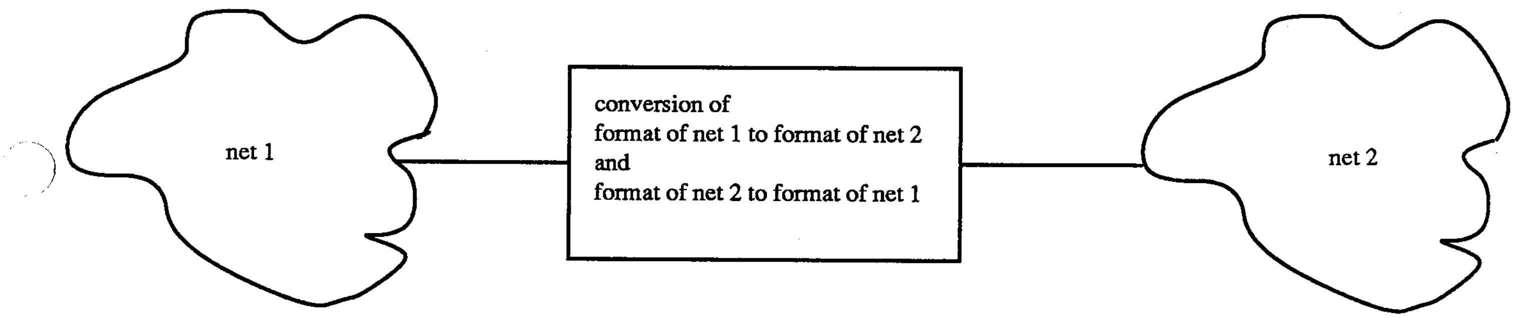
Table 5.x Requirement Factors



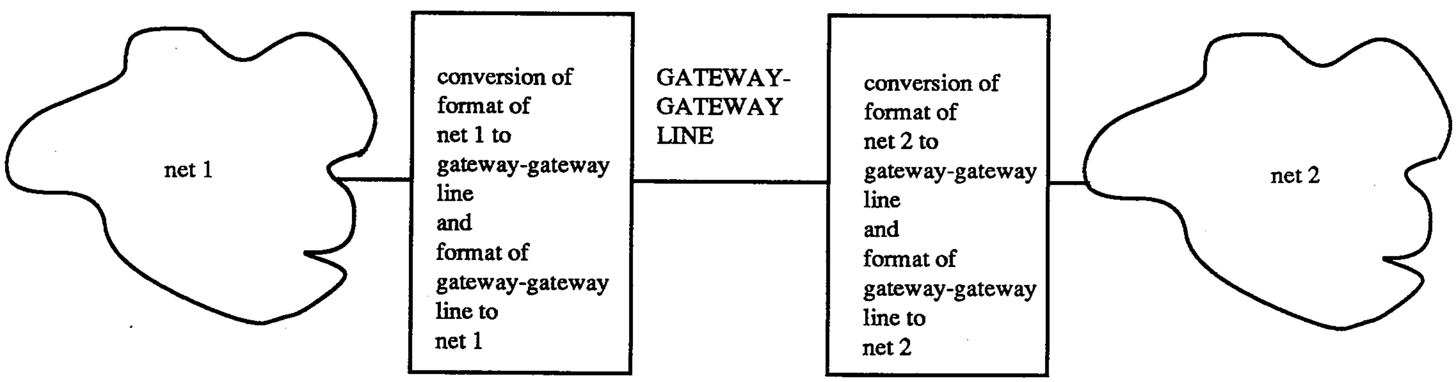
Interface Message Processor Full Gateway



Interface Message Processor Half Gateway



Host Gateway



Half Host Gateways.

Figure 5.2 : Network Interconnectivity.

5.1.3 Performance Issues

As we design a multimedia environment, we are required to develop a set of measures that detail the performance of the network. The performance measures take the loading on the network, based upon the number and types of users and their frequency of use, and determine how well the network can provide the services required. The issue of Grade of Service is a set of benchmark levels that tell the user what to expect from the network. Performance tells the users how the network can vary from the specified grade of service.

The main performance issues to be developed in this chapter are those that have been accepted as the most effective measures of network performance. These general performance measures are;

o Effective Data Rate

o Throughput

o Delay

o Errors

o Delay

o Capacity

o Variability

Figure 5.x Performance Analysis

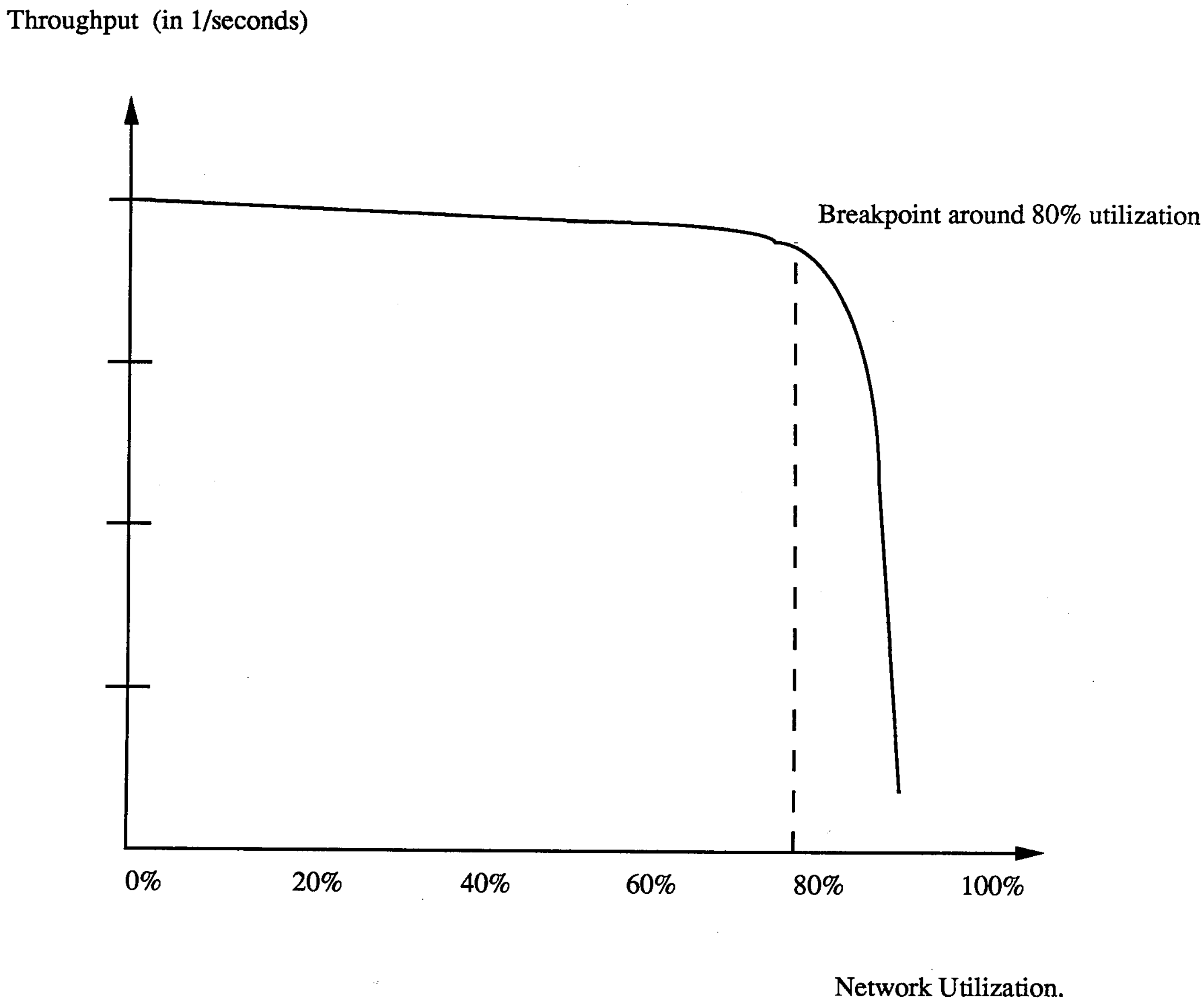


Figure 5.5 : Throughput vs. Load - Throughput vs. Network Utilization for a Typical Token-Ring LAN.

5.1.4 Sizing Issues

Sizing is the complement of performance. In the performance model, we assumed that the load on the network was determined and that we then proceeded to develop measures of performance. In the sizing analysis, we assume that the performance level have been given and we then determine how large a user base is supportable. In performing a sizing analysis, we determine the limits of usage of the system.

The major sizing elements for a multimedia system are as follows:

o Number of Users

o Number of messages per source per unit time

o Message Mix

o Maximum Data Rate per Channel

o Maximum Number of Instantaneous Users

Figure 5.x Sizing Analysis

5.1.5 Architectures

We have continuously, in this book addressed the issues of architecture. The architecture issues are the first in any system design and provide the blueprint for the development of any detailed design. The architecture elements are those that must be configured to meet the need that are articulated by the users of the system. In this section, we first detail the architecture elements and present some of the architectural alternatives that we shall be developing in this chapter for the communications element.

There are five architectural element that we consider important for the communications environment. They are;

- o Network Topology or Physical Layout
- o Addressing Formalism
- o Switching Capabilities and Alternatives
- o Functionality Interfacing
- o Interconnection

(i) Network Topologies

In Figure 5.x we depict several network topologies that are in common use. The tree topology is a common topology in such areas as cable television networks. This topology has advantages for networks where the major function is the distribution or broadcasting of high bandwidth data. Thus they find common use in cable networks (see McGarty). The tree network is a hierarchical design that assumes that the system flows both to the headend location of the communications network. The existing telephone network is of this type in certain locations and is clearly the case in the long distance communications configuration.

The star topologies has been accepted as a viable local topology for fiber networks where the fiber is generally a single mode type and cannot be adequately tapped and bridged at multiple locations. Single mode fiber is not amenable to these types of taps and thus it usually finds itself in point to point interconnects. To expand a point to point connection to a network topology, we find the star architecture as the appropriate one.

The ring topology provides a totally connected structure that allows each users to be interconnected onto the network at their location and further allows for the condition that if the network is severed at any location, there may still be a path to other locations, albeit not functioning under a ring structure. Thus, rings are used for applications where there is a need for some form or redundancy. We can further extend the ring structure to include a dual ring structure, one ring carrying traffic in one direction and the other ring carrying traffic in the opposite direction. We shall see that this becomes a common architecture in the metropolitan area network applications.

The bus network topology is a ring that has been broken. The bus may be a dual bus architecture that is similar to the ring in that one bus can handle traffic in one direction and the second bus handles traffic in the opposite direction.

Figure 5.x Network Topologies

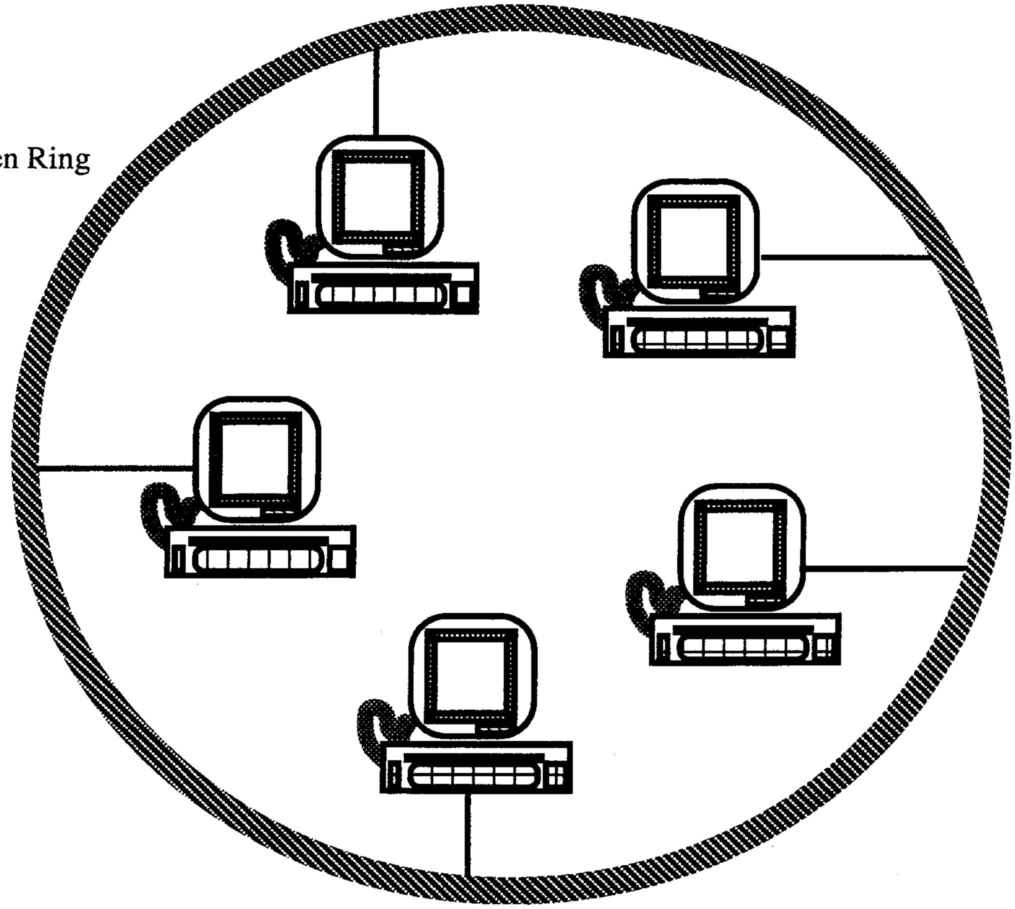
(a) Tree

(b) Star

(c) Ring

(d) Bus

Ex a: Token Ring



Ex b: Star

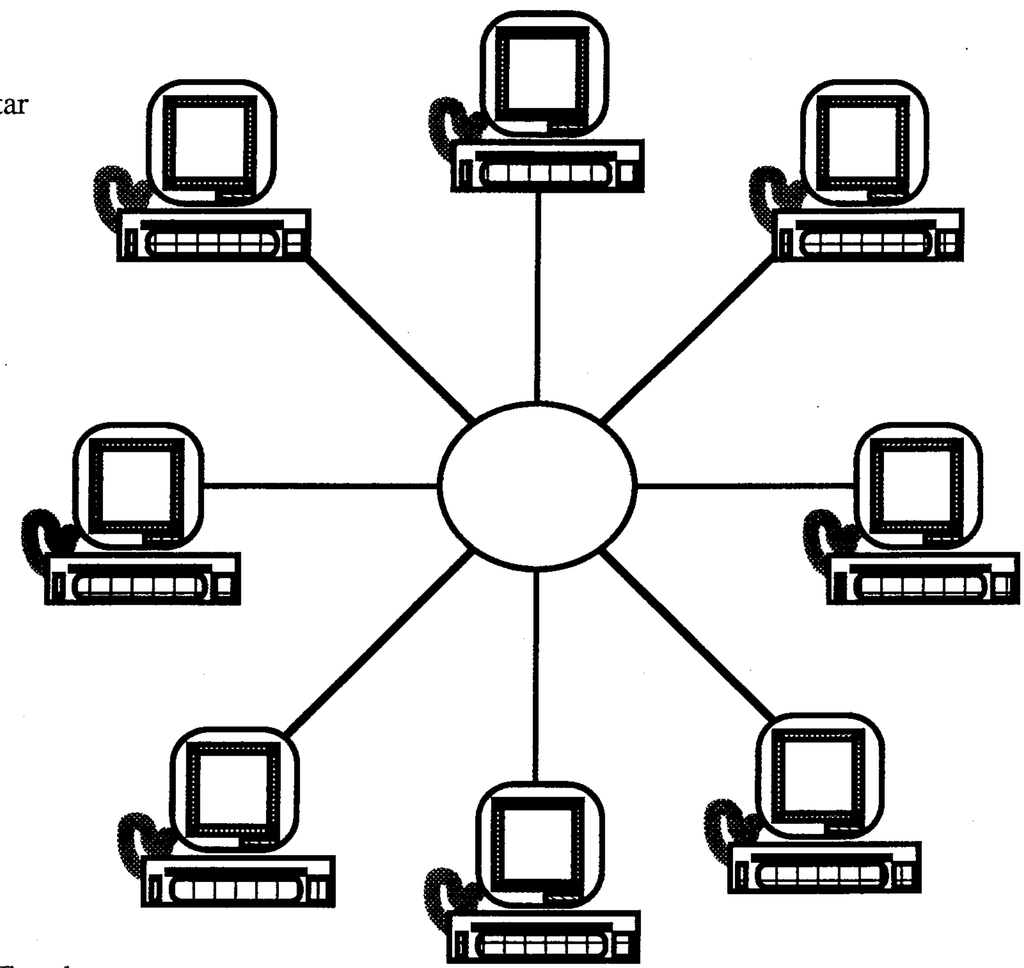


Figure 5.4 Network Topology

(ii) Addressing Formalism

Addressing is a fundamental issue in communications networks. WE frequently forget how easy it is to communicate in a telephone network because we know how to reach someone by means of their name. However if we look at the process in further detail, we see that the human naming convention is a complex data structure that we have had years of experience with and have accepted almost as part of our cultural history. We know that to call someone, we must know first where they live, we must then find the area code from an independent source, and then and only then can we dial xxx-555-1212. At that point, assuming that there is no telephone strike, we ask information for the number of John Doe. We may further have to delimit this if there are sever entries of that name. This delimitation is the location of Mr. Doe's house. Consider now that we must do this for all of the elements in the multimedia communications network.

Addressing thus applies to users, terminals, layer control elements, data bases and anything else, logical or physical that we wish to address. We shall see in this chapter that the essence of communications is not only getting the bits there but being able to say where there is. That is the addressing issue. We can look at addressing in two dimensions, shared and private networks. In a shared network, we must all agree as to how we are to be addressed. There are limited Vanity phone numbers, mostly limited to 800-xxx-xxxx. The classic number is the trouble reporting number for N.Y. Telephone for data circuits, 800-I AM-DEAD!

In private networks, the addressing methodologies are somewhat freer and depend basically upon the limitations of the supplier of the hardware and software. However, as anyone who has run a private network knows, the data and network administration task is significant.

Figure 5.x Addressing Alternatives

(a) Local

(b) Global

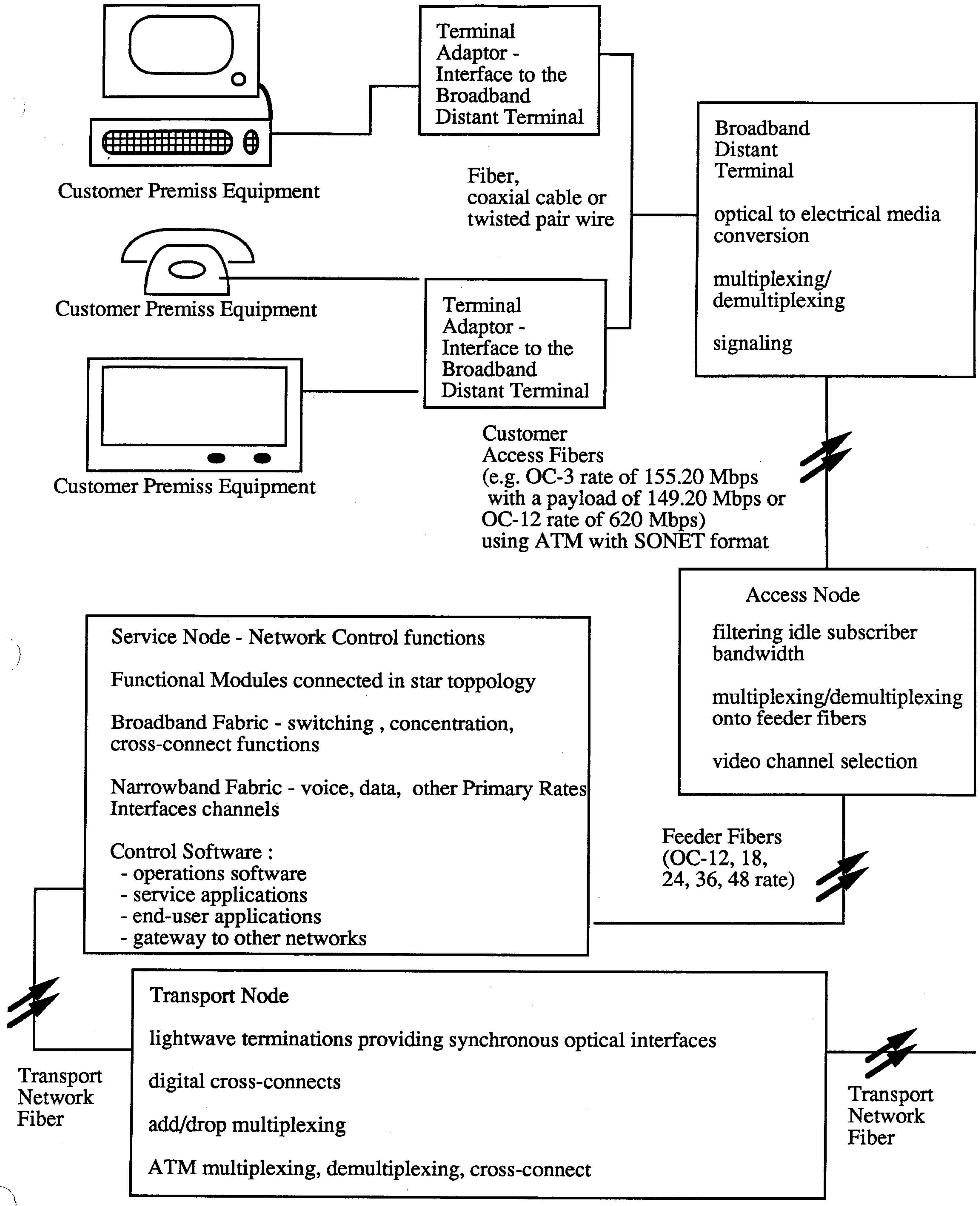


Figure 5.6 : Broadband Network Architectures - Example of BISDN Network Architecture for Local Public Serving Areas.

(iii) Switching Alternatives

In any shared network, the sharing is done by some form of switching. The function of switching is to allow any user to get to any other user and in addition to allow the user to gain access, on an demand basis, to any set of the network resources. The switching may be done in three different ways.

o Centralized: Each user accesses a centralized and hierarchical controlling element that determines the resources necessary and determines the path that is to be taken in providing the connection. The voice telephone network is an example of this type of switching architecture. The signaling to the central switch may be in-band or out of band. An inband signal is one that uses the same channel that the message transport occurs in as is done in the telephone network as today. Out of band signaling is increasing and is typical of the type of signaling in the new Signaling system 7 that will be a part of the ISDN network.

o Localized: In this form of switching, the end user equipment takes full responsibility for the selection, management, setting up, and control of the communications channel. Generally this is a complex function and there are very few system that implement this strategy.

o Distributed: This is a common switching strategy is its most common implementation is in the area of local area networks. It provides for any users to interface with any other user by all users watching message that flow across the network and determining which of the messages are for them and then selecting the message. The advantages of this type of system is that it allows for many users to activity participate in the network but it has the disadvantage of having a higher data overhead.

Figure 5.x Switching Alternatives

(a) Centralized (Hierarchical)

(b) Localized (End User Based)

(c) Distributed (LAN, MAN)

(iv) Functional Interfacing

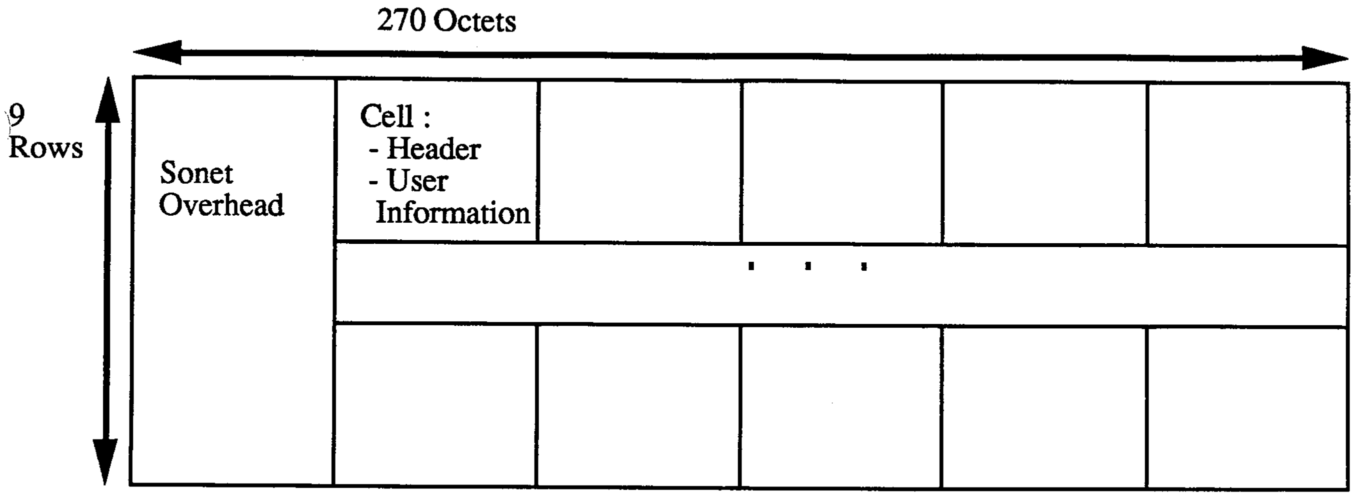
Interfacing the many elements in a communications network is generally done through a layered architecture.We shall see in this chapter, that the layered approach has been developed in such a fashion that it allows for effective and universal interconnection of all of the elements of a data communications network. The need for multimedia communications has not yet allowed for the development of an equivalent set of layers for multimedia environments.

The interfacing using layered protocols makes several assumptions about the communications environment. First it assumes hat there are well defined and structured level that can be segmented into separate control and implementation levels. Thus if the is a level that performs just electronic functions we can segment all of these function into that level. This is what is done in the OSI physical layer. If there is an agglomeration of functions that handle the switching functions or the graphics interface functions, we could just as easily load all of these elements into that layer.

Thus layering requires a commonality of functions, a collection of those functions into separate layers, and the interfacing of those layers into the overall network. The advantage of the layered architectures is that that they permit the development of software and systems that can be segmented to improvements on a layered by layer basis. The major disadvantage is that with more layers, and more segmentation, we give up speed and performance for the flexibility to develop the individual layers. We shall see in this chapter some of the effects of layering.

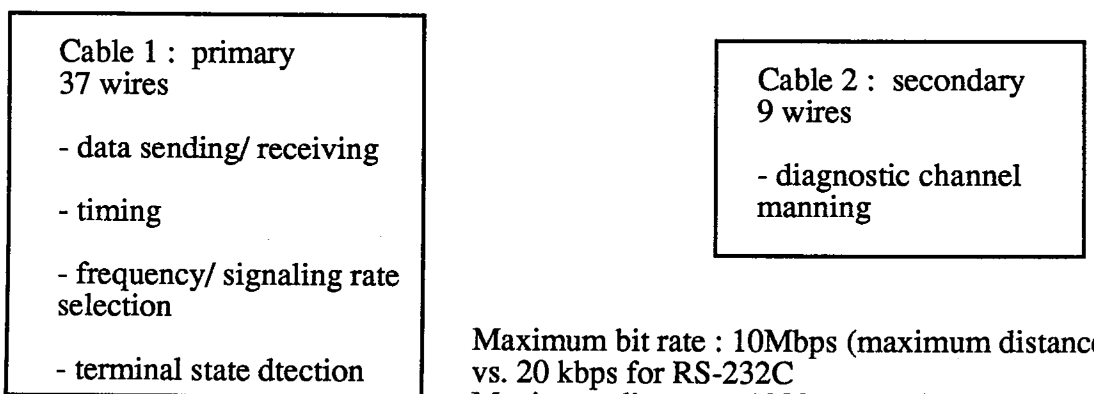
Figure 5.x Interfacing

(a) Layered Systems



The channels are labeled and asynchronously multiplexed.

Figure 5.7.a : Broadband Interfaces - User-Netwotk Interface Channel Structure Using ATM Techniques and SONET Framing.



Maximum bit rate : 10Mbps (maximum distance of 15m) vs. 20 kbps for RS-232C
Maximum distance : 1200 m (maximum bit rate of 90 kbps) vs. 50 ft. for RS-232C.

Figure 5.7.b : Broadband Interfaces - RS-449A Serial Interface.

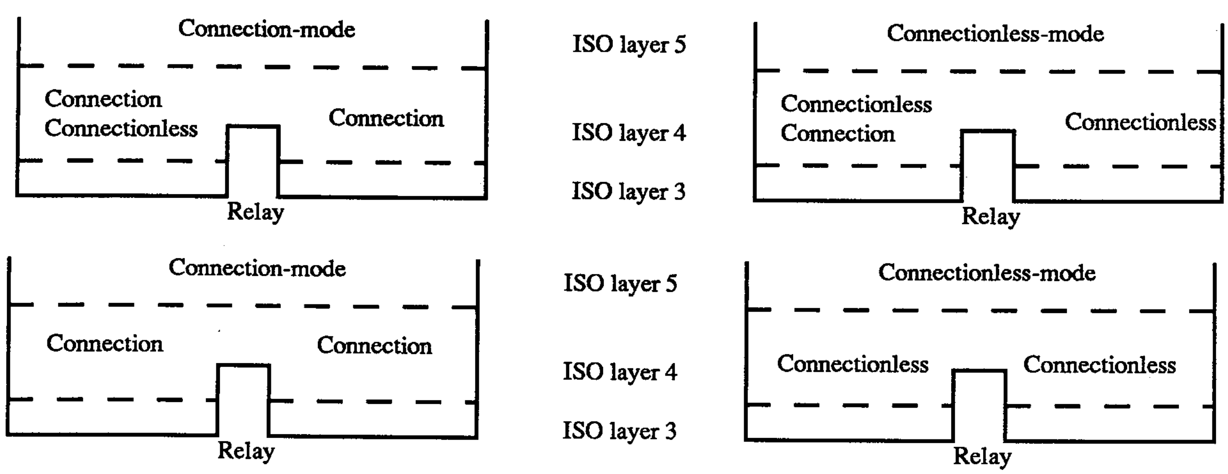


Figure 5.7.c : Broadband Interfaces - Connectionless/Connection Mode Relaying at the Transport Level.

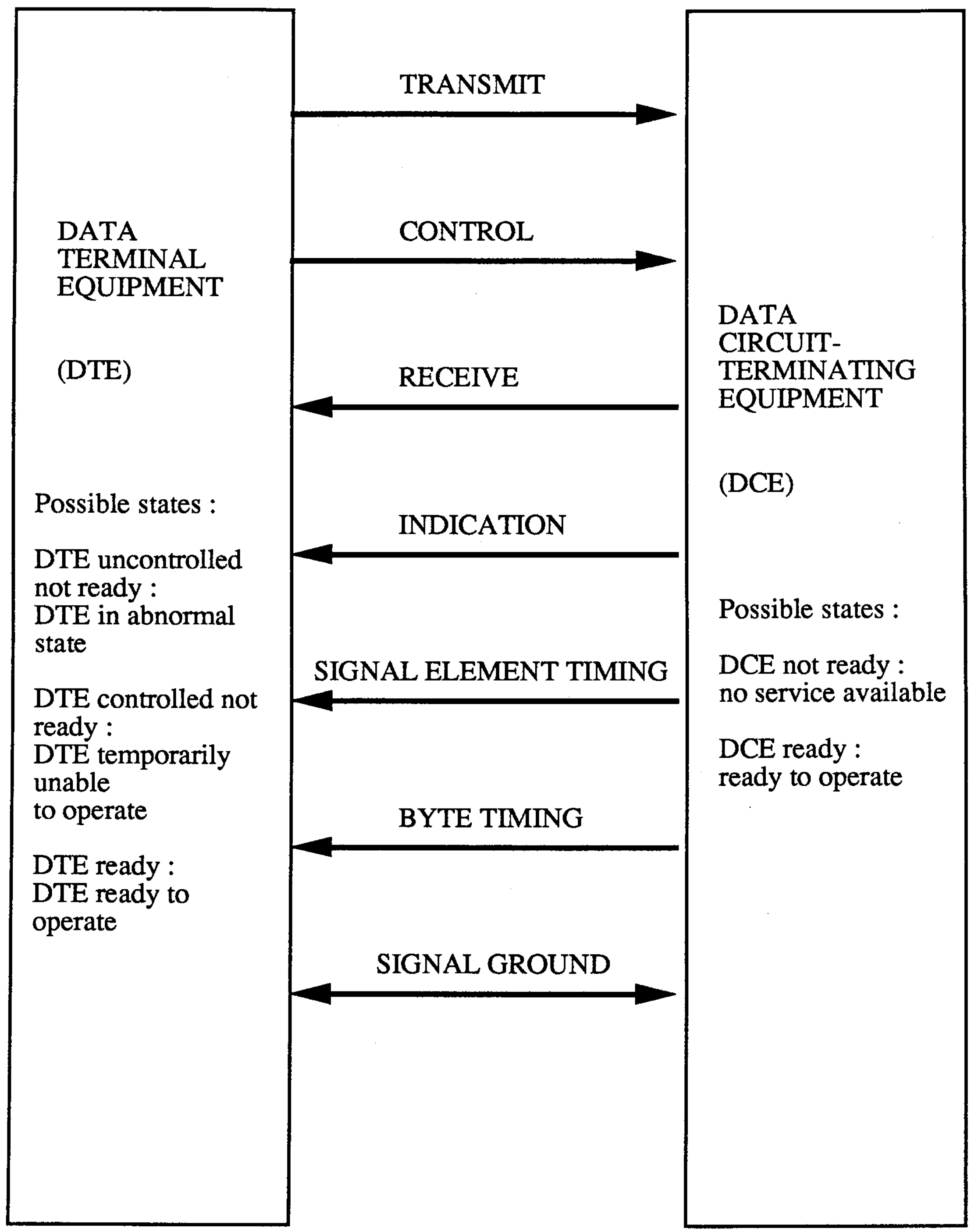


Figure 5.7.d : Broadband Interfaces - CCITT Recommendation x.21 DTE/DCE Interface.

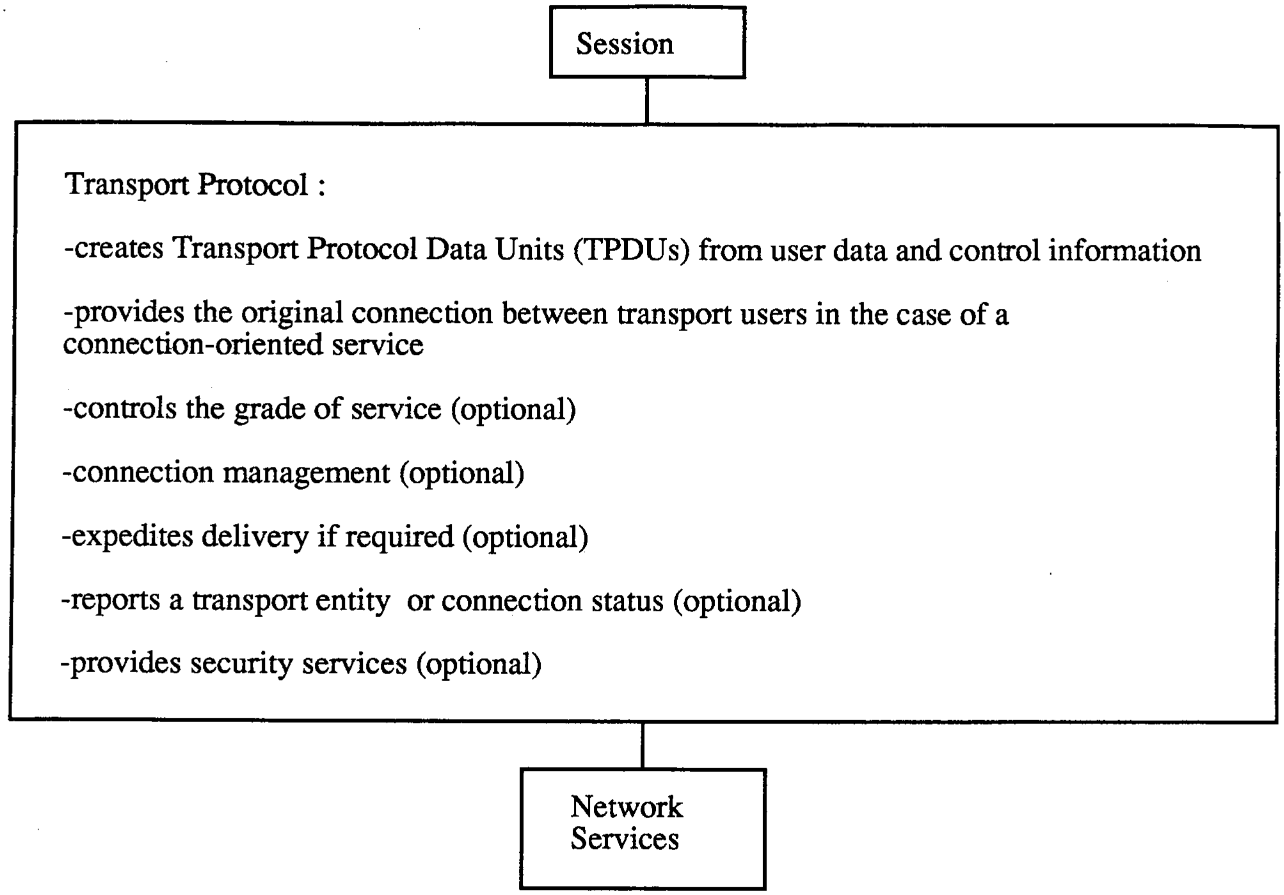
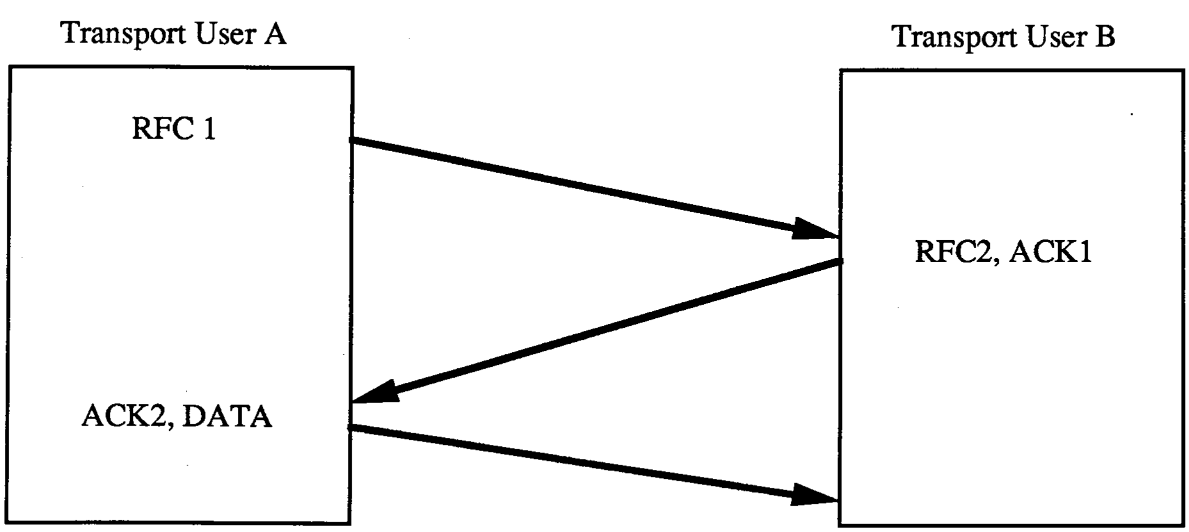


Figure 5.8.a : Broadband Transport - Transport Protocol Services.



Legend : - RFC : Request for Connection
- ACK : Acknowledgement

Figure 5.8.b : Broadband Transport - Connection Initiation Using the Three-Way Handshake.

(v) Interconnection

Interconnecting networks is a critical design and architectural function that must be considered in any communications network design. The interconnection is driven by the fact that no single network is self sufficient and that the world is becoming an amalgam of many different types of networks. Overlays of broadband multimedia networks are typical and will become a more common feature.

There are conceptually four generic types of interconnections. They are:

(i) Repeater: This provides for a physical to physical interface connection between to different networks. Thus a repeater may be nothing more than a modulation and demodulation pair of devices in a network.

(ii) Bridge: This type of interconnection takes the first step in interconnection networks where the data format of the bits may be different. Thus if we want to connect one type of LAN to another, or even a LAN to a MAN, we can do so through a bridge, The bridge matches the physical difference in the signal structure as well as the data format.

(iii) Router: This element of the interconnection hierarchies allows for not only the physical and bit format matching but also handles the changes in the addressing between the two networks. Thus we can go from one network where the addresses may be in one format to another network with a different addressing scheme.

(iv) Gateway: When everything is different from one network to another, we need a gateway to match all of the changes from one to the other

We show how these are related to ne another in Figure 5.x

Figure 5.x Interconnect Alternatives

(a) Repeater

(b) Bridge

(c) Router

(d) Gateway

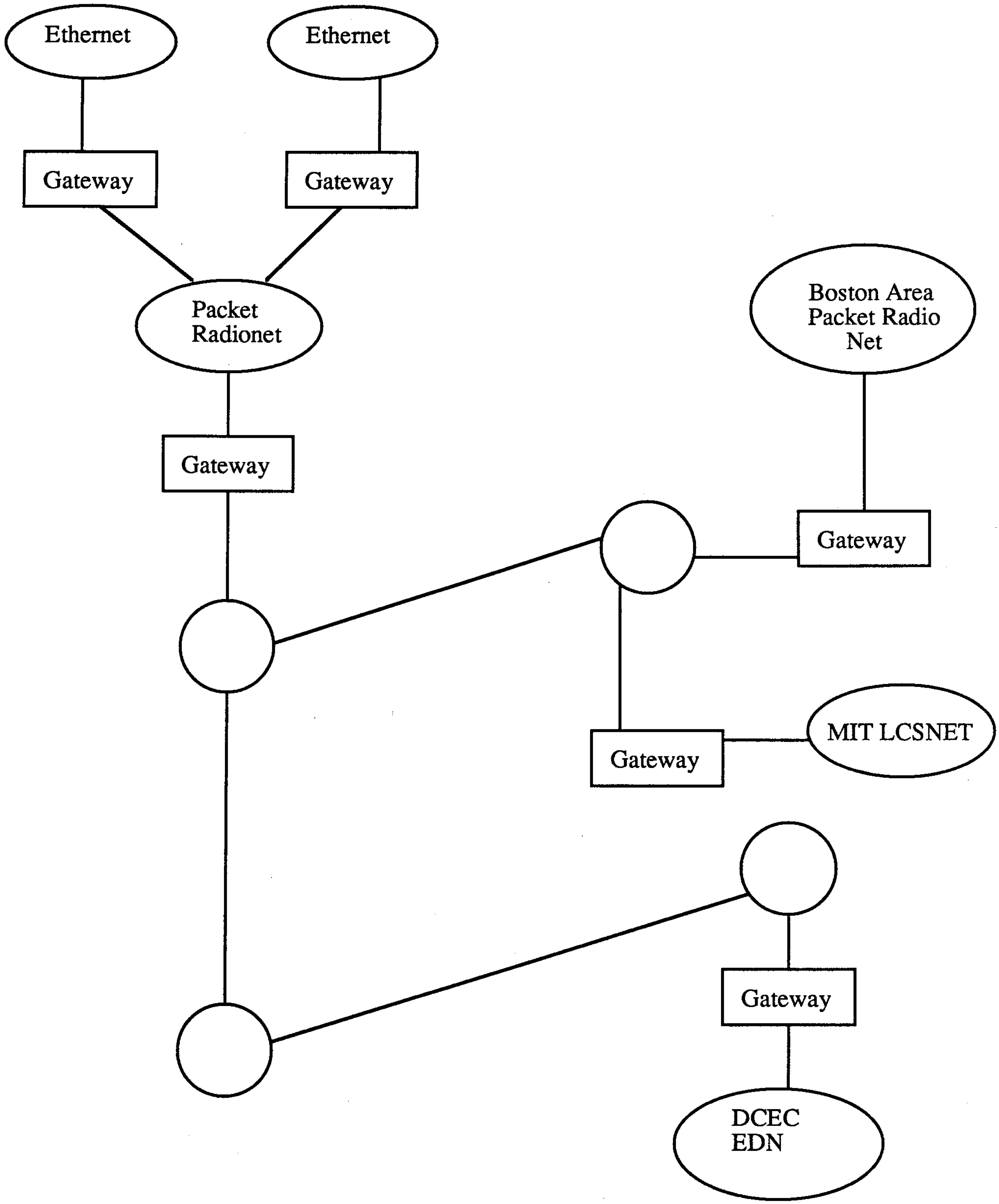


Figure 5.9.a : Broadband Topologies - Some of the Interconnected Elements of the DARPA Internet System.

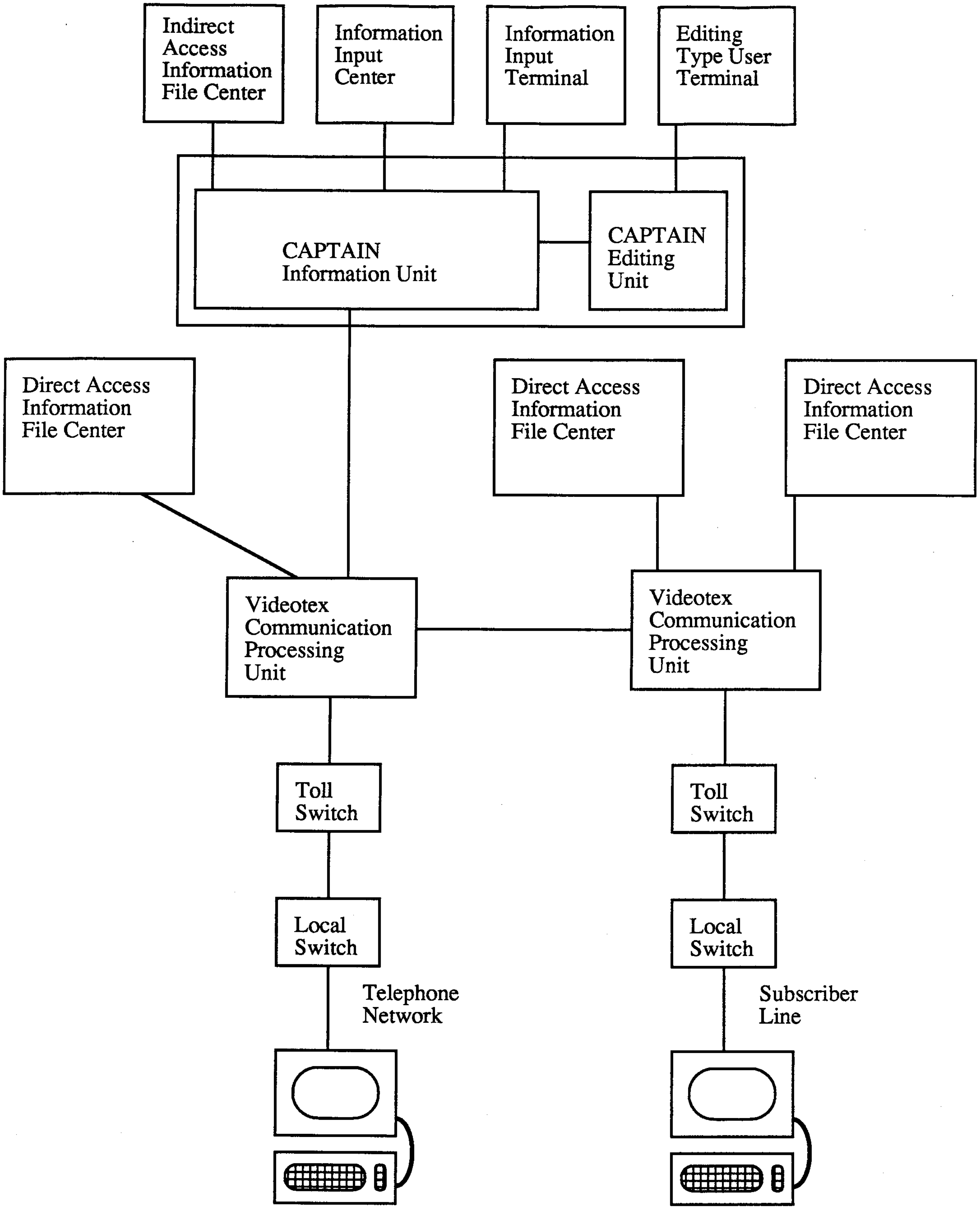


Figure 5.9.b : Broadband Topologies - the Japanese CAPTAIN Commercial Videotex Service.

5.2 OSI Layers and Standards

There has evolved a standard way to view the communications environment over the past ten years that involves the use of layered protocols. As the reader may recall, we introduced the layered approach first in Chapter one when we introduced the overall multimedia environment. We extended this to software layering in the Chapter 2 when we developed the source and in Chapter 3 when we integrated the GUI concepts with such elements as X windows. We further extended this in Chapter 4 with the data storage and file elements of the design.

In this chapter, the layered constructs have taken even a firmer form with the introduction of international standards. This section reviews the seven layer international standards for communications systems. Layered approaches are useful for both implementation and interfacing purposes. In the implementations context, the layered approach allows for the clear delimitation of functions and shows how they can be partitioned to allow for simpler system design and coding.

Communications networking, like any form of human communications, needs to have a set of protocols or agreed to sets of standards of handling many talkers at the same time. The protocols are ways in which the speakers or participants in any conversation can know how to interrupt and who has precedence in the conversation. A typical example is a meeting in a highly structured company. If the CEO is at the table, all conversation may generally defer to that persons comments and the level of interrupts may depend on the title and status of the people at the conference table. Inhuman conversations, we frequently have learned these protocols in a cultural environment and they are generally not written down. The need for addressing written protocols occurs when we are dealing in cross cultural environments such as a U.S. business man doing business in Japan. n that case, the difference in and acceptance of protocols is critical.

The same situation is prevalent in communications, and more importantly in multimedia communications. We approach this problem with the development of protocols and these protocols are developed in a highly structured manner. Historically, these protocols were developed in an environment of data communications, where the need was to allow for the interconnection of computers with other computers. The protocols were not envisioned as a way for human interaction with the communications environment. We shall see how this has evolved as we describe the protocols in some detail.

The protocols that we shall describe are in seven layers, each of the layers having the responsibility for a specific set of functions that are necessary for the communications network. Our approach to these protocols is different than most in that we shall begin at the highest layer and work downward. The typical pattern of development and the actual path of development in the standards bodies was from the bottom up since they were interested in protocols that satisfied the needs of the data communications community. In our analysis, we are interested, first , in the needs of the multimedia communications user.

Thus, we develop the seven layer protocol set from the top layers down. This is a drastically different course in the development and exposition of the OSI protocol set but it reflects the need in the multimedia environment to have the higher layer functions come to the fore.

The seven protocol layers are as follows:

Layer 7: Applications:

The applications layer provides for a wide variety of services that support the overall set of applications that may be running on the end users cpu. These applications can generally run on a stand alone basis with no need for the applications layer to even exist. It is necessary, though, when we desire to network the cpus together in some communicating fashion. A typical example of an applications layer function is the file server function that is an integral part of a local area network environment. In this applications service, a particular applications program, such as a word processor, any desire to obtain a file of letters that have been stored on the file server. The applications program then evokes the file server service and this places a request from the program, through the local area network to the target file server.

Other applications services are such functions as a mail service, a directory service for all the users , a virtual terminal service that allows users to integrate a variety of terminals on the network, and the graphics interface services that we had developed in Chapter 2. Thus the applications layer, is the highest layer that functions to allow for applications to take full advantage of the fact that they can communicate with a variety of other users or applications on the network.

Layer 6 Presentation:

This layer is probably the most improperly named for the purpose of this book. It does not deal with presentation in any way other then possibly how it "presents" data to the applications layer. It has been suggested by Tannenbaum that it be called the representation layer, and by others that it be called the data packing layer. It actually provides a useful function in how the data is to be packaged and repackaged as it comes and goes on the communications network. One service of the presentation layer is that of end to end encryption. This service takes the information that is generated by the end user in an application, and secures it through some encryption scheme that reduces the probability that some nefarious interceptor may capture the data for illegal purposes.

Layer 5 Session:

This layer is the one that the standards bodies have spent the least amount of time on but as we have seen in this book, represents the heart of a multimedia communications environment. The session is the construct that we have developed that allows many users, in a multimedia environment, to be assured of a seamless, error proof path that ensures the synchronicity of the communications across the path and the graceful interconnection and elimination of any set of users as required. In the multimedia environment, the session is an enduring dialogue between one or many human users and their interactions with applications programs and multimedia data files. Voice and video do not suffer the vagaries of a connectionless network well and require that there be a connection based entity to support their communications requirements. (see Kishino et al, Nomura et al, Verbiest and Pinnoo, and Kishimoto et al).

In these first three layers we have dealt with concepts that ensure that the higher level communications processes are properly handled. We have implicitly assumed that all other things have been handled in an appropriate fashion. The lower layers will worry about such things as the end to end integrity of the bits, the routing of the bits to the correct places and the physical interfaces being correct.

Layer 4 Transport:

The transport layer is the first "real" communications layer in that it relates to bits and bytes going across the network. It is concerned about the end to end communications of those packets of information, assuring that they get from point A to point B. It assumes that if it send a packet to the lower layers that they, on a one by one basis get where they are supposed to in an errorless fashion, but that this layer must be concerned with the whole set of packets generated by a source.

Layer 3 Network:

The network layer is an integral part of a network that has many parts going in many directions. This may be called a set of subnets. These subnets have to be controlled, and packets that rattle around in this environment must be managed carefully and assured that they get, on a packet to packet basis, from one network node to another.

Layer 2 Data Link:

The data link layer worries about bits getting from one local location to another in an errorless fashion. This layer provides for error detection and management and may also include the capability to correct for errors that occur on the channel. It assumes that the lower layer provides only a transmission path for bits but that the transmission path may have errors.

A significant part of the Data Link layer is the Media Access Control (MAC) sublayer. The MAC layer provides for the local control and access to a distributed and shared communicatiions facility. Typical amongst the MAC capabilities are local and metropolitan area networks.

Layer 1 Physical:

The physical layer includes the physical connection, the modulation scheme and the multiplexing or switching schemes that can be used. Many of the elements of the proposed ISDN (Integrated Services Digital Network) can be found at this layer. The common RS-232 interface is also an element of this layer.

These seven layers are shown in Figure 5.x. We can explain them in some further detail by considering a specific example and how each of the layers interacts and requires the services of the layers below. As we can see in Figure 5.x the basic transfer element is a data packet and that each of the layers adds a header onto the data packet to effect the services that are provided by that layer. This goes all the way down to the physical layer which is the layer that deals with bits.

Consider the example of a set of multimedia users that desire to communicate in a multimedia mail environment. Let us first assume that thy are using a specific application program to analyze a particular image on the screen. This may be the case of a single user at a single location. That user now desires to share that image with several other users and comment on the results in some multimedia fashion.

o We generate a data block that is the multimedia image element. This is the first step in the ultimate process. In Figure 5.x, the data block is in actuality a complex multimedia data object, encompassing video, voice and still image. At the higher layers, the notation of a data block is merely symbolic of a large data element that is to be transmitted across the complex network.

o We then invoke the mail service at the applications layer and append to the data block that we wish sent a mail header. This header will contain all the elements that are necessary to evoke the mail function and to ensure that the system sends this total image from one location to all other users in the system. I will further assume that the act of mailing is such that the image being sent is a compound multimedia object, consisting of video, voice and still image, and thus we must have a fully synchronized session based service in place. We evoke the mail service in an applications layer program that provides for electronic mail. It may have also been possible that this mail program may have been resident at another location on another processor. We shall deal with these options at another time.

o Since this is a highly proprietary message, we will need to encrypt this message and this service of the presentation layer is invoked and it is represented by the header that is appended on the applications header packet at the presentation layer. This header is a symbolic method of demonstrating the process of encryption. We evoke the encryption at a point in the system called the data encrypting algorithm, which may be a stand alone piece of secure hardware in the communications system.

o At the session layer we append a header to the encrypted message to send to a session manager processor. This processor may be a network based element that handles all of the control for the network. This element sends out a set of commands to ensure that a temporary session is established and that all of the session participants are brought onto the session connection. The session manager then continues to clear the channel and to assure that the ultimate block of data associated with this multimedia message is properly handled as a complex multimedia object, and not just a a computer record which can be packetized in any random fashion.

o The transport layer will now take this packaged and session secure pact and it is required to get it from one point to another, and that is all. It will assume that the data channels are errorless and that they are being routed to the proper locations.

The transport layer takes the session packet and may break it up into smaller packets for transmission and further takes instructions from the session layer so as to ensure the synchroniocity of the total packet as it moves across the network. The transport layer control is located in a transport protocol control device that may be part of the computer front end or as a stand alone device.

o The network layer is needed because we are transmitting this broken up,packets into a network composed of many smaller switching elements. The network layer functions at each of these nodes in such a way to route the packets in the most efficient form from one point to another. This layer focuses on getting the smaller packets across the disparate network.

o The data link layer deals with the error control protocol that ensures that the data is kept error free. In this case we assume that the DLC protocol is an HDLC (High Level Data Link Control) protocol that allows for acknowledgments and error detection. Error correction is not provided on transmitted packet and is achieved only by packet retransmission.

o Finally, the physical layer performs this task with an RS-232 interface to my computer using a simple on-off 9.6 Kbps modem. This is the easiest layer to understands.

We cal view the message at the top most layer as M and then view the actions of each layer as Lk where k represents the layer. At the top most layer we have:

$$R7 = L7\ M$$

where R7 is the layer 7 representation of the message M that is a result of layer seven operator L7. Thus the mail function may operate by L7 on the message M to generate R7.

In general we can say that;

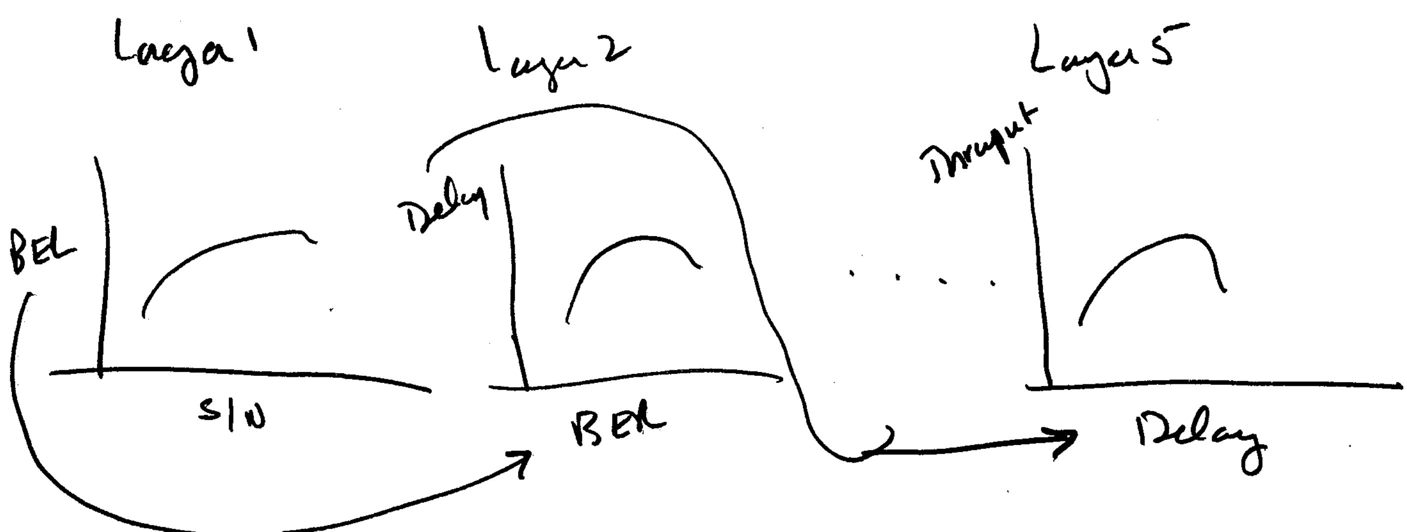
$$Rk+1 = Lk+1\ Rk$$

$$Rk+1 = Lk+1\ Lk\ \ldots\ L7\ M$$

For example, the operator L1 is a modulation operator on R6. Thus R7 is a modulated waveform with specific characteristics.

Figure 5.x OSI Layer Architecture

Also want Perf/Sizing Examples as we move up layers



Show how they relate.

There are many issues that arise in the discussion of the OSI protocol stacks. Some of them are as follows;

o Addressing: How do we know how to get from one point to another in a network. Addresses are necessary at all layers of the protocol.

o Each layer has elements, we shall call them entities, that are agents for performing tasks at those layers. The RS-232 interface or the modem are physical entities, and the mail process is an applications entity. We must be able to identify entities at all of the layers.

o Active agents are necessary at the layers to effect the operations of the services provided at those layers. These are called service access point (SAP) and these are the points that effect the service operation. We can usually envision these being at the interface between the two layers.

o There must be an agreed to format and set of rules for the layers to exchange the data downward. This format is called the interface data unit (IDU).

o The IDU consists of a set of control information called the interface control information (ICI) and the actual information to be passed along the network, called the service data unit (SDU).

o There must be a set of calls or instructions that allow for communications at the same levels in different units in the network. We shall call these primitives and they have their own syntax. The syntax of the communication language primitives fall into a form :

```
PRIMITIVE = LAYER.Service_Type.action(parameters)
```

For example, at the session layer we may be asking the session SAP to create a session and then to add several people to the session. The primitive may take the form:

```
Session.Create_Session.request(p1,p2,....,pn)
```

This sends a request to the session processor that a session creation is desired. We can further extend the syntax by including parameter within the primitive to delineate the details of the primitive request. For example:

```
SESSION.Session_Create.request(source_address,destination_address)
```

Thus the general form for a primitive will be;

```
LAYER.Service_Type.action(u1,u2,....,un)
```

In Figure 5.x we show the elements of the layers and show how this primitive form of interaction within layers functions

Figure 5.x OSI Layer Entities

There are four types of actions that can be implemented in the different protocol layers. These actions are;

o request: asks an entity to do something

o indication: entity is informed of a request

o response: entity desires to respond

o confirm: entity is made aware of request response

The type of Service_Types may vary by layer.

We shall be spending significant effort on the the development of the primitives, especially the Service_Types. These are extremely important at the higher layers for the implementation of the services that can be implemented in these types of networks. We have already discussed several primitive concepts in Chapters 2 and 3 in both the graphics and windowing areas already. The development of these primitives in the communications area will merely be an extension of that effort.

In Figure 5.x we show how the actions are presented across layer interfaces. We can envision the two entities in a communications mode, one being the Initiator and the other the Responder. There are two layers of protocol involved. At the Initiator, the user at the higher layer send a request action to their lower layer. This goes across the network and becomes an indication passed up from the lower layer (Provider) to the comparable layer (User) who then responds. This response is called the response and it goes down and up and it results in a confirm from the Initiators Provider layer to the User layer (see Schwartz p. 88).

Figure 5.x Primitive actions in interlayer Communications

(Ref Schwartz Fig 3-8 p88)

5.2.1 Applications Layer

The applications layer is the highest layer and it is structured to support the end user applications as well as provide an environment for the applications to reside in. The applications layer's functions, as we shall describe shortly, focus on direct support of the applications programs. The applications programs may have access to the applications layer services through the primitive functions that we discussed above. It is through these primitives that we obtain access form one layer service to another.

We ma view the applications layer services as if they were adjuncts to the operating system, in that we frequently call the operating system services from pure applications program.

5.2.1.1 Functions

The applications layer functions both as a service provider as well as a port to the lower layers of the OSI stack. In its function as a service entity, it directly provides services that utilize the network resources that are available to it. Thus, the applications layer can provide such network based services as;

- o Mail
- o File Server
- o Directory

The applications layer may also provide for locally resident services that may be of general use to the end user. These types of services are those of the graphics user interface, terminal emulation services and other specific translations of generic to specific nature. We shall focus on the functionality of the applications layer the arelates to its use of the network resources.

5.2.1.2 Protocols and Architectures

5.2.1.3 Performance Issues

5.2.1.4 Alternatives

5.2.1.5 Implications

5.2.2 Presentation Layer

The presentation layer is in actuality a data re-presentation layer. It takes the data that is provided by the session layer or provided to the session layer and performs certain tasks to re present the data. For example, the presentation layer takes a letter that may be generated in the mail service at the applications layer and encrypts the letter using an encryption service. The service layer may also take an image that is to be sent at the applications layer and may compress the image using a data compression algorithm of the types that we have discussed in Chapter 2. This layer may also provide a means for changing from one data format to another, especially in terms of how the data is to be presented to the applications layer.

The presentation layer does not deal with presentation of information to the end user. That, ironically, is dealt with in the applications layer as an applications service. The presentation layer deals with the presentation of data to the applications layer and to the session layer. It focuses on how to change the data from and into forms that meet the overall communications requirements. Also recall that the data or information at the higher layers is not just a block of data it is the entire message.

5.2.2.1 Functions

The presentation layer has several major functions as we have been discussing them. Tannenbaum has presented them as follows;

o Provides an access mechanism to the session level services.

o Provides a common mechanism for the definition and manipulation of complex data structures.

o Provides a management facility for the management of the data structures that are currently in use through the applications or session layer services.

o Providing a means to convert a data element from one form to another, on either transmit or receive, for the purposes of more effective or secure communications.

We can see that these types of functions become more complex when we deal with the multimedia environment. Recall, that in the multimedia environment that the session layer tasks on a more significant role and that this role requires that the images be handled in a more robust fashion. The present conceptual structure of the presentation layer deals with data structures that have little time variation. That is they are considered give as a block of data in their entirety and do not have the synchronization problems that we face in video or voice communications. We can envision, possible, the use of presentation layer services that can be used to created the multimedia data elements, and to provide these to the session layer.

Consider, for example, that there is an applications layer program that generates a complex multimedia object. Let us call that object DOk. We let DOk be given by;

$$DOk = \{DOVk, DOIk, DODk, DOTk\}$$

where these sub elements are those for voice, image, video and text respectively. We may envision that the presentation layer has the responsibility to handle the manipulation and integration of this multimedia data object. We have seen in Chapter 4 how we can generate this object in a single data file and we shall further see in Chapter 8 how we can handle these elements in distributed multimedia database. We see in the presentation layer a functionality that allows us to manipulate these data elements across the network so that we have a mechanism to maintain their total integrity.

5.2.2.2 Protocols and Architectures

5.2.2.3 Performance Issues

5.2.2.4 Alternatives

5.2.2.5 Implications

5.2.3 Session Layer

The session is the heart of the multimedia communication environment. All that the layers below the session layer do is guarantee that the data packets have gotten from one point to all of the other points that are in the session. They do not guarantee that they got there on time or in most cases even in order. In addition, these lower layers of the protocol sets only view the data transported as a single block, even if it is a created complex multimedia data object. However, the data object may be a sequenced element of an extended session between many users and the context that that block of data finds itself in is as important as the block in and of itself. The session layer is structured to deal with communication "in context". For the most part, it is the only layer that deals with the interactive dialogue nature of conversation environments as a natural part of it functional and service elements.

If we have a error free fiber connection between each and every user, it is conceivable that there is no need for the lower four layers and the only critical layer is that which maintains the session. Ironically, the session layer was not considered as an important layer in the early days of the ISO structures. Tannenbaum discusses the fact that the British had originally proposed that the session layer was not at all needed. As we shall see in this section, the session layer provides some of the most valuable functions from the perspective of a multi user multimedia environment. All the session layer assumes from the services below it is an error free path for the messages that is sends to those layers. All it expects from the layers above is a continual flow of messages that are to be forwarded to those users. The session layer takes the responsibility of managing the interuser conversation, not just the communications.

5.2.3.1 Functions

The functions of the session layer relate to the maintenance and support of the session services that we have just discussed. Some of these functions are as follows:

o Dialog Management: This provides for the control over who is is control of the session at any one time. We can view the management of a dialogue handled in a variety of forms. The baton approach is based upon a mutual agreement between the session participants that the holder of the baton is in charge and may exercise direct control. The baton may be passed back and forth amongst the session members. Another approach is a priority based session dialogue manager which assigns a priority level to each of the session participants. We shall discuss these and other latter in this section. There are performance issue that relate to deadlock of dialogue management and the delay in session control.

o Session Establishment: Session establishment allows for one user to initiate a session and to add other users onto the session in an operational context. The session establishment function allows for the identification and addressing of other virtual users and ensures that they are bound to these session. It also allows for the disestablishment of sessions. We frequently are concerned about the session establishment time as we are concerned about call set up time for a telephone call. Thus in terms of the overall performance of the session establishment function,we look at determining the overall session establishment set up time budget and allocate it to the various system resources used.

o Connection Based Service: A session is synonymous with a session. A connection based service is one that establishes a permanent virtual circuit that the session is built upon. The establishment, maintenance and disestablishment of a session is one of the functions of the session layer. The session is built upon the session layer created connection based circuit, that is the permanent virtual circuit that the session layer creates with the transport layer.

o Synchronization: When we are dealing in multimedia communications, we are dealing with information that is not only contiguous in terms of spatial relationships but also has a temporal relationship that is critical to its presentation, display and interaction. We may think of what may happen to a video display using an NTSC scan line system if instead of 525 lines being transmitted we get only 524. This would propagate down to the bottom of the system and result in a totally confused interaction. Errors and transport faults do occur and the timing of messages across the network may face significant delays and temporal variances. It is the function f the synchronization service of the session layer to keep all of these complex messages in a truly synchronous mode. The suynchronization service performs this task by inserting various synchronization ticks in the data stream and recovering them at the other locations in the session.

o Activity Management: A session is an enduring communications between two or more entities in the communications netwrk. There are moments when the actual communications in the session may be reduced to a low to zero level. We can envision breaking up a session into elements that represent the burst of active interaction between the users and during those times allocate services of the lower layers. We will always want to have the session endure, since the endurance is a quality of the session. However, it is a function of the session layer to recognize the dormant phases that may occur in a session and to allocate resources accordingly. We do this by creating or defining elements of the total session called activities that are the periods of the active participation of the users of the session and during those periods allocates the resources of the system. This session service allows for the optimum use of the communications resources at the lower layers and assists in the optimization of the shared switched services for transport.

o Exception Reporting:

5.2.3.2 Protocols and Architectures

The session layer protocols are similar to those at the layers above but focus on the implementation of the services described in the functions that we presented in the preceding subsection. Recall that the syntax for is as follows:

LAYER.Service_Type.activity

Here we have SESSION for the LAYERS and we still have the four possible activities as:

- o request
- o indication
- o response
- o confirm

These actions relate to the development and specification of the protocol to effect the service. The Service is the indication of which of the services that is being invoked. These Services are:

- o Session
- o Connection
- o Dialog
- o Activity
- o Synchronization
- o Report

The Type sub-indicator may tell that it is the beginning, end, discard, resume or interrupt of an activity service. The session Type delimiter provides additional specificity to the command.

Recall also that part of the syntax are specifiers to the command syntax the indicate the specifics of the session elements. We have detailed several detailed primitives that are used in an actual multimedia system in Table 5.x.

Table 5.x Session Layer Primitive Syntax Examples

(Insert MEDOS Examples)

5.2.3.3 Performance Issues

5.2.3.4 Alternatives

Let us consider several implementations of the session layer functionality. The first deal with those of dialogue management and then we consider synchronization and activity management as well as exception reporting.

In dialogue management, the major function is to develop algorithms that allow for the management and control of sessions. This management and control can be handled from the point of view of a token or baton or from a wide variety of other techniques. Let us consider four that are common.

o Hierarchical: In this case a session manager in control at all times and all requests for the baton are passed up to the session manager. The requests are handled in some priority fashion based on equal user priority and first come first served to some other set of priorities. It is a non distributed control algorithm and requires that each session select and adhere t the single point of control.

o Baton Passing: This is a round robin approach where the baton is held by a user for a fixed period of time if they need it otherwise it is passed to another users in a round robin fashion. each users is allowed a maximum period to maintain session control and at the end of the period must relinquish it. It may be relinquished earlier.

o Priority: In this case, all of the users have a priority, and no two users have the same priority. The priority may be a combination of several factors. Let P(k) be the priority of user k, and we define this as:

$$P(k) = R(k) + T(k) + D(k) + I(k)$$

where ;

R is the hierarchical level of rank of k

T is the length of time since k last controlled the session

D is the data waiting in k's buffer

I is the image intensity of k's session

This is a totally arbitrary priority scheme but any other priority function may be generated. Th scheme works by all users sending out their priority to the network and the users comparing theirs to others and responding accordingly. The only users to respond is the one with the greatest priority.

o Interruption Control: This scheme is a first come first served, shouts the loudest protocol. It is based upon a users sending a broadcast message out to all users demand access and only when this users messages are received first by all others is this session accepted.

5.2.3.5 Implications

5.2.4 Transport Layer

The transport layer provides for end to end integrity of the communications path. It assumes that there is an errorless and routed infrastructure in the lower layers and ensures that the message that is to be transmitted is done so errorless in toto. For example, the transport layer will receive a mail message from the higher layers and its takes the entire mail message, partitions it in a fashion to meet the limitations of the transmission paths, and sequences smaller packets across the network to ensure that the total mail piece is sent in an errorless fashion. It is important to note, however, that though this performance may suffice for a computer oriented system, it is not sufficient for the multimedia environment developed in this book. The transport layer allows for the transport on an end to end basis of a message. The session layer ensures the endurance of the conversational mode that is the essence of the multimedia environment. The session layers assumes that the transport layer exists. The transport layer makes similar assumptions of the layers below it.

5.2.4.1 Functions

Transport layer services can be of the connection based type of the connectionless type. The connection based type transport services follow the requirements that we placed on them in the session layer. In this case the transport layer may either support the session layer or may even supplant it in the session connection service. The connectionless type transport service is merely the ability to provide for a datagram service, ensuring that a single package is transmitted in and error free fashion from one point to another.

Recall that at the transport layer the services are provided to it from the Network SAP and are provided to the session layer from the T SAP. The actual hardware and software that performs the tasks in the transport layer are at the Transport Entity.

The typical functions of the transport layer are:

o Connection Establishment: This function provides for the calls down to the lower levels for the establishment of a connection between another user on the network. It is a connection which may also be a permanent virtual circuit and in some ways may emulate the session level connection service. If the transport layer can support a full connection service, then the session service is simply provided by a call to the transport service.

o Addressing: Addressing at the transport layer provides for the naming and identification of specific point to be communicated with in the total network. At the transport layer, the addressing function allows for specific identity of the physical and logical location of the virtual user.

o Connection Disestablishment: This service is the termination of the connection established in the connection establishment.

o Connection Management: The management of a connection is necessary both from the perspective of maintaining the connection as well as providing reports on the status of the connection as it progresses in time. This service allows for interface for the overall network management of the services.

o Flow Control and Buffering: The information handed to the transport layer from the higher layers is always the total message and not a fragmented form that is suitable for communications. The transmission paths generally do not support the transport of large blocks of data because the errors inherent in raw transmission would never permit an effective communications system. Thus the communications path uses some for of smaller packets, usually a small percentage of the total information package. Thus the transport layer must accommodate the buffering and segmentation of the total information element and control the flow of the packetized versions across the network.

o Recovery: Recovery is a key service at the transport level and it is used as an element of the overall session service. Recovery provides for the restoration and reconstitution of the communications path is there is a fatal failure that causes the collapse of the path. We generally assume that this function is at the transport layer since it has full control over the total network elements.

o Level of Service (Quality of Service): The transport level allows for the attachment of level, grades or qualities of service to the established communications path. These grades of service are related to the performance factors that we shall be discussing in the latter parts of this chapter.

o Multiplexing: The transport layer can provide a service that allows for multiple circuits on the same channel by the proper identification of sources and sinks in the network. We shall expand upon this latter.

These services can be created in a fashion similar to those of the higher layers. In the next subsection, we shall focus on some of the specific primitives and how they are architects to provide the transport layer functions or services.

5.2.4.2 Protocols and Architectures

5.2.4.3 Performance Issues

5.2.4.4 Alternatives

5.2.4.5 Implications

5.2.5 Network Layer

This layer provides for the point to point switching and routing of packets through the network. The services that the data link layer provides is an errorless point to point path. The network layer has to provide the transport layer with a clear path from the beginning to the end of the transmission medium.

5.2.5.1 Functions

The functions of the network layer are as follows:

- o Routing
- o Congestion and Flow Control
- o Internetworking

5.2.5.2 Protocols and Architectures

5.2.5.3 Performance Issues

5.2.5.4 Alternatives

5.2.5.5 Implications

5.2.6 Data Link Layer

The data link layer has traditionally had the responsibility of providing an error free transport on a point to point basis. It assumes that there is an underlying physical connection, but that the connection may be error prone. It further assumes that the connection has been made correctly in some gross sense by commands that have been sent down from possible higher layers in the protocol stack.

There is also a second subfunction that is now incorporated in the DLC layer and it is in the Media Access Control (MAC) sublayer. This layer provides for interconnection of physical devices in an errorless fashion on a distributed network such as a local area network or a metropolitan area network.

5.2.6.1 Functions

The functions provided by the Data Link layer are:

- o Framing
- o Error Control
- o Flow Control
- o Link Management

The MAC functions provided are;

- o Framing
- o Error Control
- o Flow management
- o Switching
- o Link Management

5.2.6.2 Protocols and Architectures

5.2.6.3 Performance Issues

5.2.6.4 Alternatives

5.2.6.5 Implications

5.2.7 Physical Layer

the physical layer describes the interfaces that the computer communications system has with the real world of the communications network. It discusses the physical interfaces, the modulation schemes and signal levels, the data rates and coding schemes, and even the switching and multiplexing schemes. We shall not focus a great deal on this layer in this book since there is a wealth of literature on this layer elsewhere. The reader is referred to such books as Tannenbaum, Schwartz, and others.

5.2.7.1 Functions

There are several functions of the physical layer. They are;

- o Physical Interface
- o Mechanical to Electrical Conversion
- o Switching
- o Transmission
- o Modulation
- o Multiplexing

5.2.7.2 Protocols and Architectures

5.2.7.3 Performance Issues

5.2.7.4 Alternatives

5.2.7.5 Implications

5.3 Broadband Alternatives

The most pressing requirement for the communications system in order to ensure the ubiquity of the service that we are developing in this book are the elements that relate to the transport of data in very high data rates, These rates are in the regions of tens of Mbps to the higher Gbps range. These rates are now achievable with the introduction of fiber technology, fibers that are made of glass filaments. These fibers have been developed and perfected in the past ten years and now are at the stage that they are both ubiquitously deployed and in addition various implementations are available for implementation.

In this section we discuss three of the most common alternatives for the deployment of high Mbps data channels, ones that also have a potential for evolving into Gbps channels. From a technical point of view, a single strand of fiber can theoretically support data rates in the Tbps range, that is 10^{12} bits per second. However, the technology to interface this data rate to the network is not yet available in commercial form. We shall not discuss these factors in this chapter but leave this to the reader. The texts by Personick, Kaiser and others provide significant detail on the fiber technology and the status of the opto electronic interfaces.

The choice of the three techniques discussed in this section is based upon the facts that they are either available of will be currently available in the near future. FDDI is a presently available system that supports 100 Mbps data channels and is being deployed for intra premise network systems. It frequently works on multimode fiber systems and can be interconnected to inter premise systems. SMDS is an embodiment of a metropolitan area network architecture that is being planned for shared switched network deployment. It works at the 45 Mbps rate and can go to the 150 Mbps range. It is a dual bus architecture and can support a shared switched network operation. Finally, ATM is the embodiment of the Broadband ISDN system and is farther off in implementation.

5.3.1 Broadband Issues

Figure 5.x Taxonomy of Broadband Networks (Daddis & Torng F1)

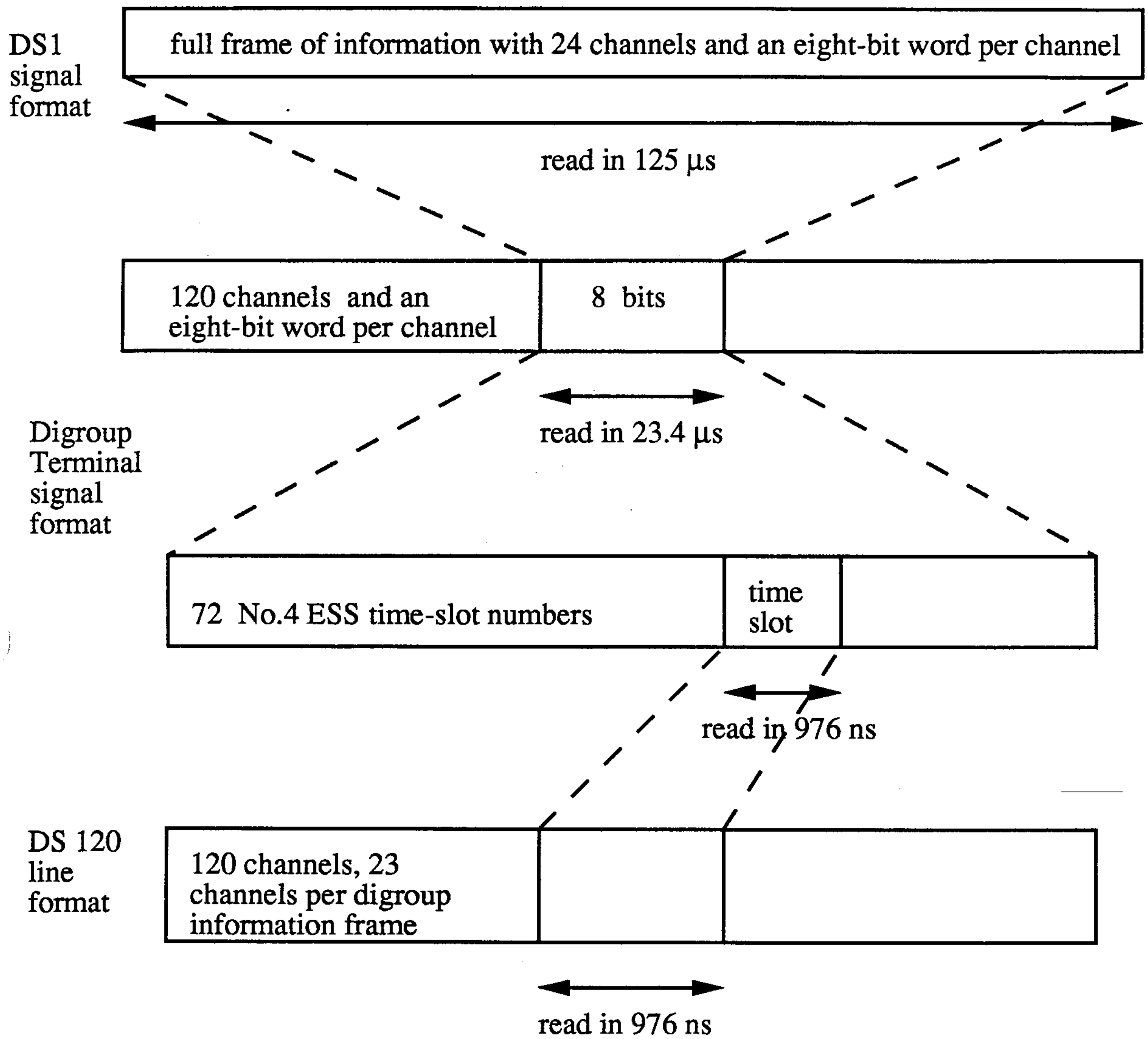


Figure 5.10.a : Broadband Switches - Example : Digital Switching Interface for the Digroup Terminal Used in no.4 ESS.

Frequency Division Multiplexing

Time Division Multiplexing

Code Division Multiplexing

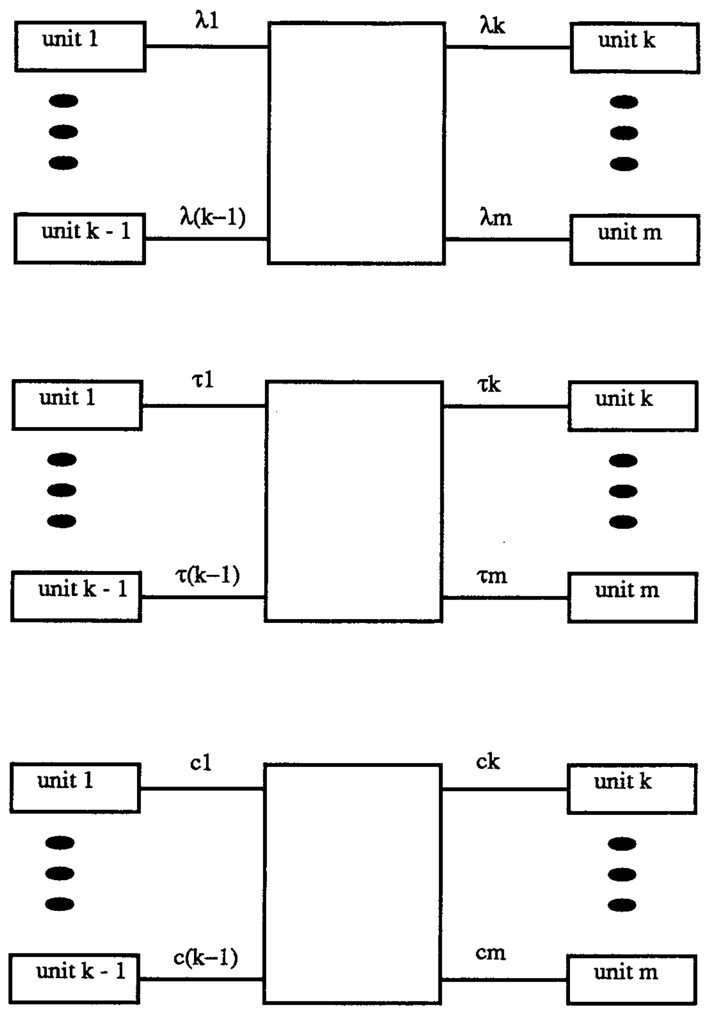


Figure 5.10.b : Broadband Switching - Distributed Switching.

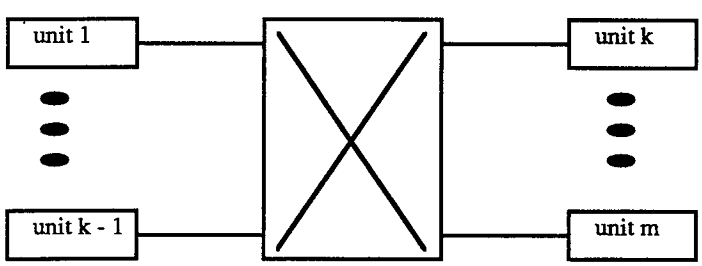


Figure 5.10.c : Broadband Switching - Centralized Circuit Switching.

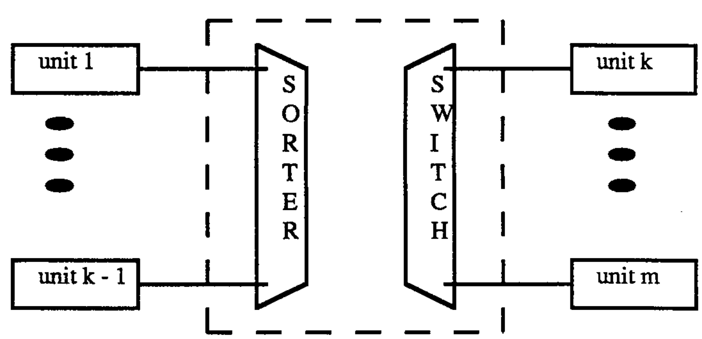


Figure 5.10.d : Broadband Switching - Centralized Packet Switching.

5.3.2 FDDI

We start first with a broadband system that is presently in operation and provides for 100 Mbps capability in internetting local area networks. The FDDI(Fiber Distributed Data Interfaces) systems are typically used in intra facility networks although they are not limited to these areas. The generally are also used in an non switched and non shared environment, although changes are being developed to permit them further capability.

The FDDI systems were developed to meet the need of the users of LAN systems and to provide the capability to allow these system to be interconnected at higher speeds. Initially, the LAN to LAN interfaces were at considerable lower speeds and thus made for slow file access and transfer time on these types of networks. recognizing the needs for expanded interfaces speeds the use of 100 Mbps was the desired bit rate on the interconnect network. There was, however, no thought originally given to the implementation or operational issue of a multimedia environment. We shall however show how this can be incorporated into this environment.

The architecture of the FDDI system is presented in Figure 5.x. The system is built upon a dual ring bus structure that uses a fiber optic cable. The fiber cable is frequently a multimode fiber and the bus architecture allows for graceful degradation of the system in the event a single node or a fiber failure occurs. The ring architecture has been chosen in place of the other forms because of the inherent reliability of the ring in the event of a single point failure. Each node has a repeater function as well as the capability to detect the FDDI packets as they are transmitted around the network.

The IEEE 802.5 MAC protocol is used on the ring for the control of the access to multiple users. This protocol is a token passing scheme that passes a token around the network that allows the users to grab the token in the event that they have data to transmit or otherwise to pass the token to other users. As we discussed in the previous section, on the topic of the 802 standards, this particular IEEE standard allows for control on ring structures. The other protocols are generally found on bus structures.

Figure 5.x FDDI Network Architecture (Ross Fig 3)

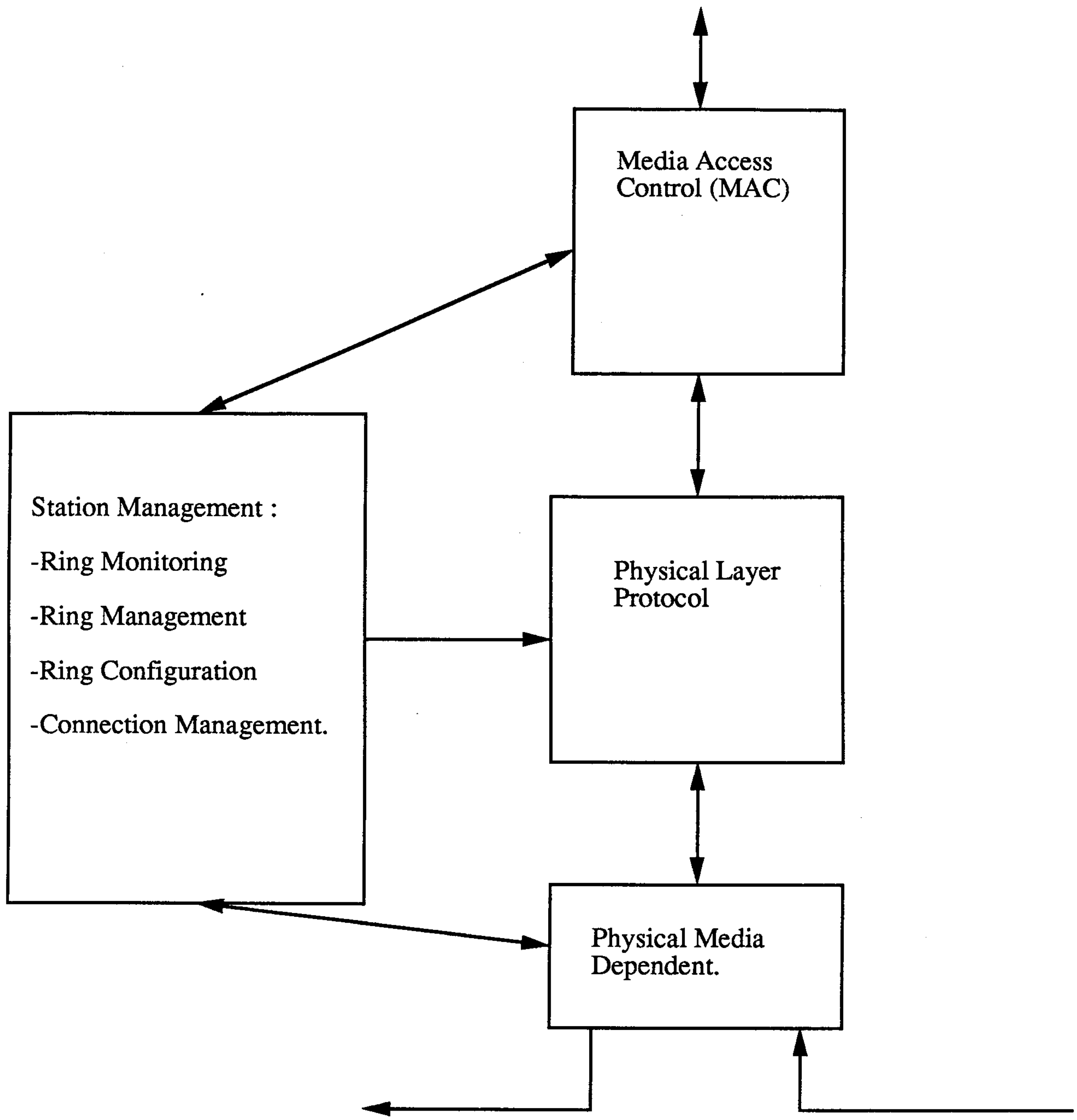


Figure 5.17 : FDDI Interfacing.

The 802.5 frame format is presented in Figure 5.x It is composed of the following elements;

o Preamble: This is a set of bits of length of at least 16 symbols to initiate a response at the local node.

o Starting Delimiter: This block provides a signaling bit frame that allows for the synchronization acquisition.

o Access Control: This frame consists of four sub-frames;

- o Priority Bits: Yields the level of priority of this packet
- o Token Bit: This indicates whether this is data or token
- o Monitor Bit: Prevents continuous token circulation
- o Reservation Bit: Allows for priority queuing

o Frame Control: This states the type of frame, whether it is synchronous or asynch, the length of address fields, and the kind of frame. It also provides for a capability to have two types of tokens that allow for limited discussions between closed set of users.

o Destination Address: This may be between 16 and 48 bits in length.

o Source Address: Similar to the Destination

o Data

o Frame Check Sequence: This is a cyclic redundancy code that provides for error detection.

o Ending Delimiter: This s simple delimiter symbol to indicate the limitation f a frame.

o Frame Status: This consists of three control indicator symbols that can be changed by a terminal as it handles the frame. It allows for the communications of errors from different stations.

Figure 5.x FDDI Frame Format (Bux Fig 6)

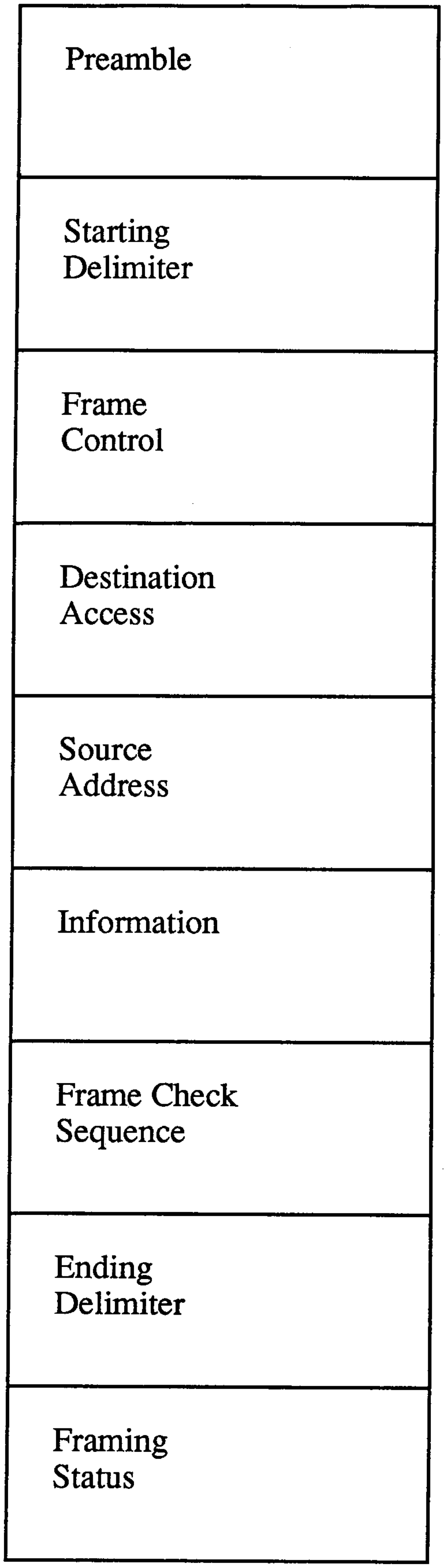


Figure 5.16 : FDDI Frame Format.

The operation of the 802.5 MAC protocol is shown in Figure 5.x. We can see how the protocol works by viewing the process under two circumstances.

o A Station Has a Token and is Transmitting

Assume that A has obtained a token and that it desires to transmit to station N. A creates the packet in the framing format and places it on the network. It is handled by each of the intermediate stations and the destination address in read by each. When it reaches N, the station recognizes its packet and translates the entire message. It will in addition, enter a data change in the message an indicate whether it has been received or not properly and retransmit this change message around the circuit.

The message will continue to transmit around the ring, and when received by the original station, it will read the message and will recognize if the message was successfully transmitted or not. If not a retransmission may be performed if it has not been timed out.

Let us first consider the implications of the circulation of this single token. The speed of propagation of a signal in a fiber is approximately 200m/microsec and the free space speed of light is 300 m/microsecond. Thus the speed of propagation of about 0.66 that of free space. Now consider a ring message packet of 100 bits and consider an FDDI ring of 1,000 m. The ring propagation time for a packet to go around the entire ring, assuming that there are no delays anywhere, is 5 microseconds. If the message is 100 bits, then in order for the bits not to overlap the data rate must be greater than 100 bits/5 microseconds or 20 Mbps. We can generalize this by saying that, if;

R = Data rate on the channel (Mbps)

B = Number of bits per message (bits)

L = Length of the Ring (m)

Then;

$$R > B / (L/200)$$

or

$$R > 200B/L$$

For an FDDI rate of 100 Mbps, and a message length of 1,000 bits, we have;

$$L > 200B/R = 200\ (1{,}000/100) = 2{,}000 \text{ m}$$

That is we need a 2 km ring at a minimum.

We can envision the ring functioning as follows;

More detail

How is it viewed by a single user?

o Assume that there is no messages to be sent. We circulate a packet of the form;

SD:AC:ED (3 bytes, one byte each)

which is two delimiters and the access control (eg the token bit). It circulates until one of the stations seizes control and changes the token bit (AC).

o When the token bit is changed, that station now can send data for a period as long as THT (token holding time) seconds, usually 10 msec, and then it must release the token.

o When a station desires to seize the next token, it does so by looking at the data message being sent around the ring and when it wants to seize the token, it enters a message in the AC byte to indicate its priority. If there is a message there already, it waits. If there is a request there already but it has a higher priority, then it may be able to bump the other request.

o When the station stops due either to time out of data finished, the token is passed to the requesting station if there is one or a blank token request starts circulating again.

Figure 5.x FDDI 802.5 Protocol (Bux Fig 7)

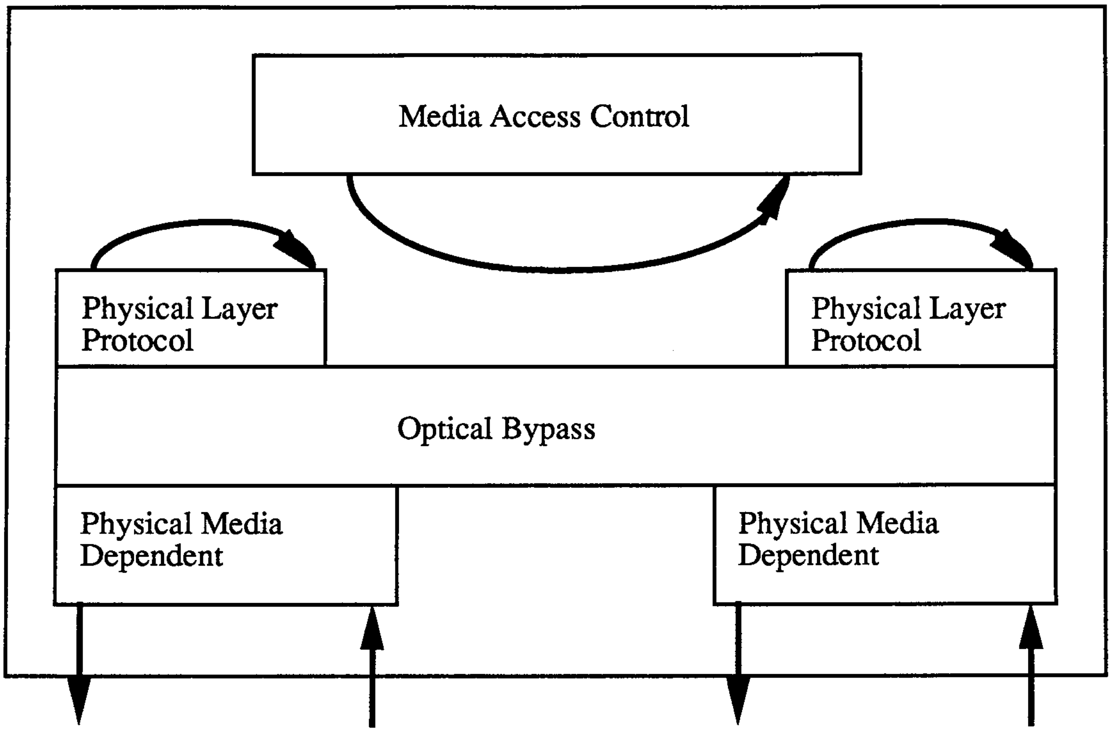


Figure 5.19.a : FDDI Interface Architecture - Optically Bypassed Dual-Attachment Station.

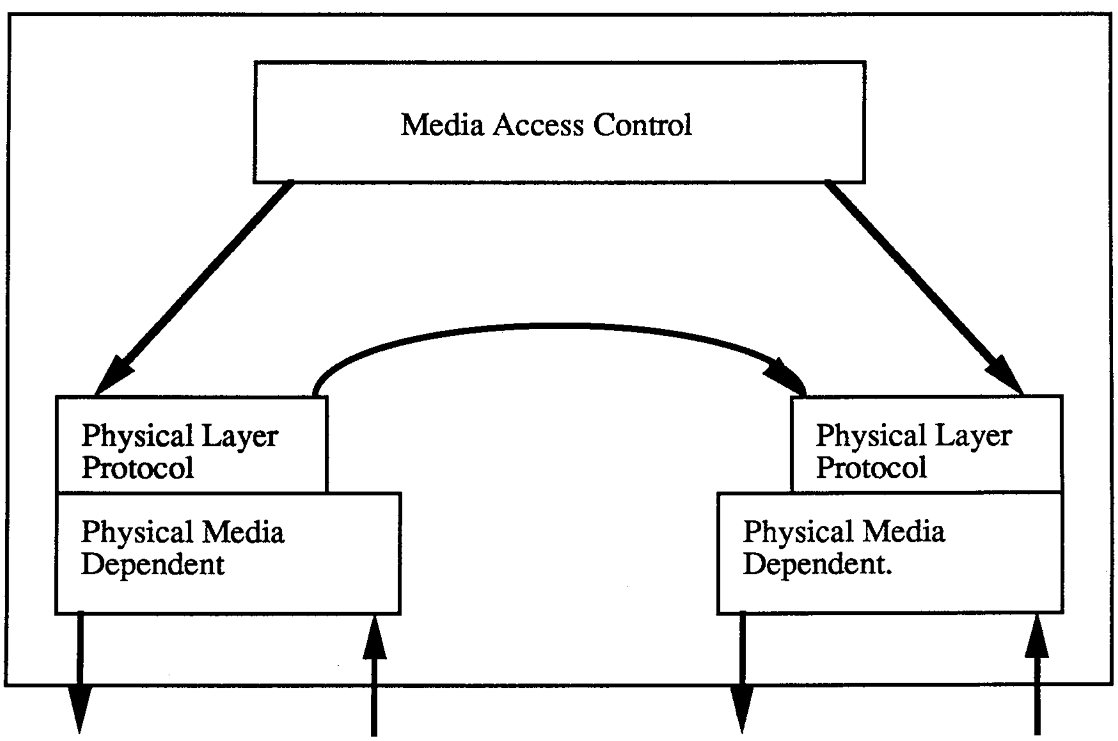


Figure 5.19.b : FDDI Interface Architecture - One MAC Dual-Attachment Class A Station.

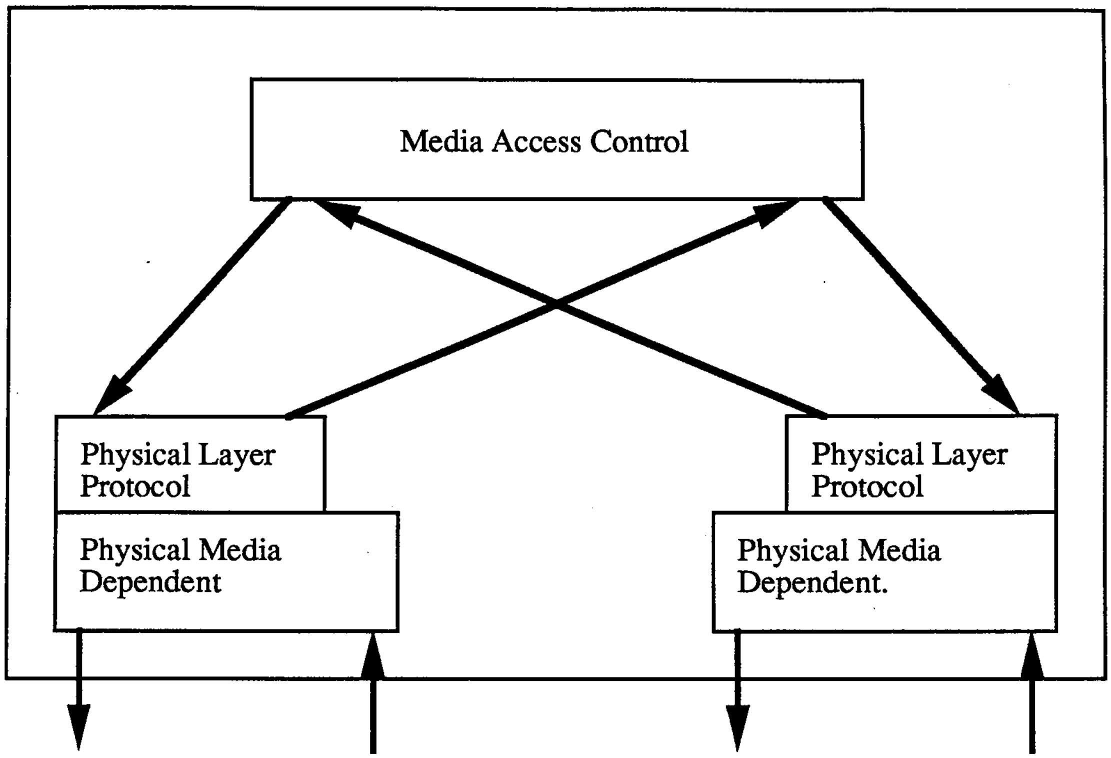


Figure 5.19.c : FDDI Interface Architecture - One MAC Dual-Attachment Class A Station.

The performance of the FDDI networks has been studied in some detail and we present the result outlined in Bux. Figure 5.x depicts some of the analysis of this protocol for two different channel rates and for the presentation of the mean transfer delay time as a function of the information throughput. Several observations are clear;

o The greater the length of the cable the greater the delay.

o The greater the length of the packet the greater the delay.

o There is an instability that occurs in the delay as the throughput attempts to reach the data rate of the bus. Thus the maximum throughput is less than the maximum bus rate as is expected.

Figure 5.x FDDI Performance (Bux Fig 14)

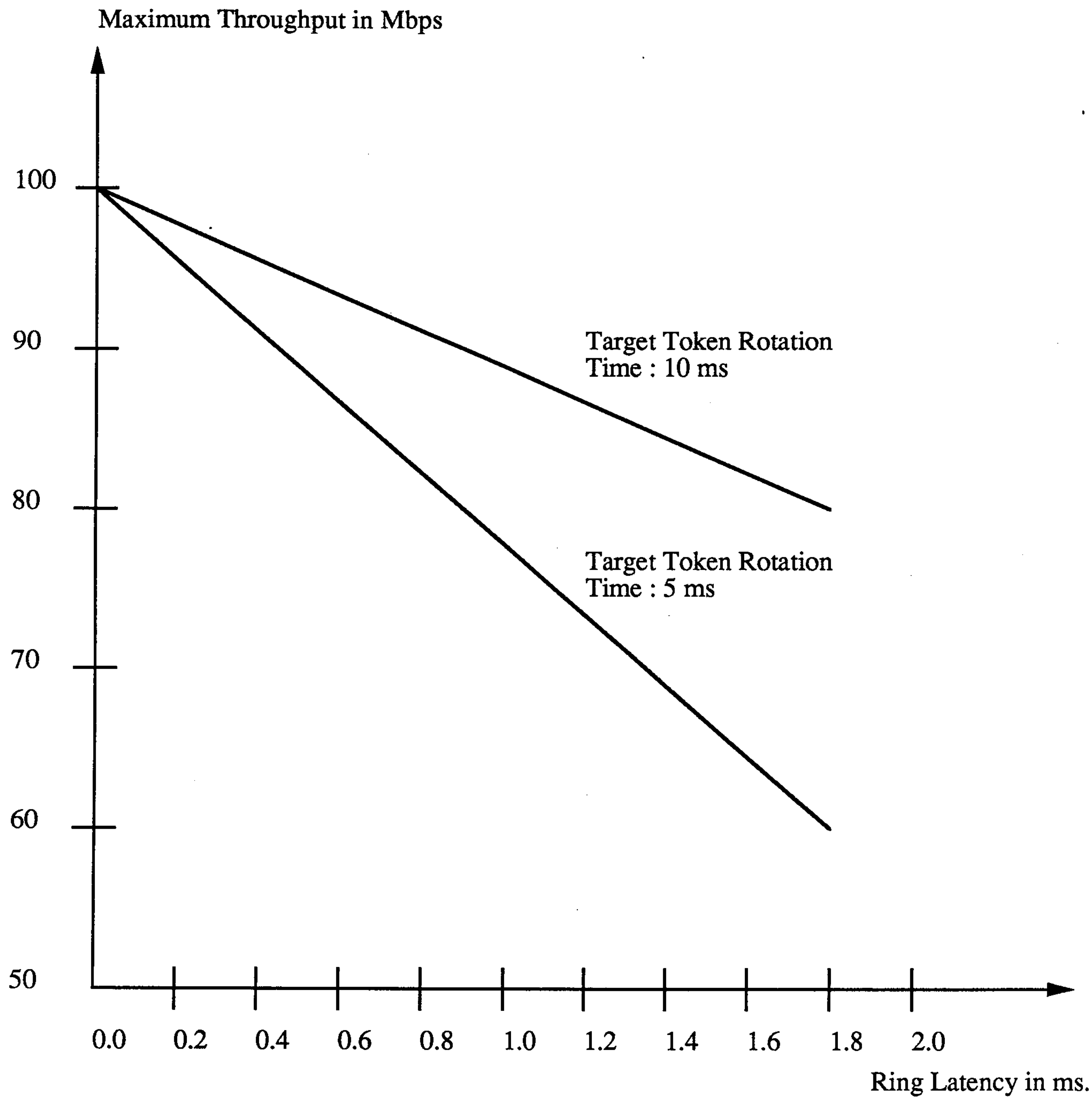


Figure 5.20 : FDDI Throughput and Performance - Maximum Throughput vs. Ring Latency with 10 Active Stations.

FDDI applications are quite appropriate for intrafacility locations or those involving shorter distances. As the distances increase we have seen that the average delay increases proportionately and that this is an inherent element of the token ring structure.

5.3.3 SMDS

This represents a second step in the development of a broadband capability. SMDS, switched multimegabit data services, is a shared switched metropolitan area network (MAN) that is being proposed for implementation in local transport areas. It is built around the 45 Mbps standard.

The architecture for the SMDS system is shown in Figure 5.x. It is comprised of a dual bus structure, not a closed ring, and the forward part of the bus transmits packets with data plus a bit that is called the busy bit. The busy bit indicates if the packet is empty or has data. At each node or station on the SMDS network, the packet is decoded and the busy bit is read. We shall discuss the status of the busy bit shortly.

The SMDS system is a connectionless datagram approach to higher speed communications networks. The network designs are developed for the use in metropolitan area networks (MAN).

There is a second or reverse bus in the dual bus scheme. The reverse bus transmits a packet containing no data but it does contain a request bit (REQ). This bit is generated by any one of the stations upwards of the receiving station. The bit is set to one by the station desiring to transmit.

The stations read both the forward and the backward bits.

Figure 5.x SMDS DBDQ Dual Bus Architecture

The SMDS protocol acts in the following fashion. Recall from the previous paragraph that on the forward channel a station that transmits data transmits not only the data but a bust bit, BUSY. We shall call the complement of this the IDLE bit. If the IDLE bit is 1 then the packet is empty. Also recall that on the reverse channel we have the request bit, REQ, that if it is 1, it means that some station upstream desires to transmit.

We can now describe the SMDS MAC protocol for transmission purposes. This is called the Distributed Queue Dual Bus (DQDB) algorithm and is the heart of the 802.6 MAC protocol.

o Each station has a Request Counter, called RQ. The contents of the RQ at any time k is called;

$$RQ(k) = \text{contents of the request counter at time } k$$

o When a data block on the reverse channel desires to send a message it generates a REQ bit that is loaded into the RQ. Thus when the station sees a reverse packet with a REQ we have;

$$RQ(k+1) = RQ(k) + REQ(k)$$

o When the forward channel sees and IDLE bit set at time k+2, it subtracts the IDLE bit from the request counter, RQ.

$$RQ(k+2) = RQ(k+1) - IDLE(k+1)$$

Thus RQ measures the number of requests left upstream less the number of idle packets that can satisfy those requests. Or, in effect, RQ is a measure of the status of the upstream demand at any time.

o When a node wishes to transmit it transfers the contents of the RQ to another buffer called the countdown buffer, CD.

o It then separates the RQ from the forward path and keeps it connected to the reverse. Thus RQ(k) becomes;

$$RQ(k) = RQ(k-1) + REQ(k)$$

o The countdown buffer is connected to the forward path. Then the CD buffer is as follows:

$$CD(k) = RQ(k); \text{ at the time of transfer}$$

$$CD(k) = CD(k-1) - IDLE(k); \text{ when an empty packet passes}$$

o When $CD(n) = 0$, the station transmits a packet.

This protocol is dramatically different than the FDDI 802.5 protocol since it incorporate a minimum distributed intelligence in each node of the network.

Figure 5.x SMDS MAC Protocol

Figure 5.x depicts the location of the SMDS protocol in terms of both the other OSI layers and in terms of its relationship to other LAN type protocols. The <\MAC protocol we have just developed and the SMDS Interface Protocol, the SIP, is also the 802.6 MAN protocol that we have developed. It is the DQDB protocol and the SNI provides "access" to it. The other MAC interface in Figure 5.x is that for the LAN that it is connected to. At the Layer 3 interface, it is assumed that the OSI Internetworking protocol be used, OSI IP.

Figure 5.x SMDS Relations to OSI Interfaces (p 2-5 BCR)

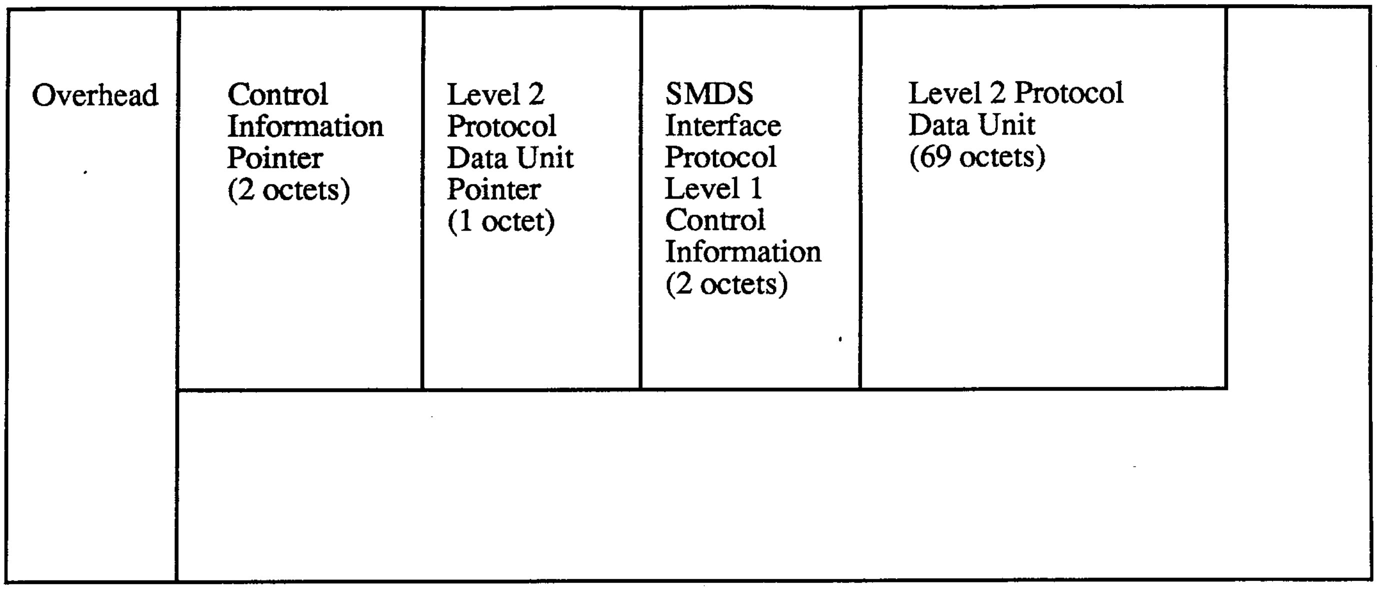


Figure 5.12.a : SMDS Interfacing - Interfacing of SMDS Level 2 with DS3.

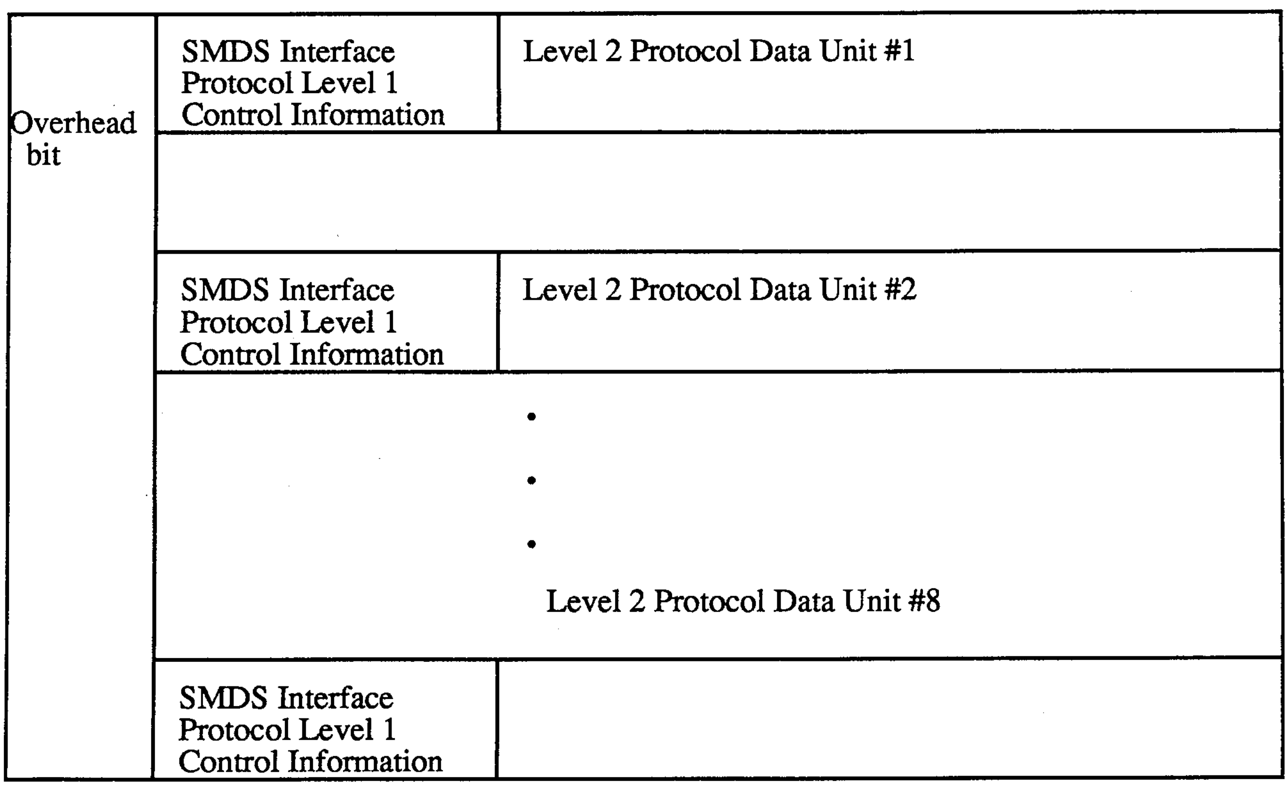


Figure 5.12.b : SMDS Interfacing - Interfacing of SMDS Level 2 with DS1 Extended Superframe Format.

Figure 5.x depicts the specific LAN/SMDS bridge. The design of the SMDS protocol is specifically developed to accommodate the LAN to LAN connection. The main driver for the implementation of the SMDS architecture was to meet the needs of the lAN users and their needs for interconnectivity.

Figure 5.x SMDS to LAN Interface (p2-6)

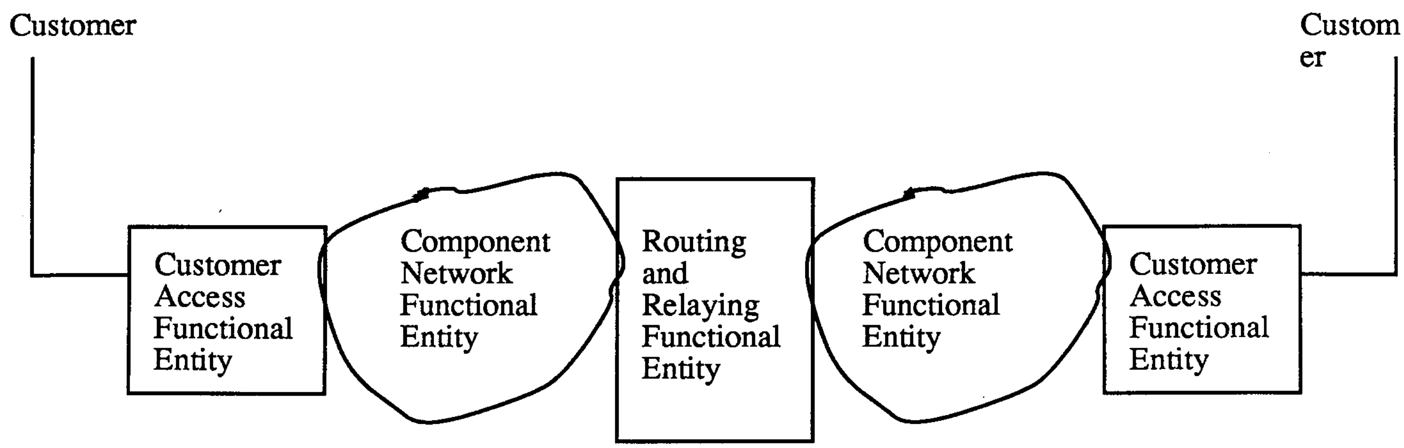


Figure 5.13 : SMDS Internetting and Conferencing.

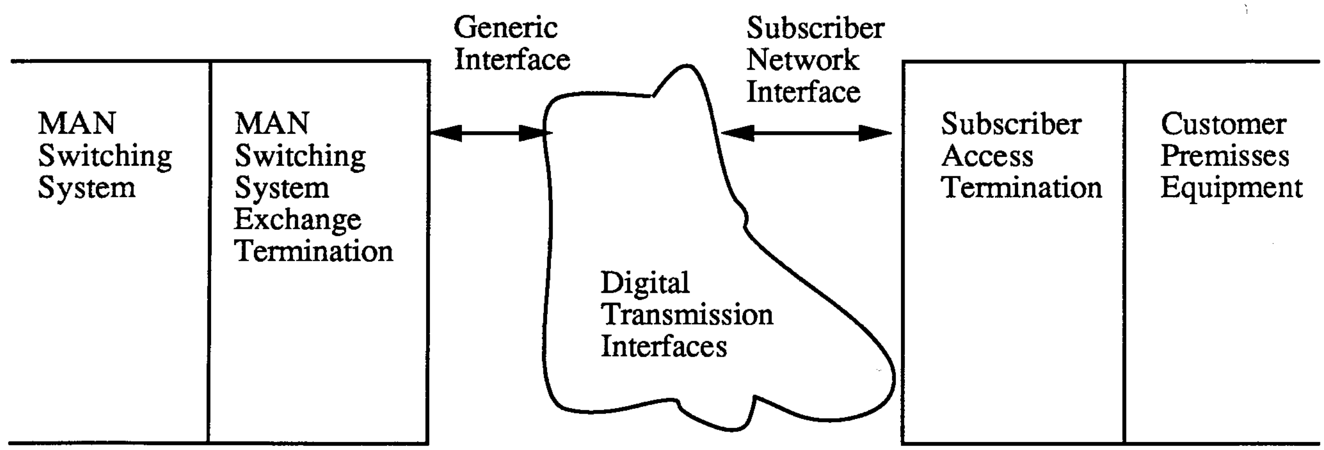


Figure 5.14 : A SMDS Interface Architecture.

The SMDS protocol has three layers. They are;

- o Level 3: Provide the SMDS addressing information. Also error detection is handled at this level.
- o Level 2: Segmentation and reassembly is provided at this level.
- o Level 1: This is the physical layer functionality.

Figure 5.x SMDS Protocol Layers (p4-3)

Destination Address	Source Address	Higher Layer Protocol Identifier	X+	Header Extension Length	X+	Length Indicator	Header Checksum	Header Extension	Information	Information Check Sequence
64 bits	64 bits	8 bits	4 bits	4 bits	16 bits	16 bits	16 bits	up to 36 octets	up to 9188 octets	32 bits

X+ : field placed to ensure alignment of the SMDS Interface Protocol.

Figure 5.11.a : SMDS Frame Format - OSI Level 3 Protocol Data Unit.

Access Control	Network Control Information	Segment Type	Message Identifier	Information
8 bits	32 bits	2 bits	14 bits	496 bits

Figure 5.11.b : SMDS Frame Format - OSI Level 2 Protocol Data Unit.

The full SMDS protocol layout is presented in Figure 5.x The SMDS service Unit , SDU, is the unit of data that is to be transmitted across the network. The SMDS Protocol Data Unit, PDU, is the unit that is exchanged between peer entities within a particular layer.

In Figure 5.x we show the SDU being loaded into the Level 3 PDU. The Level 3 PDU has a header and a check at the tail. The details of the Level 3 PDU are shown in 5.x (b). The Level 3 header contains;

- o Destination
- o Source
- o Higher Layer Protocol Identifier
- o X+
- o Header Extension
- o X+
- o Length Indicator
- o Head Checksum
- o Header Extension
- o Data
- o Info Check Sum

At the Level 2 PDU, in Figure 5.x (a) we show that we take the Level3, and packetize it in pieces, and place a Level 2 PDU header in front of each of the packetized elements. We have in the Level 2 header;

- o Access Control

- o Network Control Information
- o Segment Type
- o Message Identifier
- o Data

We further breakdown the Access control field to include:

- o BUSY: The MAC control field
- o XXX
- o REQ_1: A 1 priority request
- o REQ_N: An N priority request

Figure 5.x SMDS Level 3 and Level 2 Protocol Headers (p4-13)

(Level 2 p 4-8 Fig 4.8

Some simple performance analysis has been done on the SMDS protocol. To data, however, there is not as detailed a study of the DBDQ protocol as there is of the many other protocols. However, Newman et al have presented an analysis of the DBDQ protocol in terms of the access time delay and compared it to FDDI. As we noted in the FDI case, the delay increase as the network increase in size and also as we increase the traffic. In fact, it is possible to make the FDDI delay dependent upon the distance of the ring alone and have it independent of load. This fact is not at all evident in Figure 5.x from Newman et al.

Figure 5.x Access Delay Performance Comparison (FDDI and SMDS)
(Newmanp25)

SMDS provides an intermediate choice for data communications over a larger area, typically 100km and greater. It allows for many users to access a network and to do so at speeds that are consistent with both the telephone network and he local area network speeds. Thus SMDS presents an ideal candidate for metropolitan area networks (MANs).

5.3.4 ATM/BBISDN

The previous two networks are in various stages of implementation with their standards having been well defined and developed and the technology either available as with FDDI or clearly under development as is the case with SMDS. Broadband ISDN, is in some sense the son or grandson of ISDN which is slowly having limited introduction in some telephone areas. The broadband version of ISDN has two major directions. The first is the development of a synchronous system which has been built around the SONET standard and the second is an asynchronous system called the asynchronous transfer mode (AT) and this provides the most effective transport environment for the full end to end transport.

5.4 Network Management

The issue of network management addresses the need of both the operator of the system to control the resources available and the

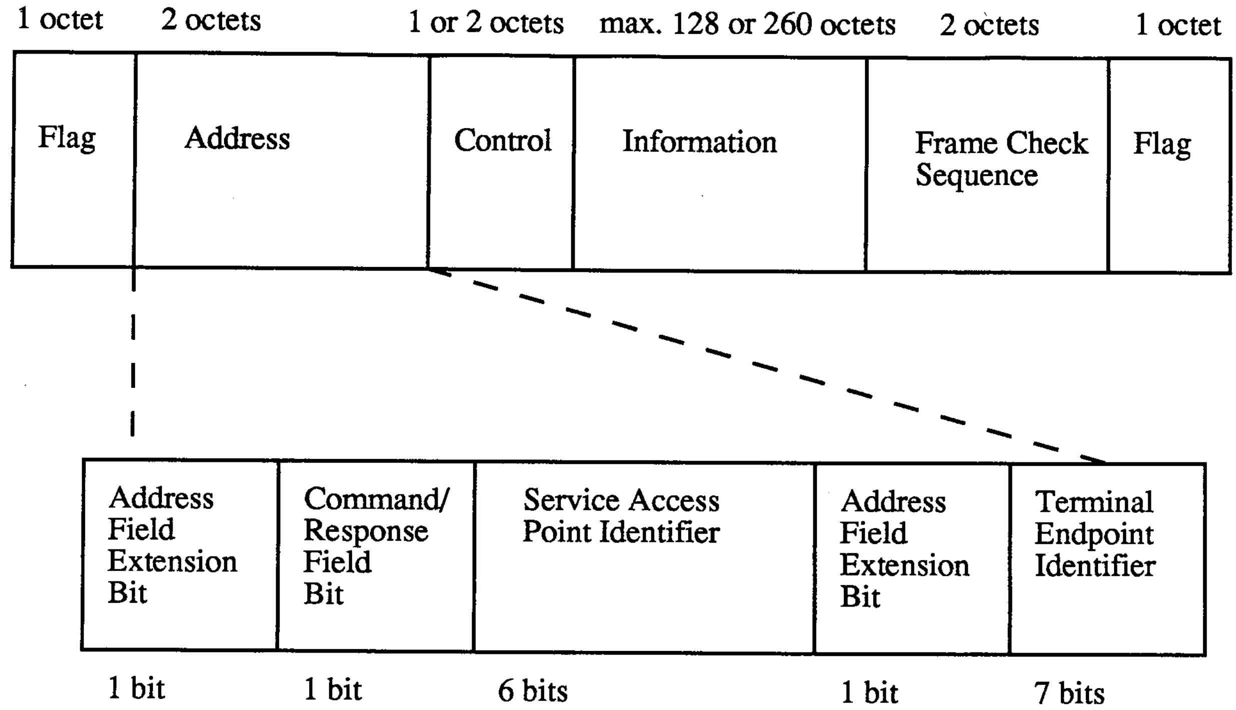


Figure 5.21.a : BBISDN Frame Format - Layer 2 ISDN Frame Format.

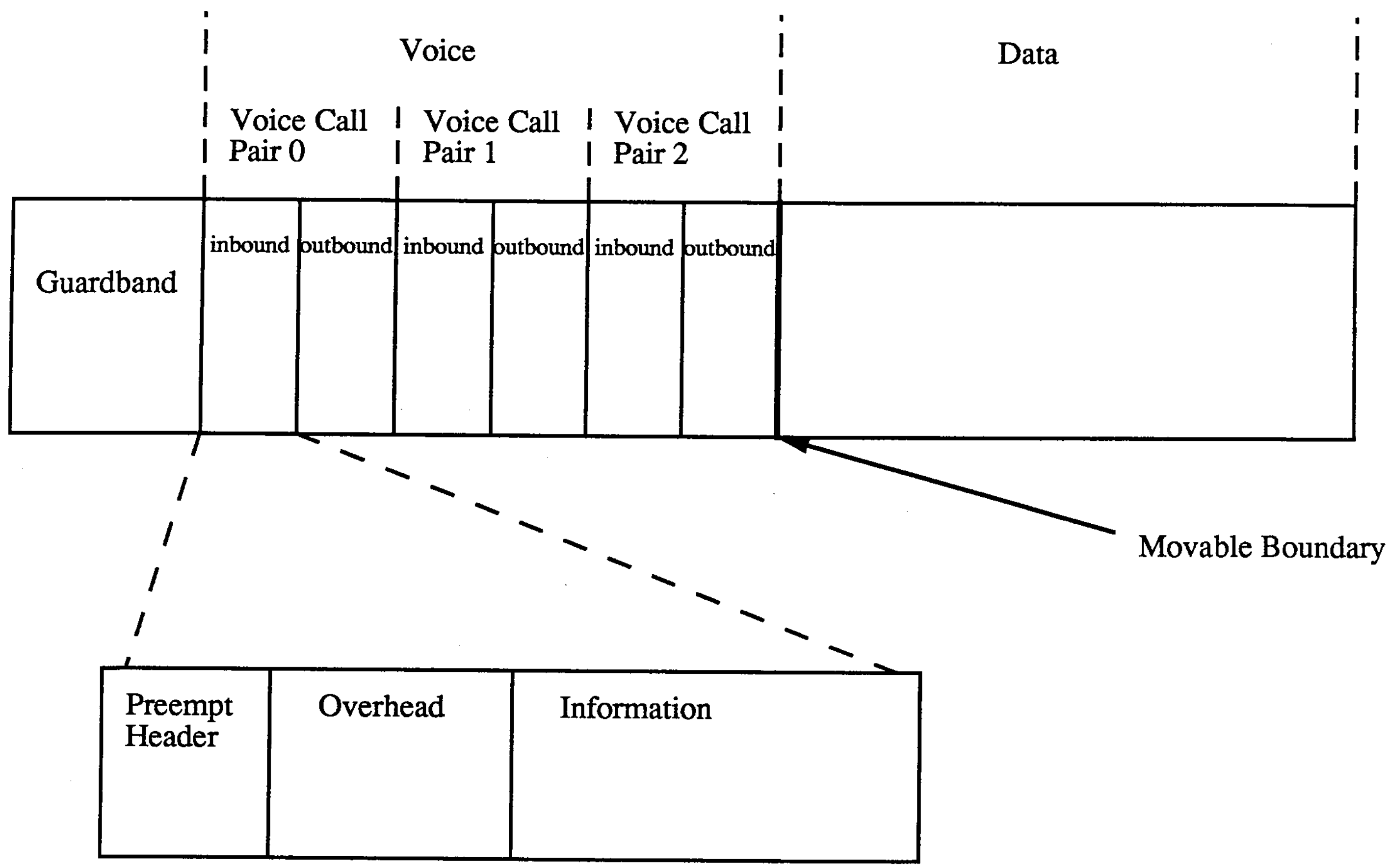


Figure 5.21.b : BBISDN Frame Format - Fixed-Frame Maxenchuck Format.

Multiplexing possibilities : - electrical frequency
- optical frequency
- time-division
- space-division.

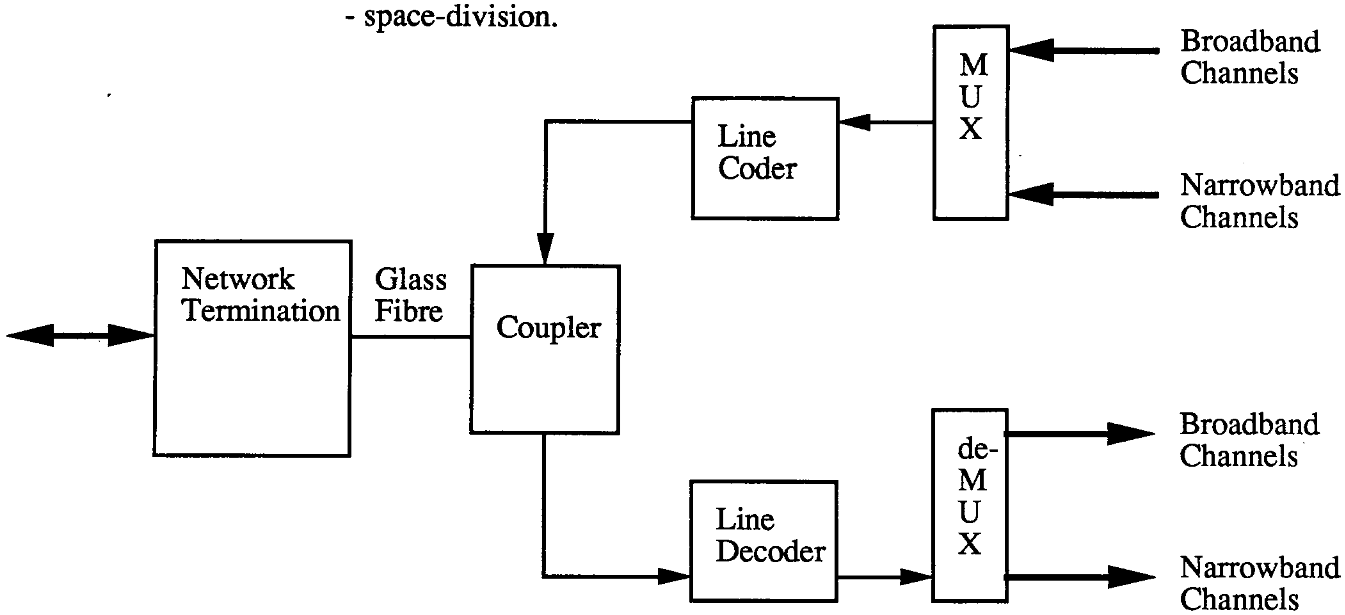


Figure 5.22.a : BBISDN Interfacing - Subscriber Access System.

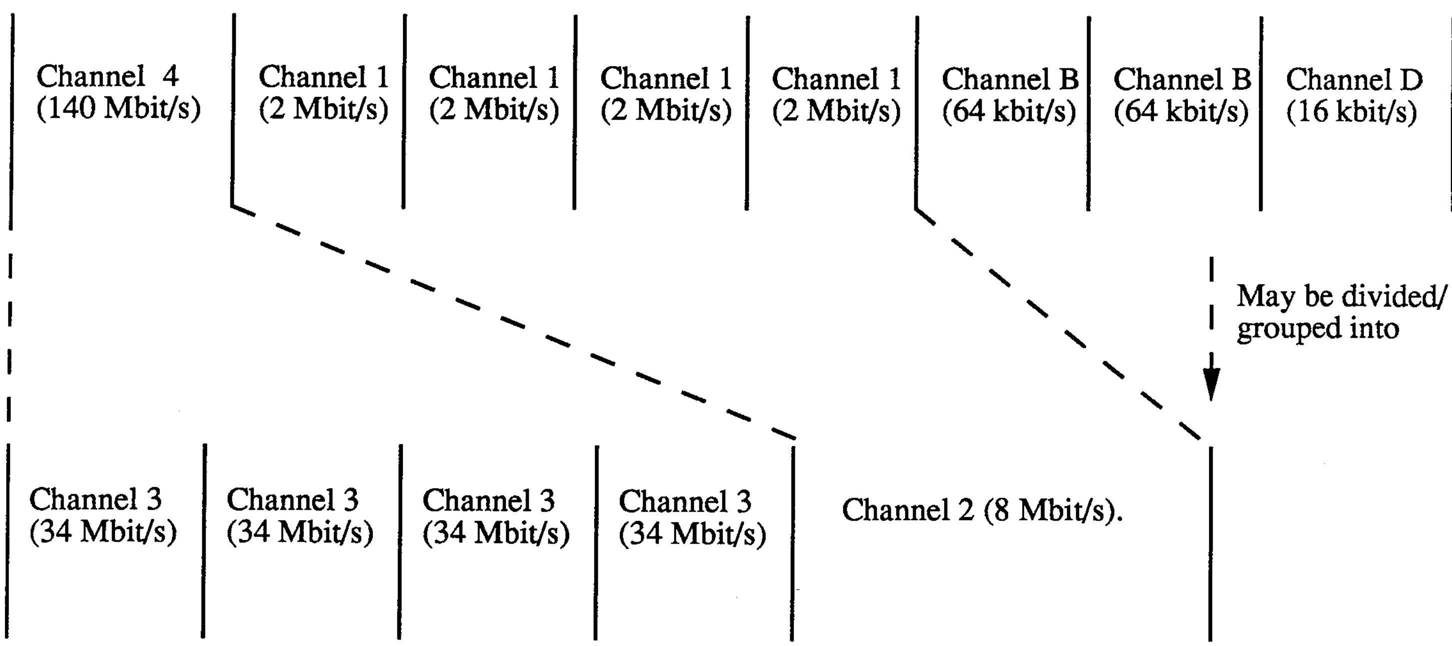


Figure 5.22.b : BBISDN Interfacing - Proposed Channel Structure for Broad-Band S Interface (e.g. for HDTV).

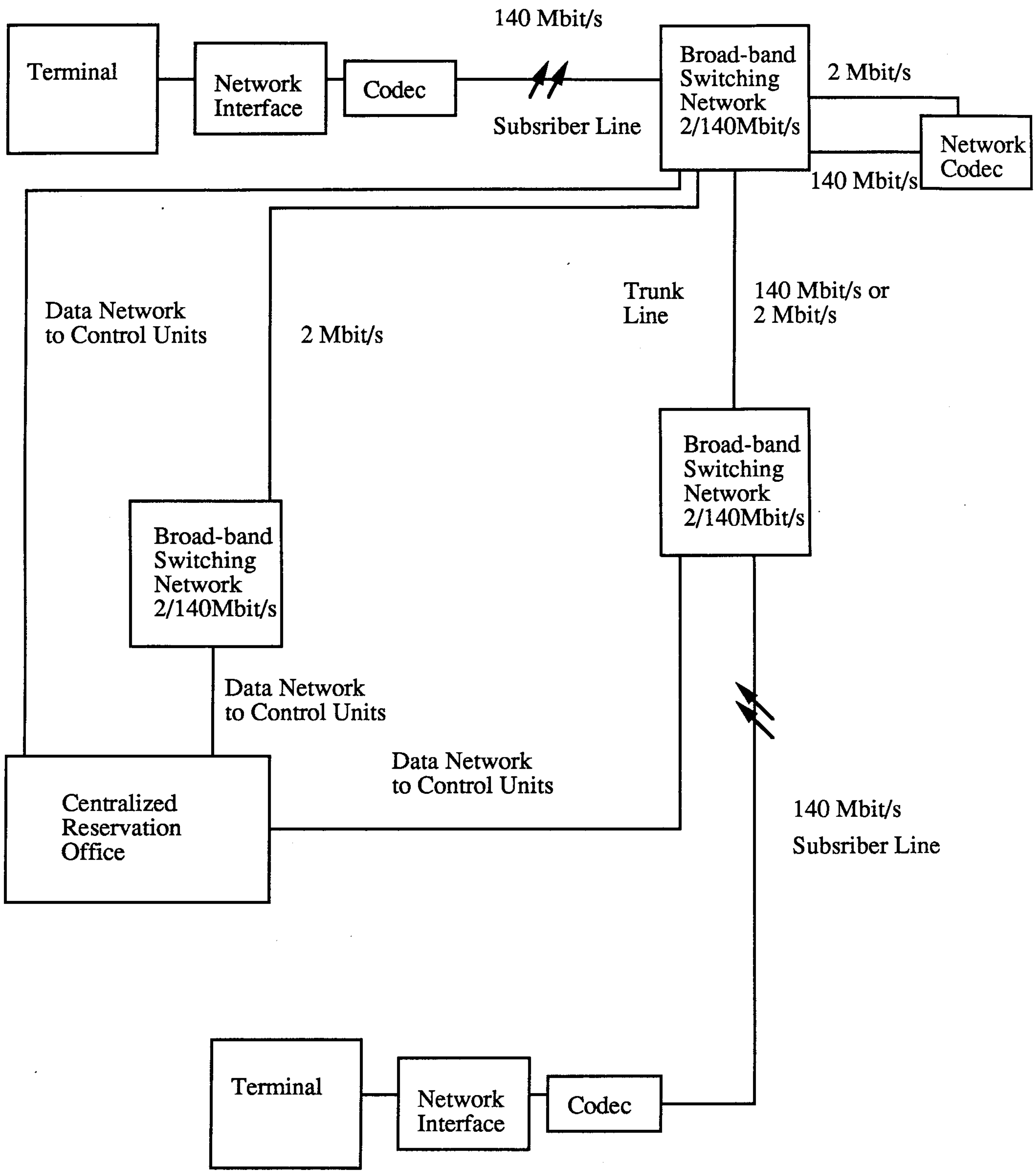


Figure 5.23 : BBISDN Internetting and Conferencing - German Experimental Videoconferencing Network.

need of the system user to obtain maximum benefit of the system and all of its resources. The task of network management is best stated by what is has to do for the user of the system. A good network management system must allow the user who is employing a wide set of system resources, and when the system fails in any way to meet the users needs, such as the return key not work, allowing the user to have a single source of contact who will return all of the resource to the user in a form which allow the user to complete their tasks. This function must be achieved in as transparent and direct fashion as possible.

In this section, we discuss network management from the perspective of the communications system, but we must remember that all of the elements of the multimedia environment must be controlled in as seamless a fashion. This network management in a multimedia environment must deal with the end user from the interface, through the data and file structures, through the communications network and including all of the network services and applications. This is a goal of such a system and this goal may not be readily achievable in the context of many of today's systems. However, if we begin to recognize the need of this function, we can more readily factor these into the overall design at the earliest possible stages.

An example of a typical network management problem is shown in the Figure presented in Figure 5.x. This is an example of a typical multimedia environment that needs the network management problem. Specifically, the network includes the following elements;

Figure 5.x Typical Multimedia Network

- Memory Storage devices including;
    - High Density Juke Box Storage
    - Voice Storage and Processing system
    - High density memory storage
- Computer Processing Units
    - IBM mainframes
    - DEC minicomputers
    - SUN workstations
- Local area networks
    - Token ring(802.5)
    - Token Bus(802.4)
    - Ethernet(802.3)
- High Speed Data Network
    - T-1 Switches
    - DS-3 Multiplexers
    - MAN Interfaces

These units each have an inherent sub network managers within their own functioning. However they are not controlled in a integrated format.

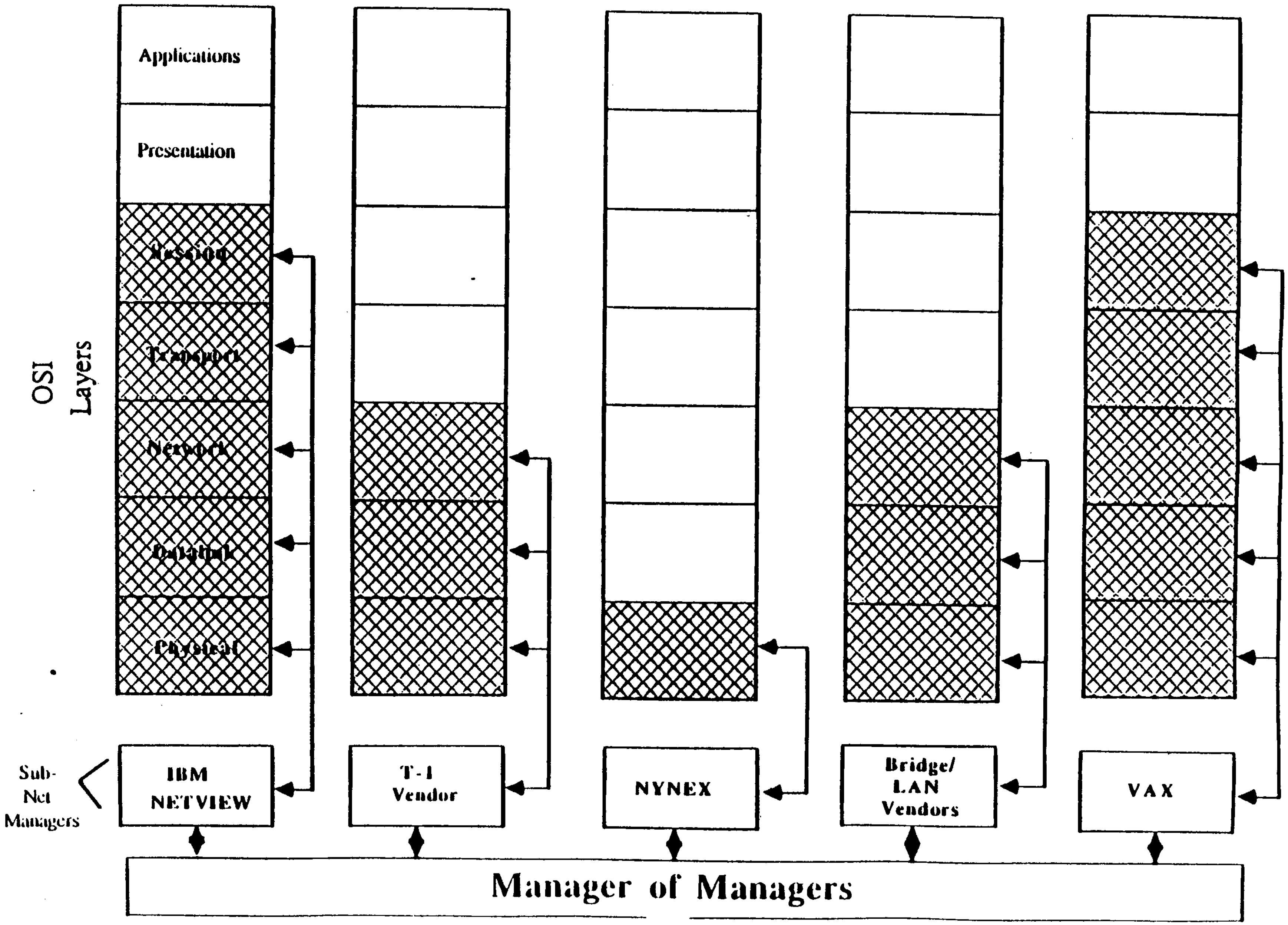
The network management problem can be stated as is shown in Figure 5.x. In this figure, we have shown several of the sub network elements and have in addition shown that the elements are comprised not only of their physical interfaces but of all seven

OSI layers. The problem then is if there is a network problem in any one layer it must be identified and corrected, and more importantly, if a problem arises as a result of the interaction between layers, then this is the critical issue associated with network management.

Figure 5.x Integrated Network Management Concept

Figure 3

Network Management Integration Concept



'/6/88

The solution of the network management problem is to develop a manager of managers that looks at all of the seven layers of each of the sub network management elements in each. It then can present an integrated system to identify, correct and restore the network functionality.

5.4.1 Functions

The network management system must perform the following functions;

o Interfacing: This function allows for the interfacing of the management system to talk to and control other sub network management systems. The interfacing function can be approached in two different ways. The first fashion is the approach of defining a standard interface to the overall management system and have all of the sub network vendors meet this interface. The second approach is to have a flexible and referable interface that does not require vendor modification.

o Status Monitoring: This function provides for the monitoring of the status of each of the individual sub network elements.

o Performance Monitoring: With the results from the status monitoring the performance monitoring function provides for an evaluation of how each of the sub network elements are functioning with regards to the standard in which they are to perform.

o Performance Analysis: The function of analysis determines the details of the system errors and faults and allows for the determination of how and where this can be improved.

o Inventory Management: This function is a key ingredient that provides for a ready access to all inventory of other elements of the system as well as the ability to achieve the restoral that is necessary.

o Restoral Activation: This function provides for the reconstruction of the assets of the network to perform its overall tasks.

o Reconfiguration Management: This function provides for the management of the overall reconfiguration task in contrast to the restoral activation function that actually implements the restoral. The reconfiguration management, looks not only at the resole functions of getting the system back to a prior state after a fault occurrence but also, and more importantly getting the system into a new state, one that is planned and orchestrated.

o Report Generation: A necessary function is the preparation of reports that are both event and time driven in nature. An event driven report is one that occurs when a particular event occurs. A time driven report is prepared at specified time intervals. The report generation function of the network manager must be flexible in its ability to meet the changes in the environment as well as flexible to be presented on various out display devices, print and video.

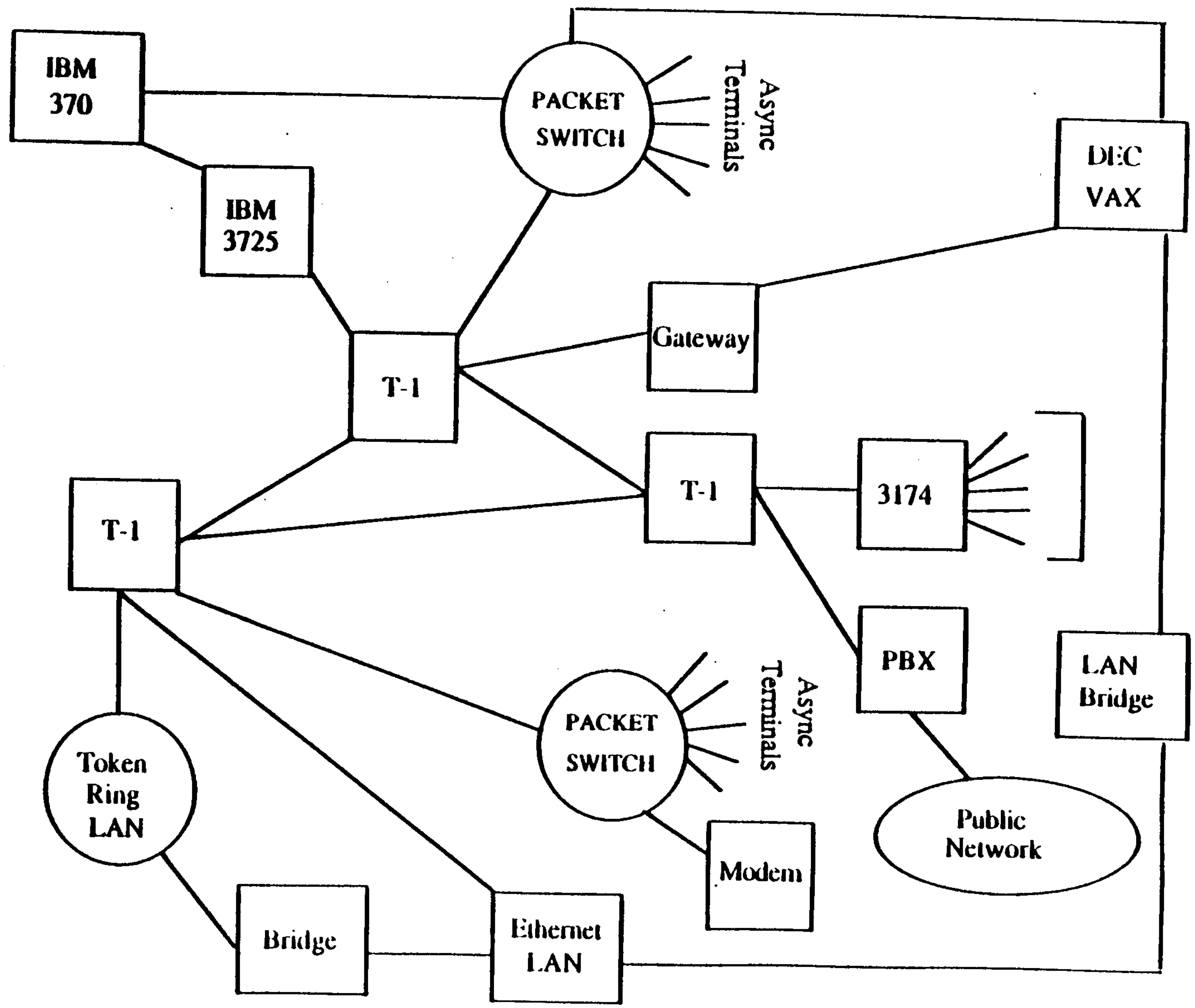
5.4.2 Architecture

The architecture of the network management system is shown in Figure 5.x. There are many subnetwork elements in the system and there are multiple interfaces in the overall architecture. The overall system architecture shows that the system operates in a multi domain applications. The system must cover two dimensions of operations. The first dimension is the interfaces across multiple sub network management systems, typically integrated at the lower level of the OSI, layers. The second level is within the same network and upwards thorough the OSI layers. A third dimension is desired but highly complex, is that integrating the first two dimensions.

The overall architecture consists of two major elements; the software and the hardware elements. The system architecture shows that there are common interfaces between the overall network manager and the sub network elements. these interfaces may be of a customizable type or even matched directly to each interface element. The software of the system must provide for the flexible interfaces to each of the systems and then must also be able top provide for the functions that we have defined above. The display of the results as also a key element in the design of the network management capability. The display function may have to be flexible to meet the needs of all the end users.

Figure 5.x Network Management Architecture

Figure 1
Typical Private Communications Network



5.4.2.1 Software

The system software architecture is depicted in Figure 5.x. It is composed of two layers that internally match the interfaces to the local subnetwok managers and externally face the end user. The inner facility of the network manager is the kernel functionality that must meet and interface with the multitude of other subnetwok elements. The outer layer is called the shell and it is through the functionality of this layer that the end users have access to the system.

We further take the functionality at each of the two layers and further divide it into foreground or real-time activities and background or nonreal time activities. The need for this partitioning is both for the implementation of the codes as well as for the implementation of the manager on a real time platform or target machine.

The kernel functions include the following:

o Communications Interface: This function provides for the subnetwok interface and handles the protocol support to manage the vagaries of the different interfaces. This function allows for the parsing of incoming messages and the same function for outgoing commands. The general parsing function provides for a flexible command interface and allows for the operation of that interface as a common element in the network manager.

o Performance Monitor: This is a time and event driven real time function that monitors all of the incoming status reports and determines if the report indicates an error status. The error status can be divided into several categories;

o Error Level Detection: This provides for the determination of the severity of the error and categorizes it instantly for action.

o Error Correlation: This is a complex task that takes less severe errors or faults and generates an assessment of the interaction of these faults to generate more complex errors.

o Performance Analysis:

o Status Monitoring

o Report Generation:

o System Analysis:

In a similar fashion, the shell,level functions can be give as:

o Performance Management

o Fault management:

o Configuration Management:

o Security Management:

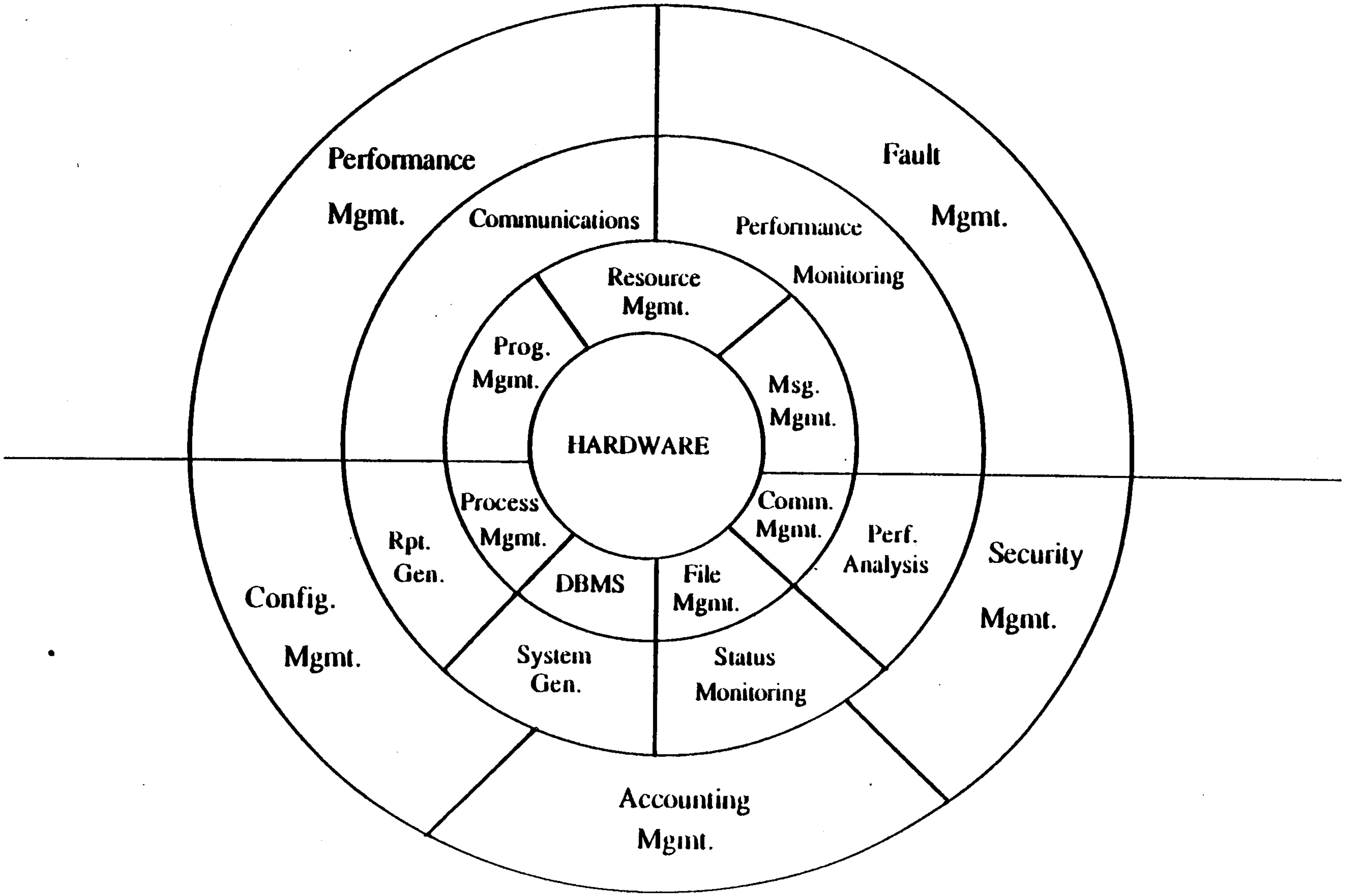
o Accounting Management:

Figure 5.x Network Management Software Architecture

Figure 2

Network Management Software Architecture

Foreground



Background

1/6/88

5.4.2.3 Hardware

The network management hardware architecture is fairly simple. The network is a complex collection of many elements and the network managers function is to acquire from a wide set of existing interfaces a complete collection of the network faults. These faults are locally generated and they are collected either in band or out of band of the existing signal paths. The in band collection uses the existing network paths during the periods in which they are properly functioning. In the event that they cease functioning a separate set of communications paths must be established. These are the out of band paths. Some architectures have these out of band paths in place for all communications. The latter approach may be costly and also occur the risk that they themselves may be error prone.

Figure 5.x Network Management Hardware Architecture

5.4.3 Implementations

This section describes several of the network management system implementations that are currently available; These systems are supplied by the following companies:

o NYNEX

o DEC

o AT&T

o IBM

5.5 Network Performance and Sizing

The issues of performance and sizing are complimentary issues. Performance addresses the issues of; given a specific load on the system and a specific set of system capabilities, what are the performance factors for the system in such areas as delay, throughput, response time and others. Sizing, on the other hand, addresses the issue of; given the fixed set of performance goals in terms of capacity, delay, throughput etc, how many users can the system support and how large should the system resource should be. Thus it is clear that when we address on of the two issues we address both.

In this section, we address the two issues from the viewpoint of the overall multimedia environment and do not provide a detailed discussion of either of the two elements. We leave these as details to be developed in the references and in the problems at the end of the chapter.

Figure 5.x Performance/Sizing Duality

5.5.1 Performance

5.5.2 Sizing

5.6 Conclusions

The communications element of the multimedia environment is frequently thought of as the most easily understood part of the overall system. All that is needed is faster bandwidth an greater connectivity. It is however the mosey complex because of the time scale of communications network evolution and system design. There are communications network that are intra premises and those that are inter premise. In this chapter we have developed the concepts of the standard architectures that we all assume will take care of the environments and their mixes. However, we see that there is clearly a asset of multiple alternative for the multimedia environment even at the lower layers of the protocol.

In this chapter we have also introduced the concept of network management. This is a critically important element in the overall multimedia system design and must be integrated into all of the elements that we have developed in this book. Unfortunately this is not the case and the user is often let to fen for themselves in the event of errors or faults occurring in some part of the network.

We have also further developed the clear distinction of sizing and performance in this environment and we shall extend it to the overlap environment of the multimedia network as we progress.

5.1 Multimedia Requirements.

5.1.1 Session Management.

'Multimedia Communications - Computer-Based Real-Time Conferencing Systems'
S. Sarin, I. Greif in
'Computer', IEEE October 1985

5.1.2 Interconnectivity.

'Communication Support in Operating Systems for Distributed Transactions'
A.Z. Spector
Department of Computer Science, Carnegie-Mellon University, 1986

'Request-Response amd Multicast Interprocess Communication in the V Kernel'
D.R. Cherion in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

5.1.3 Response Factors.

'Link-by-Link Bandwidth Allocation in an Integrated Voice/Data Network Using the Fuzzy Set Approach'
D. Tcha, C. Jin, E. Lutz in
'Computer Networks and ISDN Systems', 16, Elsevier Publishers, 1989

'A Typology of Local Area Networks -
Confused by the array of LANs, some users fall back on switching.Pity. Only LANs may be able to support mixed-media services and high-powered workstations'
P. Gregory in
'Telecommunications and Data Communications Factbook'
edited by J. Abbatiello, R. Sarch
Data Communications and CCMI/McGraw-Hill, 1987

'Communication Support in Operating Systems for Distributed Transactions'
A.Z. Spector in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis

J. Shantz in
'Telecommunications', May 1989

5.2.5.2 FDDI.

'Optical Interconnect : Upcoming LAN Technology'
L. Green in'Proceedings of the 12th Conference on Local Computer Networks of the Computer Society of the IEEE'
IEEE, 1987

'Implementation of FDDI : a 100 Mbit Token Ring'
L. Green in
'Localnet' 86'
Online Publishers, 1986

'How to get Ready for FDDI : Challenges and Opportunities'
H. Salwen in
'Proccedings of the 12th Conference on Local Computer Networks of the Computer Society of the IEEE'
IEEE, 1987

'First FDDI Local Area Network'
I. Kolnik, J. Garodnick in
'Proccedings of the 12th Conference on Local Computer Networks of the Computer Society of the IEEE'
IEEE, 1987

'Planning for FDDI'
L. Green in
'Proccedings of the 12th Conference on Local Computer Networks of the Computer Society of the IEEE'
IEEE, 1987

'Getting a Handle on FDDI'
K.J. Thurber in
'Data Communications', June 21, 1989

'Migrating to FDDI on Your Next Big LAN Installation'
M.R. Swastek, D.J. Verreke in
'Data Communications', June 21, 1989

'Token-Ring Local Area Networks and Their Performance'
W. Bux,
'Proceedings of the IEEE', vol.77, no. 2, February 1989

5.2.5.3 BBISDN.

'New Directions in Signaling for Broadband ISDN'
S.E. Minzer, D.R. Spears in
'IEEE Communications Magazine', February 1989

'Evolution of ISDN towards Broadband ISDN'
R. Händel in
'IEEE Network', January 1989

'Broadband ISDN Technology and Architecture'
W.R. Byrne, T.A. Kilm, B.L. Nekson, M.D. Soneru in
'IEEE Network', January, 1989

'Sonet and BISDN : A Marriage of Technologies'
T.C. Miller in
'Telephony', May 15, 1989

'An Overview of Integrated Services Digital Networks (ISDN)' in
'Datapro'
McGraw-Hill, March 1989

5.2.5.4 ATM.

'Conceptual Issues for ATM'
J. Gechter, P. O'Reilly in
'IEEE Network', January 1989

'Evolution from ISDN to BISDN : a Logical Step towards ATM'
M. de Prycker in
Computer Communications, vol. 12, no. 3, June 1989

5.3 Connectionless Systems.

'Comparison of Communication Services with Connection-Oriented and Connectionless Data Transmission'
O. Gihr, P.J. Kuehn in
'Computer Networking and Performance Evaluation'
edited by T. Hasegawa, H.Tagaki. Y. Takahashi
IFIP, North-Holland, 1986

5.3.1 Definition.

'Chapter 3, Section 9 : The Reference Model/

The Connectionless Model' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

5.3.2 Examples.

'Chapter 5 : Network Services',
'Chapter 4 : Wide Area Networks' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

'Codex Views on LAN Systems Integration'
R. Ueltzen in
'Localnet '86',
Online Publishers, 1986

5.3.3 Implementation.

'Communication Support for Distributed Processing : Design and Implementation Issues'
L. Svobodova in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

5.3.4 Performance.

'Chapter 9 : Network Control and Performance' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

5.4 Connection Based Systems.

'Comparison of Communication Services with Connection-Oriented and Connectionless Data Transmission'
O. Gihr, P.J. Kuehn in
'Computer Networking and Performance Evaluation'
edited by T. Hasegawa, H.Tagaki. Y. Takahashi
IFIP, North-Holland, 1986

5.4.1 Definition.

5.4.2 Examples.

'Chapter 5 : Network Services',
'Chapter 4 : Wide Area Networks' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

'Codex Views on LAN Systems Integration'
R. Ueltzen in
'Localnet '86',
Online Publishers, 1986

'Image Transmission over Grade Telephone Lines'
A.F. White in
'Applications of Digital Image Processing X - Proceedings'
edited by A.G. Tescher
SPIE - The International Society for Optical Engineering, 1987

5.4.3 Implementation.

'Communication Support for Distributed Processing : Deisign and Implementation
Issues'
L. Svobodova in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

5.4.4 Performance.

'Chapter 10 : Network Control and Performance' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

5.5 OSI Models

'Communication-System Models Embedded in the OSI-Reference Model, a Survey'
M. Reiser in
'Computer Networking and Performance Evaluation'
edited by T. Hasegawa, H.Tagaki. Y. Takahashi
IFIP, North-Holland, 1986

5.5.1 OSI Architectures

'Chapter 3 : The Reference Model' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

'European Harmonization of OSI Usage and Conformance Testing Systems'
H. Zimmermann in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Methodology and Framework for OSI Conformance Testing'
D. Rayner in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Commitment, Concurrency and Recovery - ISO-CASE and IBM LU.62'
J. Larmouth in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Considerations for Security in the OSI Architecture'
D.K. Branstad in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'NBS Program in Open Systems Interconnection (OSI)'
R.P. Blanc in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Chapter 1 : Introducing the ISO Reference Model',
'Chapter 2 : Principles of the ISO Reference Model',
'Chapter 9 : Consolidation by Example' in
'OSI Explained -
End-to-End Computer Communications Standards'

J. Henshall, S. Shaw
Ellis Horwood Series in Computer Communications and Networks, 1988

5.5.2 Physical Layer.

'Chapter 4/ Section 9 : Layers 1 to 3/
Physical Layer' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

'Implementing Fiber Optic in a Local Environment :
Opportunities and Experiences'
N. Housley in
'Localnet' 86'
Online Publishers, 1986

'Implementation of FDDI : a 100 Mbit Token Ring'
L. Green in
'Localnet' 86'
Online Publishers, 1986

'Chapter 3/ Section 3.2.1 : Communications and Computer Networks/
Computer Network Architecture/
The Physical Layer' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

5.5.3 Data Link Layer.

'LAN Communications Spurs New Technology'
J.L. Janson in
'Digital Review', February 6th, 1989

'Chapter 4/ Section 8 : Layers 1 to 3/
Data Link Layer' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

'Implementation of FDDI : a 100 Mbit Token Ring'
L. Green in

'Localnet' 86'
Online Publishers, 1986

'Chapter 3/ Section 3.2.2 : Communications and Computer Networks/
Computer Network Architecture/
The Data Link Layer' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

5.5.4 Network Layer.

'The Role of OSI in Distributed Computing'
L. Sbobodova in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Chapter 4/ Section 7 : Layers 1 to 3/
Network Layer Addressing' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

'Chapter 3/ Section 3.2.3 : Communications and Computer Networks/
Computer Network Architecture/
The Network Layer' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

5.5.5 Transport Layer.

'Getting the Most out of Ethernet's Data Link and Transport Layer'
U. Elzir in
'Data Communications', March 21, 1989

'A Performance Study of the ISO Transport Protocol'
B. Meister in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Illustration 5 : The Transport Station of the Danube Network' ,
'Illustration 3 : An Example of Transmission Protocol on Reliable Link :
PDP11 Unibus' in
'Distributed Computing Systems :
Communication, Cooperation, Consistency'
CORNAFION (group name)
Elsevier Science Publishers, 1985

'Chapter 5 : Transport Layer' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

'Chapter 3/ Section 3.2.4 : Communications and Computer Networks/
Computer Network Architecture/
The Transport Layer' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

'Chapter 7 : The Transport Layer' in
'OSI Explained -
End-to-End Computer Communications Standards'
J. Henshall, S. Shaw
Ellis Horwood Series in Computer Communications and Networks, 1988

5.5.6 Session Layer.

'Chapter 6 : The Session Layer' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

'Chapter 3/ Section 3.2.5 : Communications and Computer Networks/
Computer Network Architecture/
The Session Layer' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

'Chapter 6 : The Session Layer' in
'OSI Explained -

End-to-End Computer Communications Standards'
J. Henshall, S. Shaw
Ellis Horwood Series in Computer Communications and Networks, 1988

5.5.7 Presentation Layer

'Standardizing the Presentation Layer. Why and What?'
F. Caneschi, E. Merelli in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Chapter 5 : The Creation of Text and Graphic Displays' in
'Vieotex/Teletext - Principles and Practices'
A.F. Alber
McGraw-Hill, 1985

'Chapter 8 : The Presentation Layer' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

'Chapter 3/ Section 3.2.6 : Communications and Computer Networks/
Computer Network Architecture/
The Presentation Layer' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

'Chapter 5 : The Presentation Layer' in
'OSI Explained -
End-to-End Computer Communications Standards'
J. Henshall, S. Shaw
Ellis Horwood Series in Computer Communications and Networks, 1988

5.5.8 Applications Layer

'Chapter 9 : Common Application Service Elements' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth

McGraw-Hill, 1988

'Chapter 3/ Section3.2.7 : Communications and Computer Networks/
Computer Network Architecture/
The Application Layer' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

'Chapter 4 : The Application Layer' in
'OSI Explained -
End-to-End Computer Communications Standards'
J. Henshall, S. Shaw
Ellis Horwood Series in Computer Communications and Networks,
1988

5.6 Multimedia Applications of OSI.

'Multimedia Communications - Office Document Interchange
Formats : Current Status of International Standardization'
W. Horak in
'Computer', IEEE October 1985

'Office Communications'
J.S. McConnell, C-U. Manros in
'Computer Communications', vol. 12, no. 2, April 1989

'Telecommunications Trends in Europe'
M. Boisseau in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'The Work of CCITT in the Field of Telematic Services'
R. Rüggerberg ,
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Chapter 8 : File Transfer Access and Management',
'Chapter 10 : Message Handling Systems, X.400/MOTIS' in
'OSI Explained -
End-to-End Computer Communications Standards'
J. Henshall, S. Shaw

Ellis Horwood Series in Computer Communications and Networks, 1988

5.7 Implementation of Layers 1 to 3.

'Packet Distribution Network'
J.R. Rao in
'Proceedings of the 12th Conference on Local Computer Networks of the Computer Society of the IEEE'
IEEE, 1987

'The Knockout Packet Switch : Principles and Performances'
M.J. Karol, M.G. Hluchyj in
'Proccedings of the 12th Conference on Local Computer Networks of the Computer Society of the IEEE'
IEEE, 1987

'Implementation of Physical and Media Access Protocols for High-Speed Networks'
M. Skov in
'IEEE Communications Magazine', June 1989

'X.25 : It's Come a Long Way'
F.M. Burg, P. Puges in
'Computer Networks and ISDN Systems', 16, Elsevier Publishers, 1989

'Chapter 9 : Networks Architectures Standards and Protocols' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

'High-Level Communications Protocols on the Token-Ring Network'
J.T. Carlo, G.R. Samsen in
'Localnet' 86'
Online Publishers, 1986

'Chapter 4 : Layers 1 to 3' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

'Fiber Optic LAN Implementation Issues : a User's Perspective'
D.J. Cunningham in
'Localnet' 86'

Online Publishers, 1986

'Implementing Fiber Optic in a Local Environment : Opportunities and Experiences'
N. Housley in
'Localnet' 86'
Online Publishers, 1986

'Chapter 3/ Section 3 : Communications and Computer Networks/ Network Topology' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

5.8 Implementation of Layers 5 to 7.

'Chapter 7 : Overview of Presentation and Application Layers' in
'Standards for Open Systems Interconnections'
K.G. Knighston, T. Knowles, J. Larmouth
McGraw-Hill, 1988

'From Protocol Specification to Implementaion and Test'
H.J. Burkhardt, H. Eckert in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

5.9 Network Management.

'Network Management Systems' in
'Datapro'
McGraw-Hill, October 1988

5.9.1 Requirements.

'Modelling of New Services in Computer and Communication Networks'
P.J. Kuehn in
'Computer Networking and Performance Evaluation'
edited by T. Hasegawa, H.Tagaki. Y. Takahashi
IFIP, North-Holland, 1986

'Data Encryption Devices' in
'Datapro'
McGraw-Hill, December 1987

'Chapter 1 : General Principles of Networking' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wilet & Sons, 1986

'Networks : The Management Dimension'
M.D. Perlitch in
'Localnet '86',
Online Publishers, 1986

'Central Office Local Area Networks - a Viable Solution!'
B.A. Forest, J.F. Maranzano in
'Localnet' 86'
Online Publishers, 1986

'Chapter 4 / Section 4.1 : Operating Systems for Distributed and Parallel
Computing/
Network Operating Systems' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

'Directory System Requirements and the CCITT-Model'
B. Butscher, M. Tschichholz in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'The Work of CCITT in the Field of Telematic Services'
R. Rüggerberg in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Chapter 6 : Communications' in
'Building an Open System'
Jacob Slonim, Abraham Schonbach, Michael A. Bauer, Lachlan J. MacRae,
Keith A. Thomas
Van Nostrand Reinhold, 1987

5.9.2 Architecture.

5.9.2.1 Overall Architecture.

'Data Networks and Their Users'
J. Saint-Remi in
'Computer Networks and ISDN Systems', 14, Elsevier Science Publishers B.V. , 1988

'Chapter 2 : Local Area Networks',
'Chapter 3 : Personal Computer Networks and Micronets',
'Chapter 4 : Wide Area Networks',
'Chapter 6 : Network Structures' ,
'Chapter 19 : Recent Developments in Network Systems and Services',
'Chapter 23 : Case Studies of Networks' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

'The DUAL Backbone Network : Distributed & Parallel Processing on a Large Scale'
A. Endrizzi in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Central Office Local Area Networks - a Viable Solution!'
B.A. Forest, J.F. Maranzano in
'Localnet' 86'
Online Publishers, 1986

'A Local Net Product Family for DoD Networking'
C.A. Sunshine in
'Localnet '86'
Online Publications, 1986

'The Emergence of TCP/IP'
S. Spanier in
'Localnet '86'
Online Publications, 1986

'Networking in a Heterogeneous Environment'
T.W. Bajzek in
'Localnet '86'
Online Publications, 1986

'SNA : Recent Advances and Additional Requirements'
R.J. Sundstrom in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'ISDN Technology, Networking Concepts and Applications'
P.J. Kuehn in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

5.9.2.2 Hardware Architecture.

'Chapter 8 : Transmission Technologies'
'Chapter 20 : Recent Developments in Network Technologies' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

'Central Office Local Area Networks - a Viable Solution!'
B.A. Forest, J.F. Maranzano in
'Localnet' 86'
Online Publishers, 1986

'The Local Network Solution'
J.E. Snyder, H.M. France in
'Localnet '86'
Online Publications, 1986

'A User-Designed Voice/Data Facilities Wiring System'
T. Stoner, R. Nevins in
'Localnet' 86'
Online Publishers, 1986

'Fiber Optic LAN Implementation Issues : a User's Perspective'
D.J. Cunningham in
'Localnet' 86'
Online Publishers, 1986

'Implementing Fiber Optic in a Local Environment :
Opportunities and Experiences'
N. Housley in
'Localnet' 86'

Online Publishers, 1986

5.9.2.3 Software Architecture.

'Chapter 13 : Distributed Computing Systems' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

'Scheduling Remote Processing Capacity in a Workstation-Processor Bank Network'
M.W. Mutka, M. Livny in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Development Tools for Communication Protocols'
N. Nounou, Y. Yemini in
'Current Advances in Distributed Computing and Communications'
edited by Y. Yemini
Computer Science Press, 1987

'High-Level Communications Protocols on the Token-Ring Network'
J.T. Carlo, G.R. Samsen in
'Localnet '86'
Online Publications, 1986

'Central Office Local Area Networks - a Viable Solution!'
B.A. Forest, J.F. Maranzano in
'Localnet' 86'
Online Publishers, 1986

'The Local Network Solution'
J.E. Snyder, H.M. France in
'Localnet '86'
Online Publications, 1986

5.9.3 Interfaces.

'Bridges or Routers?'
W.M. Siefert in
'Proceedings of the 12th Conference on Local Computer Networks of the Computer Society of the IEEE'

CHAPTER 6

Distributed Databases

Breakout a chapter on distributed systems ① Discuss theory of DB closeness + Query appr'ch.

In developing the multimedia communications network, there were many elements that were left defined but not addressed. Specifically, we discussed many of the higher layer services that can be provided in sessioning, presentation and application layers. In a multiuser environment, these services must be implemented to support a fully distributed capability.

What can info theory do to the answers!

Whenever a network is working across several departmental, organizational and geographical boundaries, it becomes inefficient to have a fully centralized system design. In those cases, it is more appropriate to have a fully distributed design.

This chapter discusses the issues and alternatives of distributed elements. In particular we focus on distributed data bases, distributed operating systems and distributed processor configurations. We develop these in the overall context of the architectural requirements of such distributed architectures. In addition, we focus on understanding the performance factors and their tradeoff factors.

6.1 Environment Factors

The distributed environment includes the combination of the distributed users and the ability to include many of the systems resources into the ongoing session. The session will have the capability to access the processors and the databases and should be able to do so in as real time fashion as possible. The evolution of distributed environments has developed over the past twenty years. The earliest system was with the development of the MULTICS operating environment that was developed at MIT for the use in a time sharing computer system. It further evolved with the development of IBM's SNA architecture in the early 1970's. Yet in all of these systems there was still a hierarchical system that controlled all of the systems resources.

A true distributed environment requires that all of the resources and the users can have an arbitrary but definable relationship to all other users in the system. For example, the session concept allows for the development of a complex environment wherein any user in the session can play the lead of follower roles and any resources available in a session can be shared in any fashion with any user or sets of users.

Figure 6.1 depicts the overall structure of a distributed environment. The environment entails all of the resources being at several locations and that there is now single point of management and control. The distributed environment is distributed in five dimensions that we shall discuss in this chapter:

o Processors: This is a physical distribution of the systems resources. Typically we envision the location of computing resources to be a one location and the users share that resource in some fashion. Current computer architectures allow for local distribution of assets such as is found in the VAX cluster concept. Research efforts have allowed for the fuller distribution of computing resources through the use of simple distributed operating systems. However, the fully distributed processor environment is possible with the use of the techniques that are discussed in this section. Thus no single processor is in total control and the processors share resources in an almost parallel fashion. We can envision this as an adaptive parallel processing environment, wherein the connection between the parallel processors is logical and not just physical.

o Processes: Distributed processes are already a common capability in many systems. The distributed process environment allows for the management of the process execution over several processors at the same time. This may be of use in both real time distributed systems as well as in fault tolerant transaction designs.

o Database: The distributed database environment has evolved over the past several years and there is now a body of technology that can support such operations.The distributed database is a key element in a distributed system. It becomes the backbone of operating in an environment where there is data generated by many users and this data has time sensitive nature.

o Operating System and Management: The operating system environment is the overall mechanism for controlling the multiple system resources. With the use of distributed processors and the need for coordinating the set of all system resources, there is a need to do so in a fully distributed environment. The overall management of these resources is done through the operating system.

o Communications: The essence of the communications environment is the session. As we have discussed, the session is the access mode for enabling multiple users to share resources, data, and process applications. The distributed communications environment must be built upon the overall structure of the distributed processors and operating system.

In Figure 6.1 we detail these elements and show their interrelation.

Figure 6.1 The Distributed Environment

Figure 8.1 The Distributed Environment

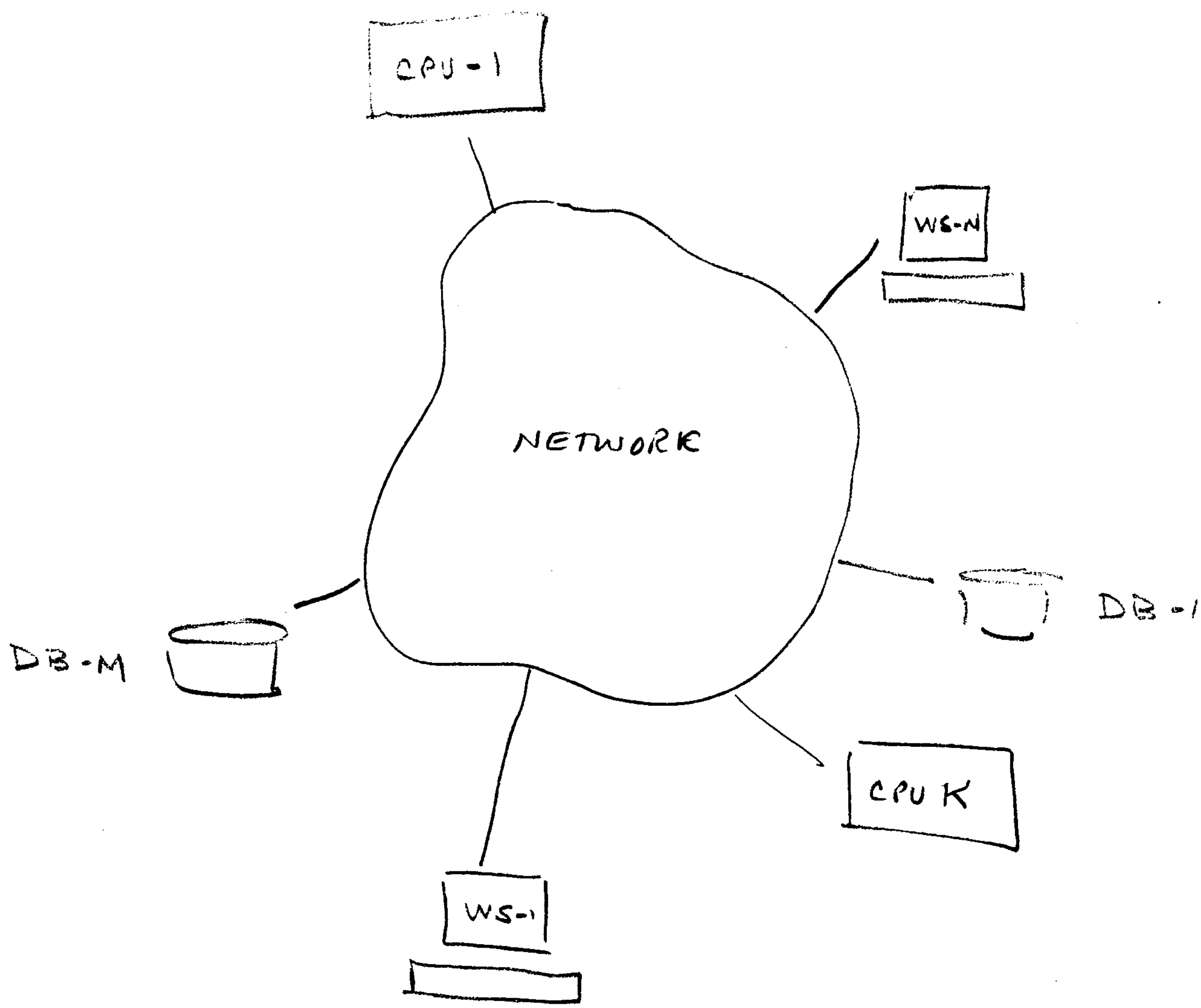


Figure 8.2 Distributed Processors

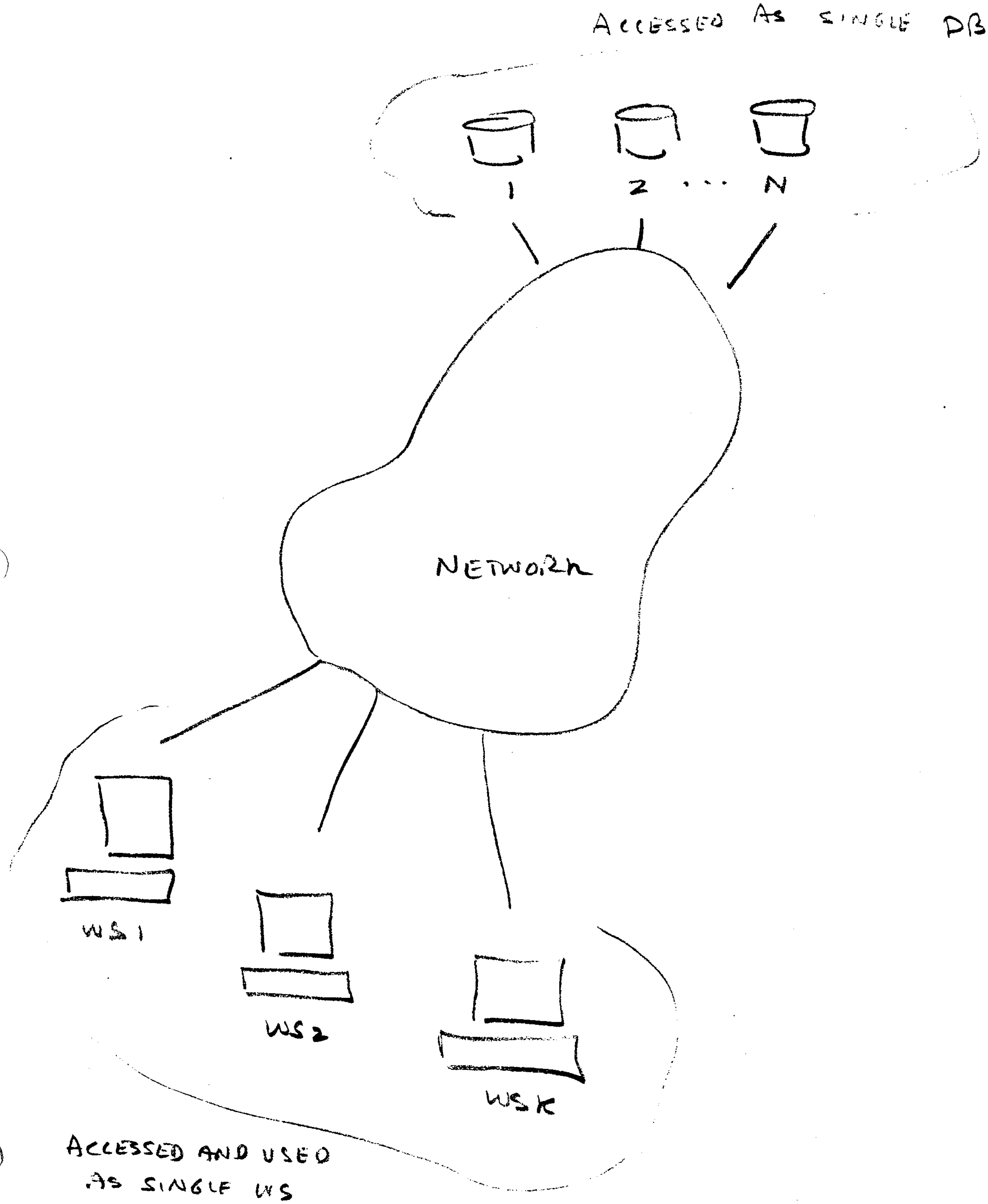


Figure 6.2 shows the structure of the distributed processor environment. We have depicted two possible architectures. The first is the heterogeneous environment, one that is more common in systems operation. The heterogeneous environment shows the need to share such different resource as memory, CPU, data files, image storage devices and I/O devices. The specific application envisioned is that of the printing and publishing industry.

The second environment is the homogeneous environment. This is the truly distributed processor environment wherein we can envision multiple processor locations of similar devices all sharing in the overall processor loads. An example is the internetting of multiple work stations so as to allow the utilization of all the resources simultaneously.

Figure 6.2 Distributed Processors

Figure 6.3 shows the configurations for the distributed process environment. Recall that a process is a program in execution. A process is typically associated with a single processor and the functioning a process across multiple processors is generally not expected. However, in a fully distributed environment, there is a clear need for such functioning. Consider the following example.

In a medical application are, there is the need for radiological consults between the primary physician, the neurosurgeon, the neuroradiologist and the oncologist for a patient with a primary brain tumor. The consult is conducted in the context of a session and uses the resources of several processors as well as several database. There is an application program that can take the multiple radiologic dat, MRI, CAT and nuclear scans, and combine them into a three dimensional image of the patient. The neurosurgeon needs to manipulate this for the selection of laser surgery procedure and the oncologist is interested for the impacts of blood brain barrier effects for post operative chemotherapy.

The process that is in operation is resident on each of the specialist processors and is shared amounts them.

Figure 6.3 Distributed Processes

Figure 8.3 Distributed Processes

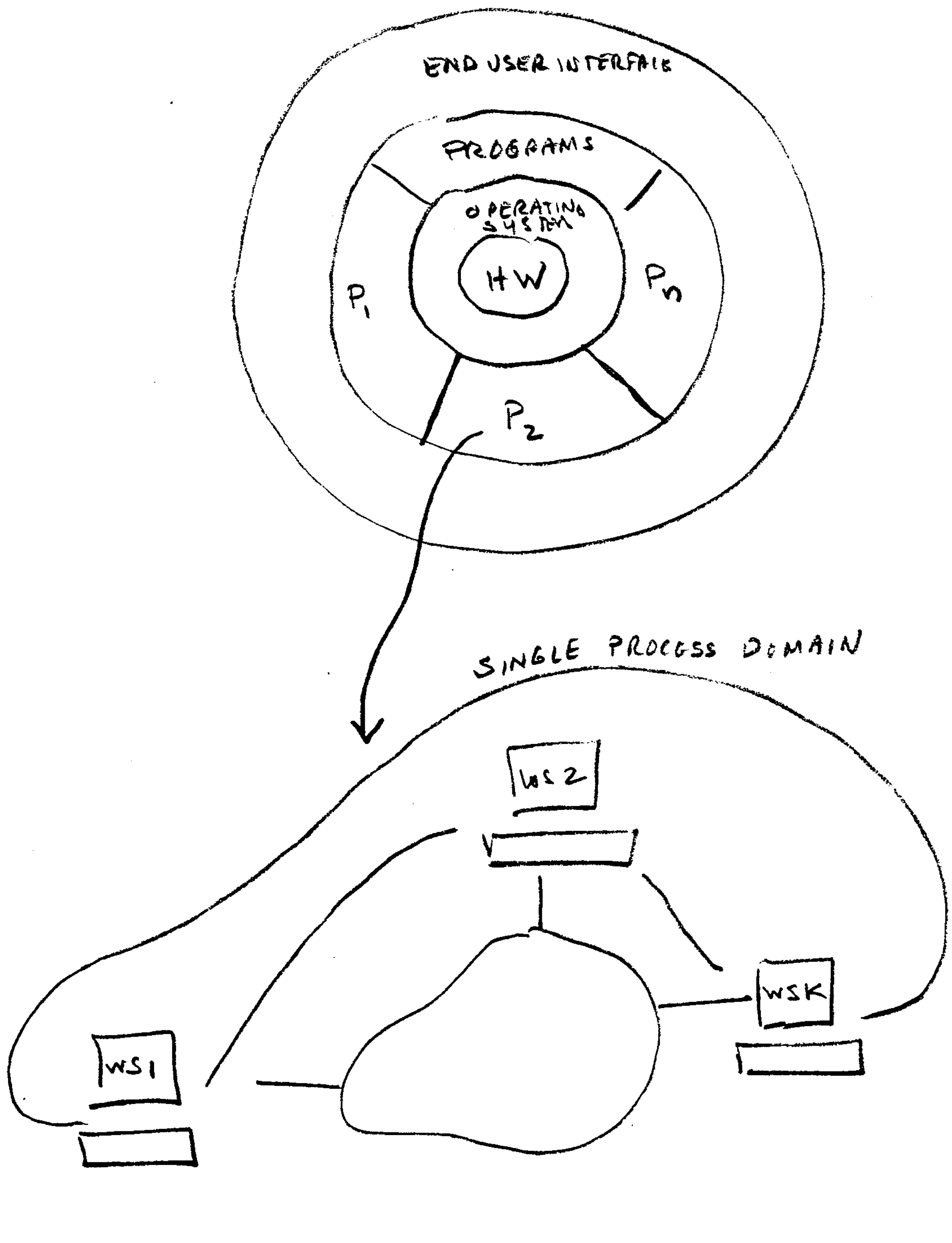


Figure 6.4 depicts the structure of a distributed data base example. In this particular case we can envision the application being in the area of financial transactions. The system shown in the figure is used for the checking of credit card validation and assuring that the seller can obtain payment from the buyer. The system must track the activity of all the purchasers and seller in the national network. In particular ,care must be taken to track the performance of any single card and to see if it has been used an excessive number of times.

In present day credit card systems, the data base is centralized and allocate checking is done in a hierarchical manner. The data base is accessed via a communications network and the overall cost per transaction can be quite high due to the communications overhead. The distributed database design as shown in the figure, allows for minimum communications and a distributed set of datafiles with full interconnectivity between them. This system requires that a single card users has their activity tracked as they move geographically through the system. One way to do this is to pass the file off to the local database and then to act as if the system were a fully segmented design. The second alternative is to keep parts of records at a distributed set of locations and to have a set of composite numbers, such a total exposure per card tallied at all or a few locations.

Figure 6.4 Distributed Data Files

Figure 6.5 shows the example of the distributed operating system environment. We shall develop this concept in detail in latter sections. However, the overall requirement for a distributed operating system is to allow for the overall management of the systems resources by managing processes, managing data, memory and controlling events. Also the operating system must allow foe control of the overall I/O resources of the system.

Figure 6.5 Distributed Operating System

Figure 8.4 Distributed Data Files

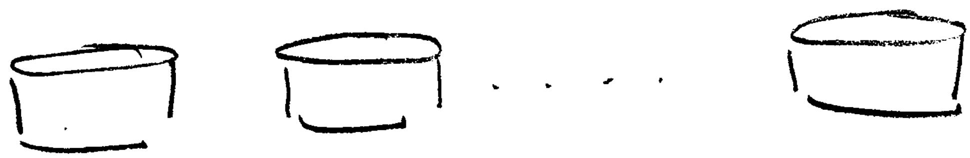


Figure 8.5 Distributed Modules

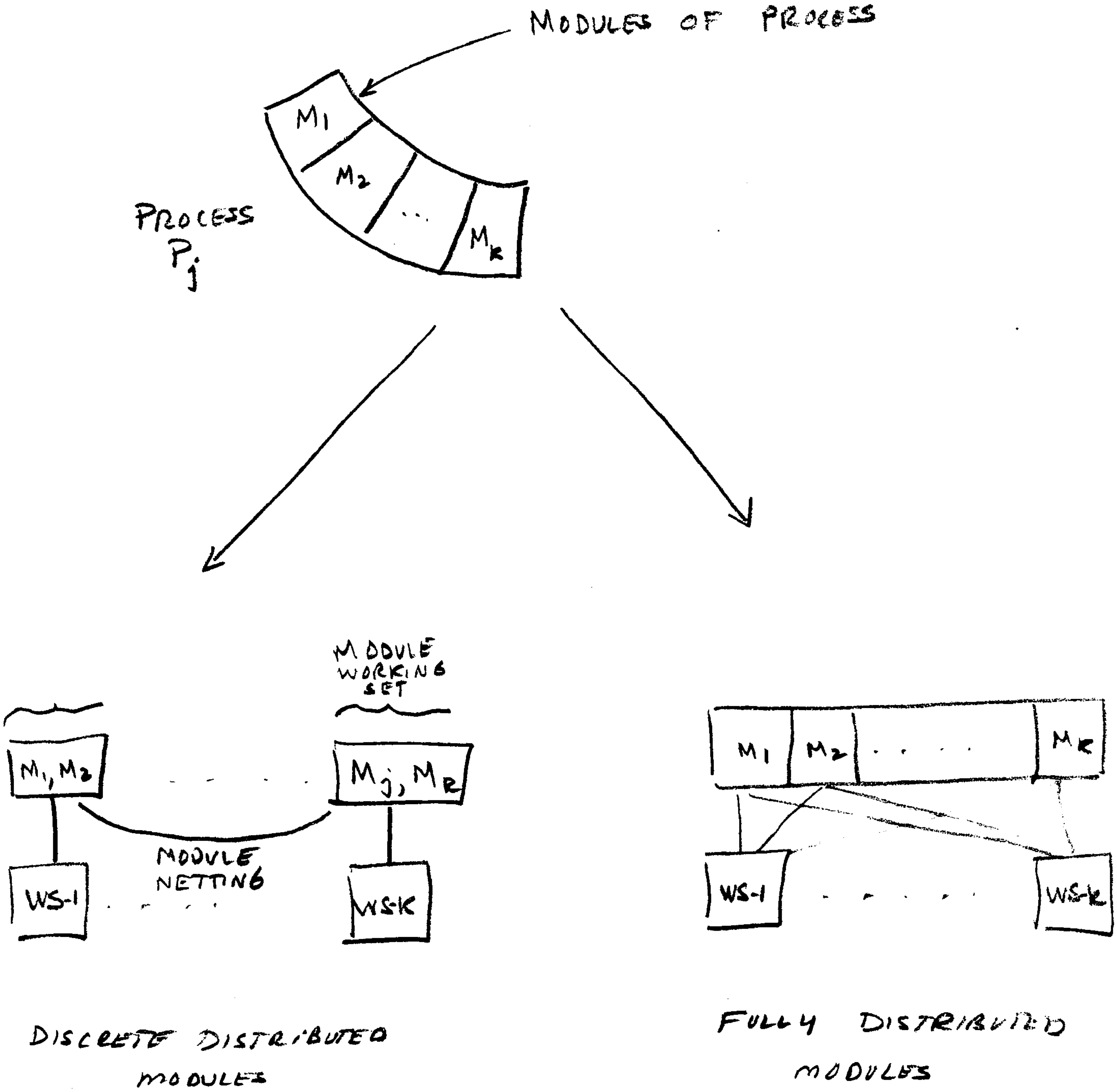


Figure 8.6 Interrelationships of Distributed Elements

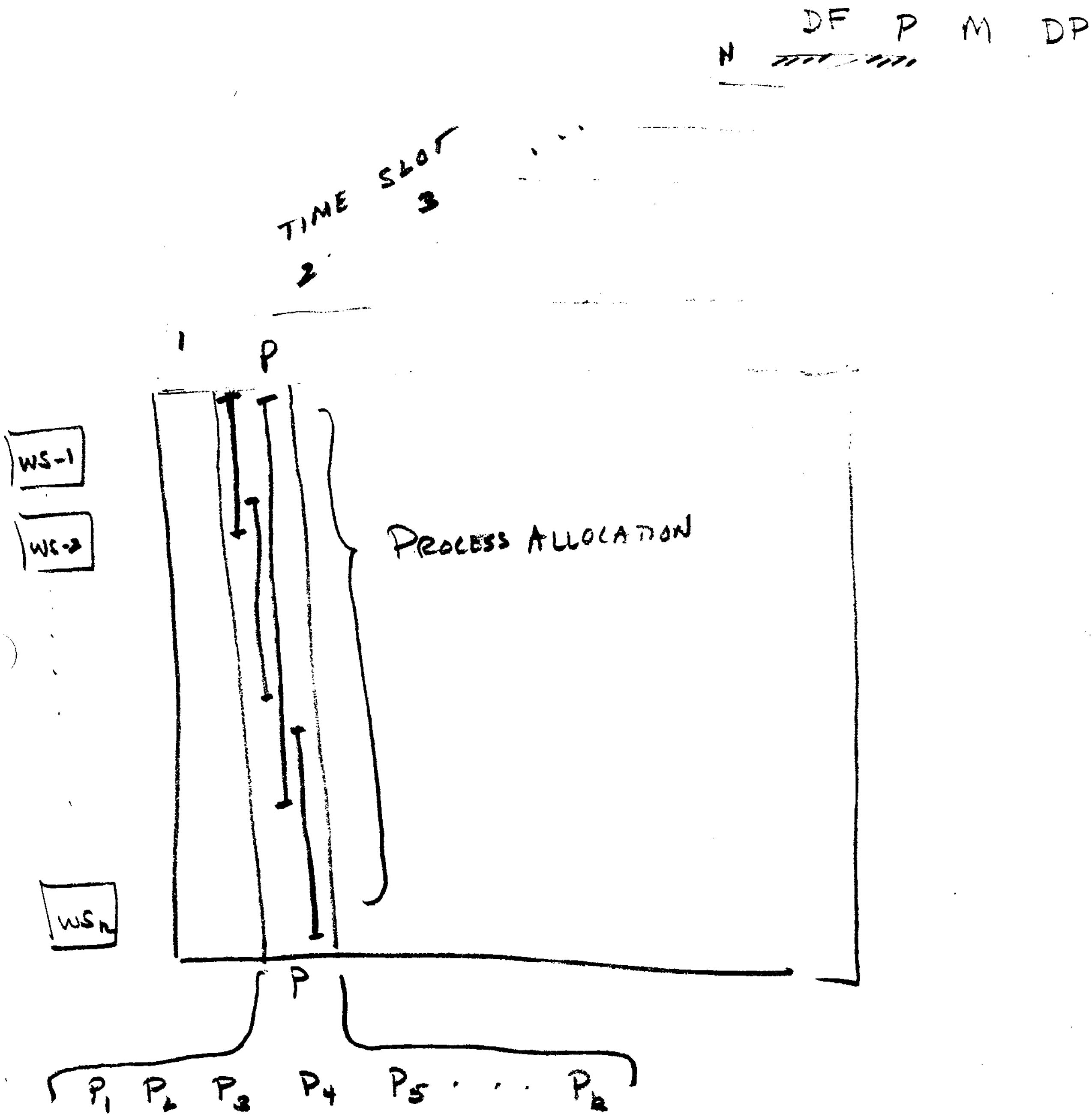


Figure 8.7 a Financial Transaction Processing Example

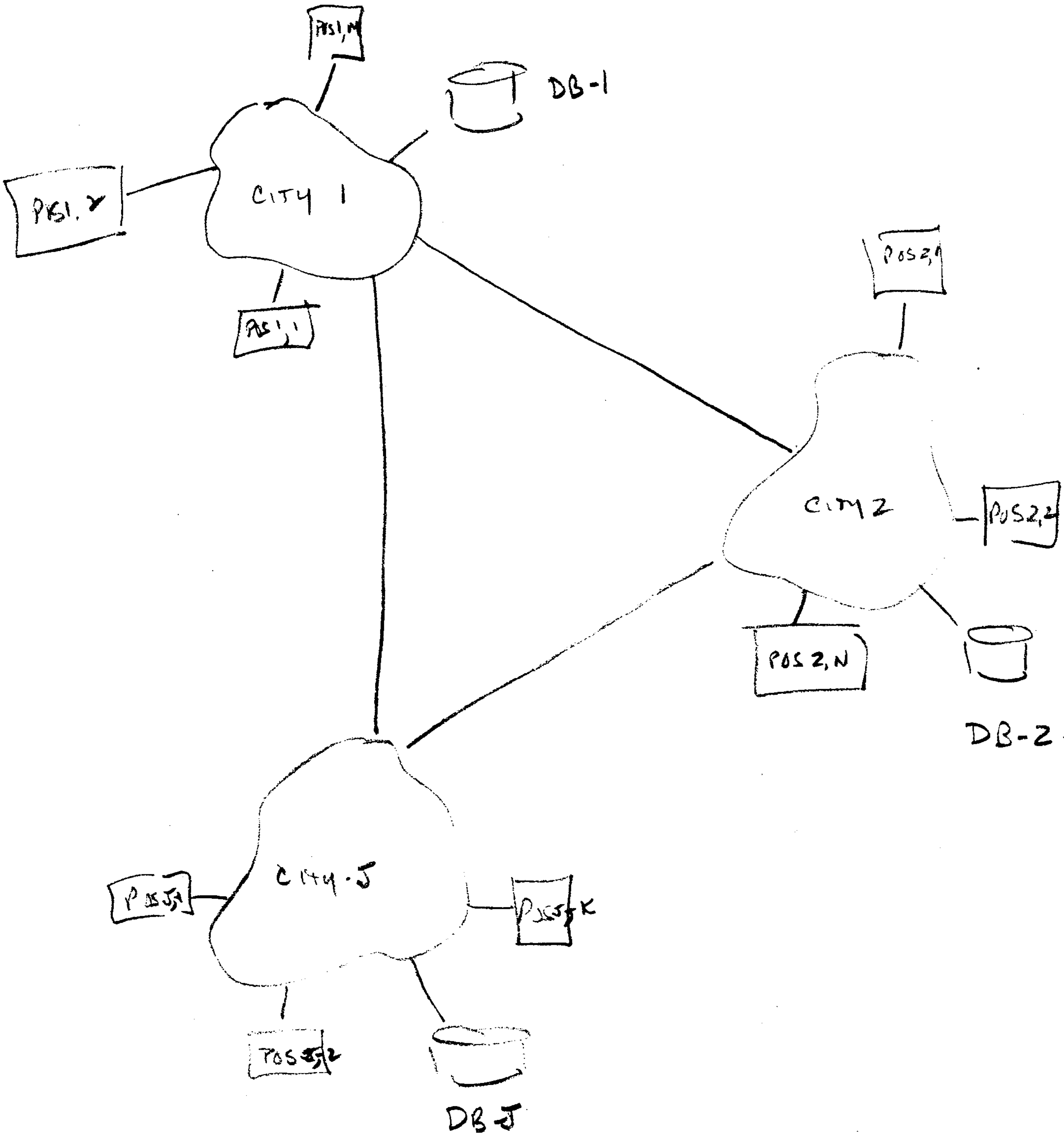


Figure 8.7 b Medical Records Example

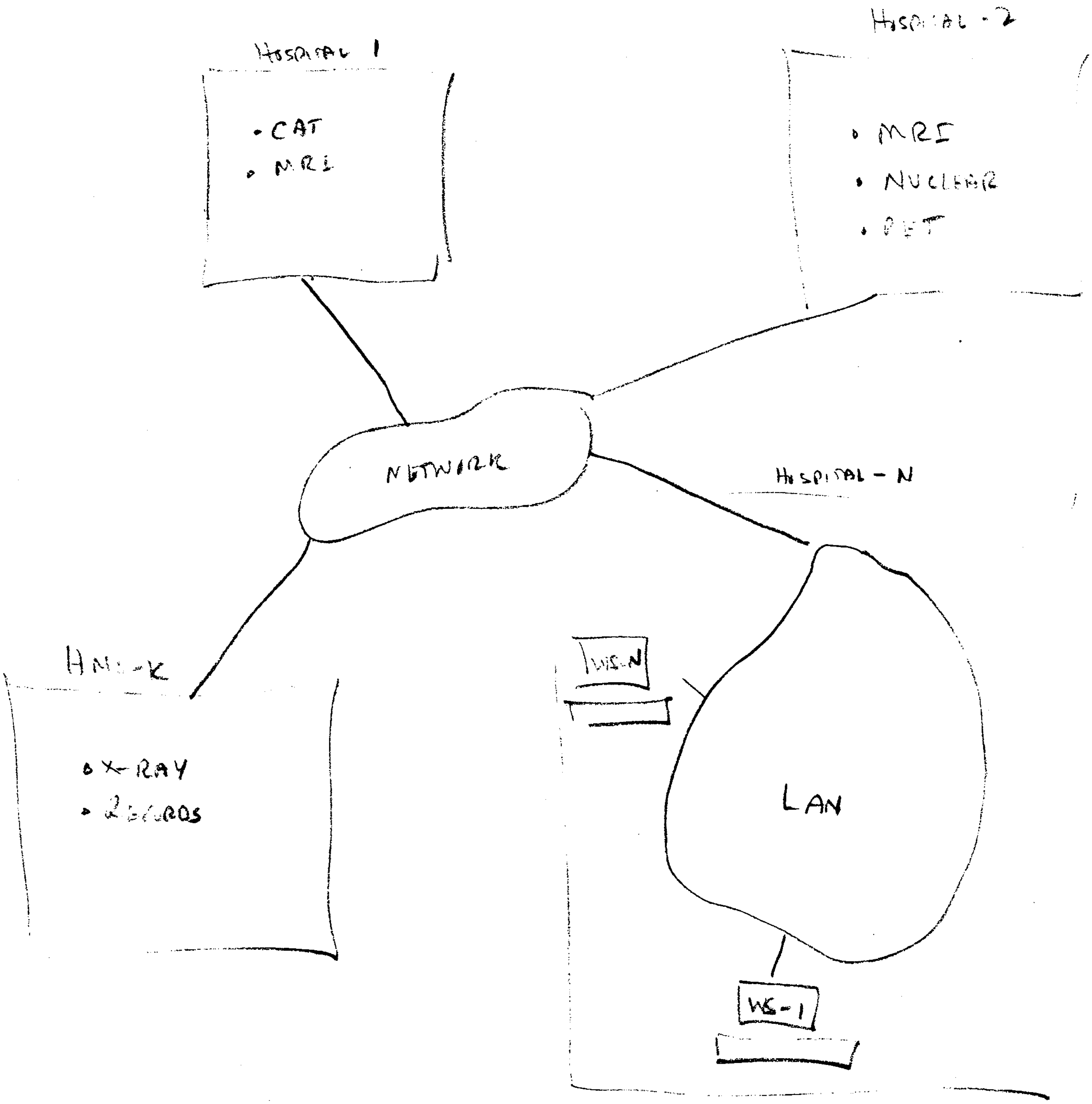
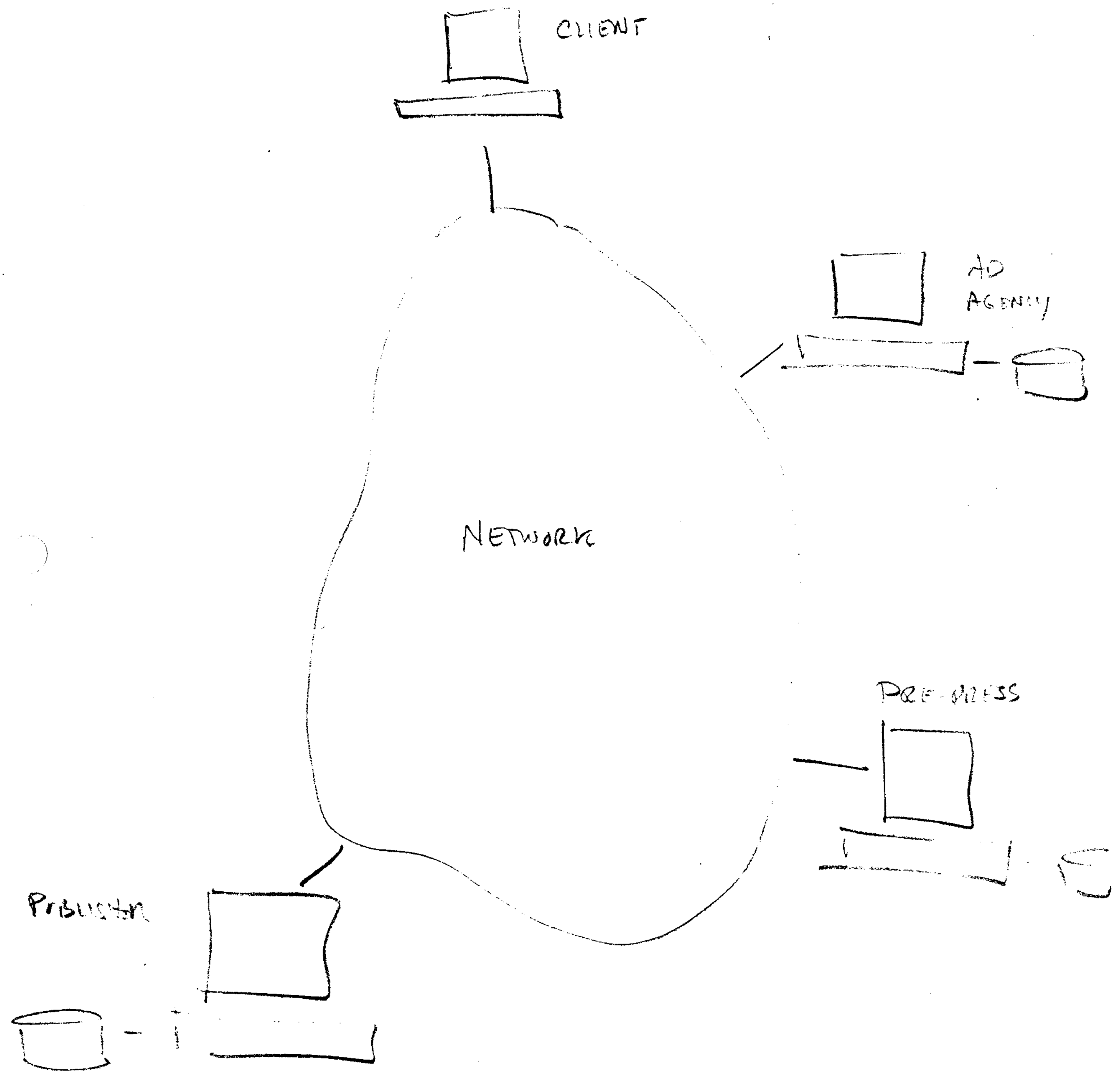


Figure 8.7 c Image Bank Storage Example



The communications issues in a distributed environment revolve around the need to provide a seamless connection between users in such a way as to make the presence of the communications network appear to result in minimal system impact.

Communications in a distributed environment has evolved over the years. As we have shown in Chapter 5, there are standard protocols to handle this communications issue. hate we wish to show in this chapter is that the interaction of database, operating systems , processors and processes may require a modification of such standard communication architectures. In particular the time required of all the protocol overhead may be prohibitive in the operation of a fully distributed system.

Figure 6.6 Distributed Communications

We can combine these separate elements and show how they interelate. This has been done in Figure 6.7. The specific example that we have used is that of a Hospital information management system for the care of patient records and one that connects multiple teaching hospitals into a single network.

Figure 6.7 Interrelationships of Distributed Elements

6.1.1 Network Requirements

The network requirements focus on the needs for the network to,provide more than just a data path. The requirements are driven by the high data rates and bandwidth available to the end user and the ability to configure the channel in a more fluid fashion.

Figure 6.8 Distributed Network Interfaces

Figure 6.9 Multiuser Distributed Environment and Processors

Figure 6.10 Single User versus Distributed User Interfaces: Logical and Physical

Figure 6.11 a Example of Financial Trading System Distributed Environment and the Total System Interfaces

Figure 6.11 b Example of fully Distributed Medical Records Keeping System

6.1.2 Session Requirements

The session was the key element of the multimedia multiuser communication -paradigm. Sessions have been discussed in detail in their many element. In this section we shall develop the overall requirements on the session concept when it is impeded within a fully distributed system.

Figure 6.12 Session State Organization: The Distributed Environment

Figure 6.13 Session Data Flow and Control

Figure 8.12 Session State Organization: The Distributed Environment

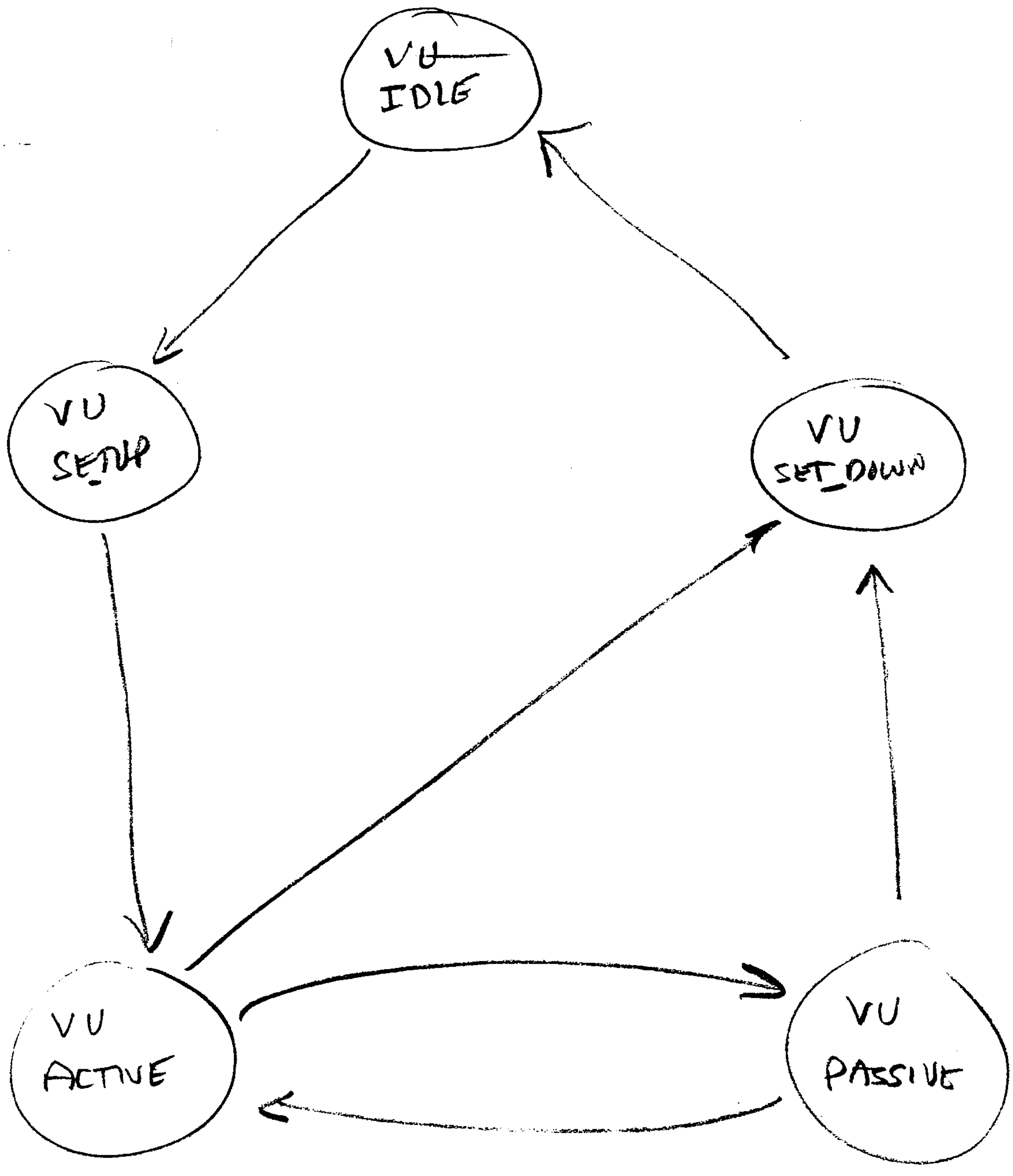
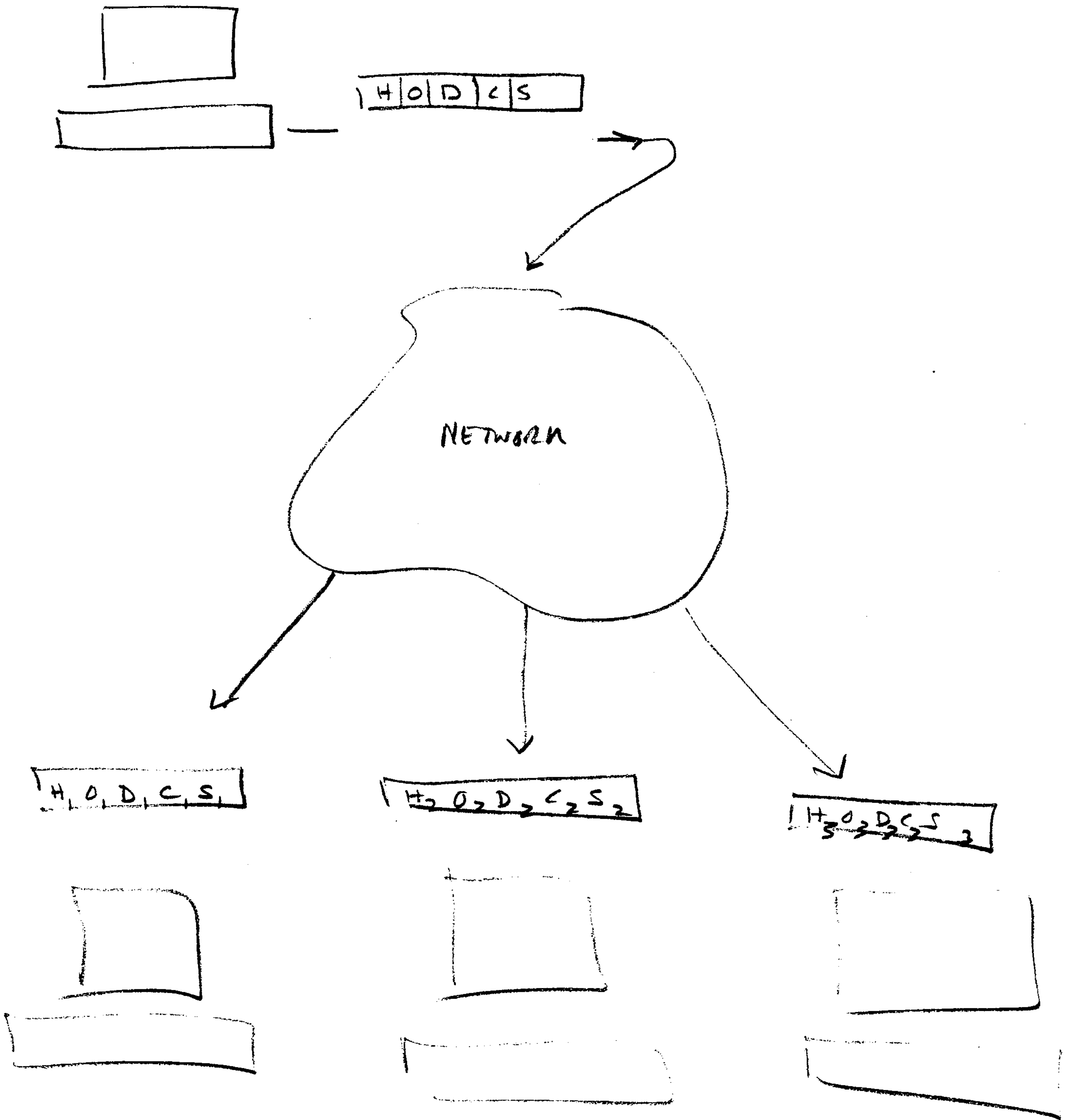
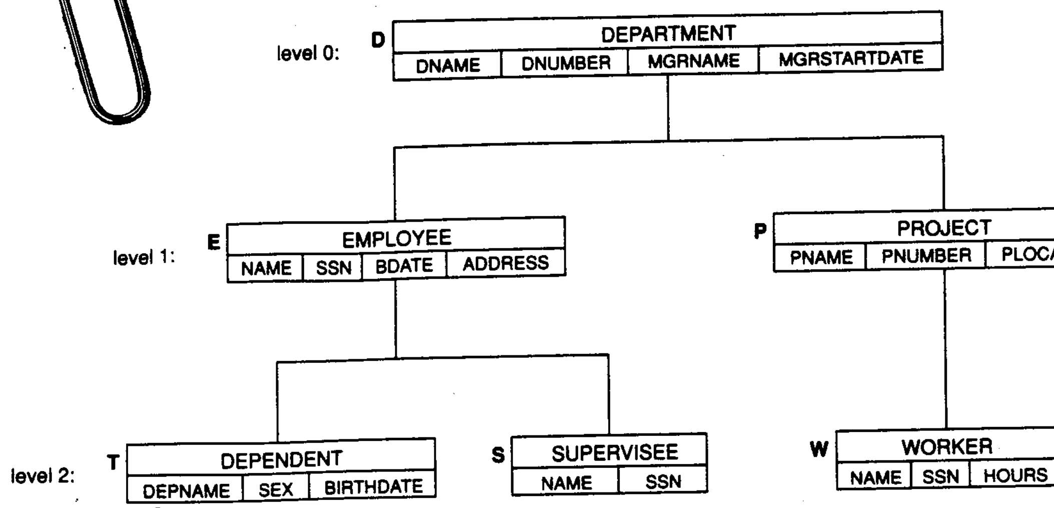


Figure 8.13 Session Data Flow and Control





[redacted].5 Hierarchical schema for part of the COMPANY database

have **level** zero. The level of a nonroot node is one more than the level of its parent node, as shown in Figures 10.5 and 10.6. A **descendent** D of a node N is a node connected to N via one or more arcs such that the level of D is greater than the level of N. A node N and all its descendent nodes form a **subtree** of node N. An **occurrence tree** can now be defined as the subtree of a record whose type is of the root record type.

The root of an occurrence tree is a single record occurrence of the root record type. There can be a varying number of occurrences of each nonroot record type, and each such occurrence must have a parent record in the occurrence tree; that is, each such occurrence must participate in a PCR occurrence. Notice that each nonroot node, together with all its descendent nodes, form a **subtree**, which, taken alone, satisfies the structure

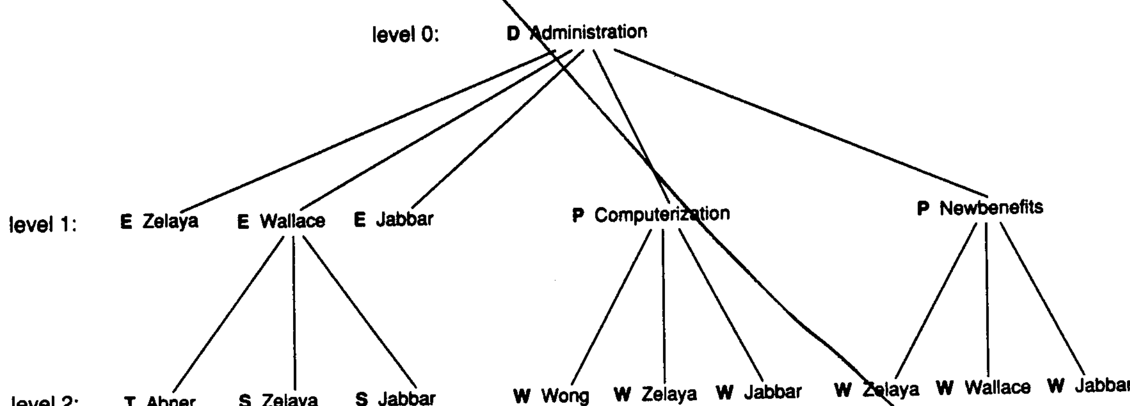


Figure 10.6 A hierarchical occurrence (or occurrence tree) of the hierarchical schema in Figure 10.5

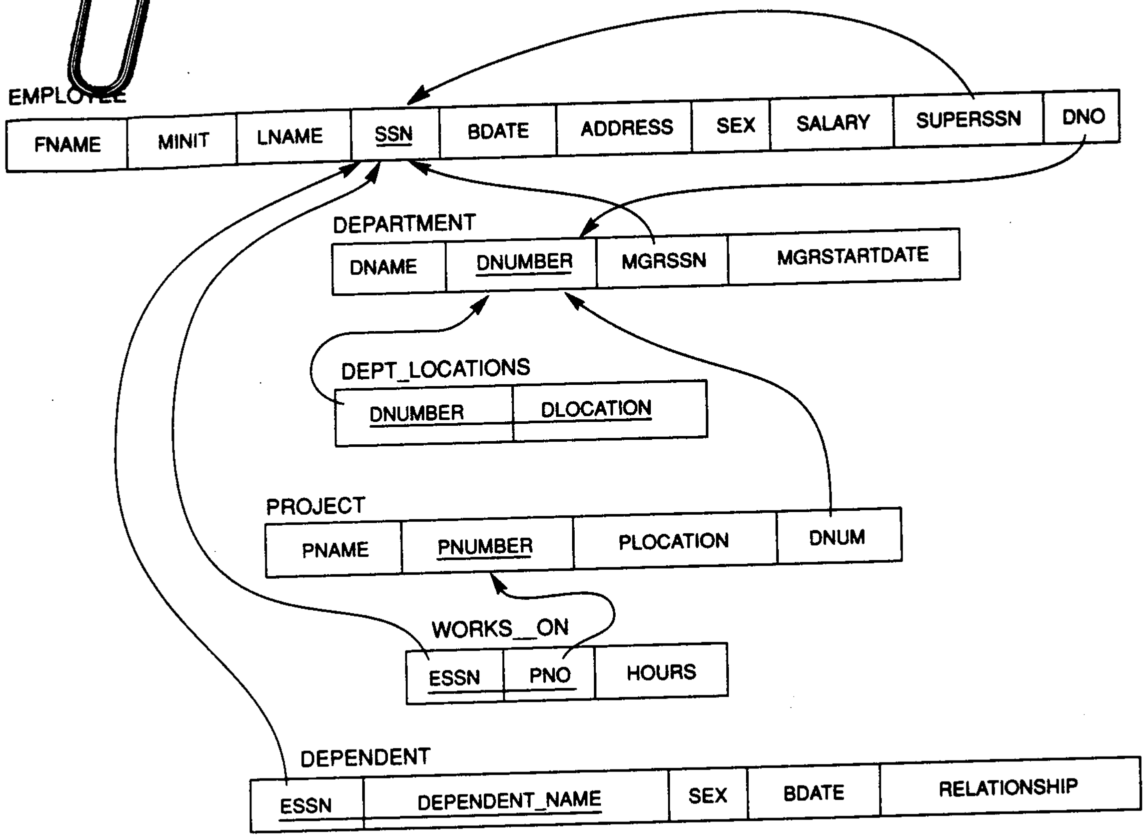


Figure 6.7 Displaying referential integrity constraints on the COMPANY relational database schema

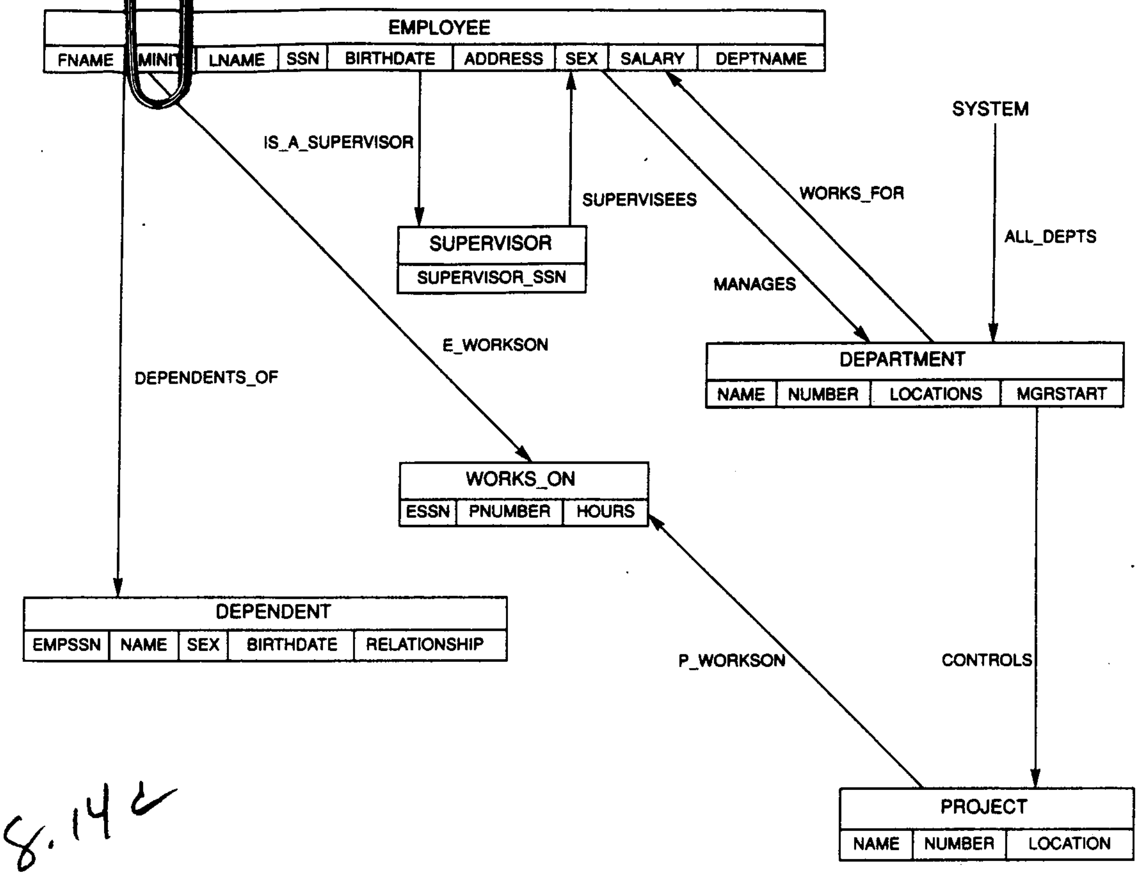
check that the integrity constraints specified on the relational database schema are not violated. In this section we discuss the types of constraints that may be violated by each update operation and the types of actions that may be taken in case an update causes a violation. We use the database shown in Figure 6.6 for examples and discuss only key constraints, entity integrity constraints, and the referential integrity constraints shown in Figure 6.7. For each type of update we give some example operations and discuss any constraints that each operation may violate.

Examples of the Insert Operation

1. Insert < 'Cecilia', 'F', 'Kolonsky', '677678989', '05-APR-50', '6357 Windy Lane, Katy, TX', F, 28000, null, 4 > into EMPLOYEE.

   —This insertion satisfies all constraints so it is acceptable.

2. Insert < 'Alicia', 'J', 'Zelaya', '999887777', '05-APR-50', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 4 > into EMPLOYEE.



8.14

Figure 11.10 A network database schema for the COMPANY database

For the COMPANY schema of Figure 11.10, if an interface between PASCAL and the network DBMS were available, it could create the PASCAL program variables shown in Figure 11.11. A single record of each record type can be copied from or written into the database using the corresponding program variable of the UWA. To read a record from the database, we use the GET command (see Section 11.2.3) to copy a record into the corresponding program variable. Then we can refer to the field values to print or to use for calculations. To write a record into the database, we first assign its field values to the fields of the program variable and then use the STORE command (see Section 11.2.3) to actually store the record in the database.

Currency Indicators

In the network DML, retrievals and updates are handled by moving or **navigating** through the database records, and hence keeping a trace of the search is critical. Currency indicators are a means of keeping track of the most recently accessed records and set occurrences by the DBMS. They play the role of position holders so that we may process new records starting from the ones most recently accessed until we retrieve all the records

6.1.3 End User Requirements

6.1.4 Call Processing Requirements

6.2 Databases

Databases are key elements for the storage of the multimedia data elements. The database issue has received considerable analysis and development for the standard computer files that exist today. These are typically local data bases that are accessed either locally or remotely. The key issue of multimedia communications is the need for both distributed database as well as multimedia databases. The issue of a distributed database is expanded now that we have the capability to transmit data at Giga bit rates. We can now reduce the access time per file for transport to a factor less than that for disk access. This opens new issues for the design of such databases. The multimedia data base represents a new and innovative method of storing multimedia storage. It has two dimensions. The first is the fact that the data may be stored on different media themselves and that may require a nonhomogeneous access and storage protocol. The second is that a multimedia element may no longer be bounded as simply as we have known such computer fils as bank records of income tax records. Thus the multimedia database issue is one of inhomogeneity and unboundedness. Both of these characteristics are typically lacking in standard data bases.

Figure 6.14 Database Schema: Hierarchical, Network, Relational

6.2.1 Database Structures

Databases are typically structured collections of data elements that are to be processed on an as needed basis by the end user. Quite simply, we may think of a database as a collection of elements that may be capable of representing the totality of information needed for a specific application. For example, consider the data base that may be needed for the distribution of Christmas cards. Such a data base may need the name, address (street, city, town, and zip code), the relationship (whose side of the family and the specific relationship, such as aunt), the status of sent or received from in the past year, and the gifts that may have been sent or received. What we see in this simple example is that each element of the data base is definable and also is definable in terms of a certain well defined alpha-numeric quantity.

In a multimedia database however, we do not have as simple a set of structures. Consider a similar example but for a multimedia case. In this example we are trying to track the progress of a patients recovery from a head trauma sustained in an accident. The characterization is that of patient John Doe. The data associated with this patient includes, X-rays, MRI scans, CAT scans, oral records of the radiologist and neurologist, written records from the primary attending physician, and the cardiologists electrocardiograms. What we immediately see is that the records are not so readily characterized as a finite set of alphanumeric data. The dat are video, voice, text, graphics and image data. The data in addition are of significant and varied length, and the relationship between the different media are suggested but not precisely structured.

6.2.1.1 Data Base Elements

A standard structure for database is the entity-relationship (ER) model. An entity is any real object that one desires to discuss and store dat on. For example, the patient may be en entity. Associated with any entity is a set of attributes. In the patient example, the attributes may be the name of the patient, the address, age , diseases, and attending physician. We can see that any attribute may be further extended as its own entity with its own attributes. Thus in the present case, we have the disease and it may be characterized by organ, name, level of involvement.

Attribute are generally of bonded form, that is the patients name, has a first, middle and last part, and each part is composed of a set of ASCII characters that do not exceed 50 in length. Thus the name can be at most 150 characters or 1200 bits.

Consider now a multimedia data base in the medical application area. The entity is the patients radiology report. The patient had a set of X-rays and an MRI scan. The "radiology report" entity has the following attributes:

- o X-rays, including 5 scanned images

- o MRI images, three sets of 12

- o Written report of radiologist

- o Written report of neurologist

- o Written report of pathologist

- o Oral report of radiological consult

- o Oral report of attending physician

In the case of a standard data base, the attributes have a finite set of values that they can take. These finite set of values are called the value sets or domains of the attributes. Names can be up to 50 characters for example. However, the domain of an attribute such as the oral report from a radiologist may be less well structured and in fact may be unbounded.

We can envision that there can be multiple entities in a large data base. Again consider the hospital data bases that we have been developing. Consider several entities:

- o Patient
- o Physician
- o Insurer
- o Procedures
- o Prescriptions

The patient entity may contain the attributes:

- o Patient

  - o Name
  - o Address
  - o Age

- o Insurer
- o Physician
- o Phone

The physician entity may contain the following attributes:

- o Physician
  - o Name
  - o Address
  - o Phone
  - o Specialty
  - o Backup Physician
  - o Specialty

We can now relate these two entities through a RELATIONSHIP called Patient_Of. Thus we have:

PATIENT Patient_Of PHYSICIAN

This shows that there are two entities that are related through the relationship. This types of relationship is called a binary relationship because it relates one entity type to one other entity type. There are relationships, that is action statements that relate one entity to another, that ca relate multiple entity types. Those that relate three entity types are called ternary etc.

Consider now a set of three multimedia entities that are prepared for a patient. They are:

- o RADIOLOGY REPORT

- o PATHOLOGY REPORT
- o CARDIOLOGY REPORT

Each of these reports contain attributes the are text, image and voice. We can also have standard entities that can be used in a normal data base. These entities are:

- o PATIENT
- o PHYSICIAN
- o INSURER

Consider now two relationships:

- o Requests_Report
- o Prepares_Report

We can now have the following statements:

- o PATIENT Requests_Report RADIOLOGY REPORT
- o PHYSICIAN Requests_Report RADIOLOGY REPORT
- o INSURER Requests_Report RADIOLOGY REPORT

Thus we can see that data base are merely ways of storing recorded facts, called attributes, in well defined bundles called entities, and associating them with each other in relationship. The structure of the language that uses the data base is called the data base language and has its own syntax. However, the syntax is governed by the underling structure of the database.

6.2.1.2 Relational Databases

A relational database is a means of characterizing the data and structuring it so that it may be accessed and altered in a fashion that allows for maximum flexibility. In a relational database, we develop a flexible set of relations and build these from the bottom up to have as flexible and accessible format as possible.

The basis building blocks of a relational database are as follows:

o Domain of Atomic Values: These are irreducible entries that represent final data values, such as the age of a patient, their first name, or their phone number.

o Attribute: This is the name or role played by a particular domain. Thus phone number is the attribute name played by the domain xxx-xxx-xxxx.

o Relation Schema: This is a set of attributes, {A1,...,An} that has a name R, which may be the term patient, and has a degree equal to the number of attributes, or irreducible elements.

o Relation: A relation is a set of tuples, where a single tuple is the set of attributes of the relation. For example, the relation schema, address may contain, the attributes, number, street, city, state, and zip. This is a 5-tuple. The relation ADDRESS now is the set of all such tuples that are relevant to this application. It could be zero, one or even one million addresses.

o Tuple: As we have just noted this is a specific relation entry. In the above case for address, we have:

t = {24, Wood St, Greenwich, CT, 08534}

or for the general case:

t = {number, street, city, state, zip}

and the relation:

ADDRESS = {t1,t2,.....,tn}

o Key: The key of a relation is an attribute that can be used to uniquely identify the relation. In the case of address, we may use the zip as that attribute if and on if we know that people are in one town at a time. There may be many keys in relationships, and there can always be a primary key.

We can now see how relations can be referred to one another. Let us define three relations through the three schema:

o R1(A1,A2,A3,A4,A5)

and r1 = {t1,t2,t3,t4,t5,t6,t7,t8,t9}

where ti are all 5-tuples

o R2(A6,A7,A1)

and r2={u1,u2,u3,u4}

where ui are all 3-tuples and A1 is the same as in r1.

- R3(A8,A5)

  and r3={v1,v2,v3,v4,v5,...,v114}

  where vi are 2-tuples and A5 is the same as r1.

We can now point from r3 to r1 and from r2 to r1. This pointing is through the common attributes in both cases.

There is an algebra that can be developed using the relations that entails the use and definition of operations on the relational schema. These operations are of the type:

- Union
- Difference
- Select
- Divide

and many others (see Elmasri and Navathe)

6.2.1.3 Hierarchical Databases

The hierarchical data structure is an older scheme of database and follows the need for a structured environment. In the relational case we define general relations, many of which can change and evolve in time and are related to one another through keys. In a hierarchical scheme, we have a much more structured system that is related a priori.

In the hierarchical scheme we have several elements that make up the data base paradigm. These elements are:

o Record: This is the collection of data that are the basis of the database. The record could be the patients account and would be composed of name, address, age, etc.

o Parent-Child Relationship: This is the relationship between one record and other records. For example, we can say that Department (eg Cardiology, Radiology, Urology etc) cane be the highest record in the hospital. This may relate down to Patient and Physician. In turn, Patient relates down to Tests and Insurer records.

o Field Type: This is the detailed contents of the Record type. In the patient Example we may have:

o Patient = {Name, Address, Age, Disorder, SS Number}

The patient Tests record may include the following Field Types:

o Test = {Type, Date, Physician, Result, Next Test Date}

Hierarchical schemes have a fast access time for most applications but have many major drawbacks. It is generally difficult to change data elements or records and we have to be cautious in handling non 1:N relationships in the parent child area. In addition, as we have noted in this example, the patients are grouped by department first and to get statistics on patients, we must sum across departments, not across patients.

The patients, and tests etc are segmented by the hierarchical scheme.

6.2.1.4 Network Databases

The network database structure owes its early existence to Bachman and it represents a way to structure the relationships between data records in a logical fashion on the overall data structure. In the hierarchical scheme we had developed the notion of an overlaying hierarchical relationship between all elements of the database. In the relational scheme, we had much greater freedom in defining data elements and then after the fact overlaying relationships in a much freer manner. In the network scheme, we find that there is a middle ground in which we do not have the hierarchical structure but that there is an overriding relational structure in the database that is called the networking of the data elements.

The network data base is composed of the following elemental concepts:

o Data Values: These are the actual numerical or letter values that are entered into the data base. Example would be specific names or numbers.

o Data Item: This is the name ascribed to the data value. For example, we can use NAME, or ADDRESS.

o Record: This is a collection of data values in an organized form. It may represent all of the data values stored on a particular patient.

o Record Types: This is the name used to identify a collection of records, such as PATIENT or PHYSICIAN.

We can then find a way to relate these together through the following concepts:

o Set Type: A relation between two record types. One of the record types is called the owner and the other is the member. There is only one owner and there may be several members. A set type may be the one Treated_By and it relates PATIENTS to PHYSICIANS.

We can see an example of a typical network data base in Figure 6.x.

6.2.2 Data Base Access

The structure of databases is only one of the key factors in understanding the usage and implementation of a database. The second element is the access to the data records themselves. In this area there are three major issues of concern. The first is the issue of the physical access to the data.Specifically what medium is the data stored on and how is the data physically stored on that medium.As with the file structure, we consider the more common forms of file structures in this section and defer the issues of multimedia databases to latter sections.

The primary form of data storage is the magnetic disk for linger term storage and random access memory for shorter-term storage. The issue of medium is significantly impacted upon by the cost of random access memory and its ability to be supported is a nonvolatile environment. With the advent of multi-megabyte chips and the dramatic cost reduction in cost per bit, there are many tradeoffs that can be made in the selection of physical storage medium strategy. We shall not discuss those elements in this text but refer their reader to XXX.

The second issue that we shall discuss in this section is the issue of the logical access to databases. Specifically, what languages do we use to readily access the information. The languages typically used here are called query language and typical example are those of SQL and the other languages that have developed syntaxes for access of structured databases. The query language is the user interface issue that we have discussed at length in other chapters. The query language has develop for structured database from a set of typically arcane commands to a set of commands that today resemble a natural language. It is actually possible to ask such a question as "How many people are there in the Drafting Department". A system may take this sentence and parse it and then know enough to be able to recognize that it must add together all the name files for a specific department.

6.2.2.1 Physical Access

6.2.2.2 Logical Access: Access Languages (DBL)

6.2.2.3 Database Access Interfaces

Figure 6.15 Data Base to Operating System Interface

Figure 6.16 Example of Financial Database Access

6.2.3 Distributed Databases

Distributed data bases are essential in environments where there would normally be significant communications from one location to a distant one. A typical example that w have used is the need for distributed database in the financial transaction processing area. Here we have to keep track of many transactions that occur at various locations.

Let us first consider an example of such a distributed system of databases that is slightly different. This example is that for the storage of 800 numbers in the telephone system. In this example we will first deal with a passive database that is read by the user and there is no transaction made in the database.

The 800 number system in the telephone network works on the principle that local users can dial 800-xxx-xxxx and can get a toll free phone call. The normal billing system will recognize a prefix other than 800 as one to bill and will bill the calling party directly. When the 800 number is referred to in a call, the number is looked at in a database table and it is cross referenced to an actual area code number and that number is fed into the network as the dialed number. At that point the end user does not know that he has skipped around the billing computer and has had his dialed number converted. The conversion table could be placed at multiple locations or it could be centralized. In this case the database is static. The difference between having this data base centralized or distributed is the issue of call set up time, that is the time to place the call.

Let us examine the 800 database tradeoff. Consider the two cases of centralized versus distributed. In the centralized case we have a large data base that is at one location. Let us assume that there are 1 million calls per busy hour and that each 1 thousand calls requires a 1 MIP processor. The call set up time for this system is given by:

$$T\ SU = T\ TR + T\ I/O + T\ pro + T\ Resp + T\ TR$$

Here T TR is the transport time, T I/O is the I/O time and T resp is the cpu response time. For a heavily loaded machine we have decreased I/O and response time.

In the 800 distribute case we assume that we can place an 800 processor at each of the 1,000 central office locations and this yields 1,000 MIP machines and reduced I/O per machine and reduced processing time. In addition we have to split the database in such a way that we do not have to multiply it 1,000 times. We do this by recognizing that only x% of the 800 numbers are used by y% of the people in each of the locations. This is often called the 80/20 rule. Thus we can save only a small percent of the conversions for translation and leave the rest for a centralized system or route them to other systems.

Let us now consider a second example that is both read and written into in a distributed fashion. This data base is one used for the buying and selling of commodities such as gold, oil, silver and other such items. This differs from the first case because now there is a dynamic bidding process that must carefully take into account the timing of the transactions and assure the end user that bid and asked prices for commodities are met.

Let us examine the process in some detail. There are sellers of commodities that present an asking price. For example Seller 1 could say that they are willing to sell gold at $450 per ounce. There are several buyers who are interested in the gold. However they are not willing to pay the asking price. One bids at $448 per ounce as the highest. Then the seller reduces the price to $449. One of the buyers sees this as an opportunity and bids $446.50. The other buyer seeing that this is a good deal bid $449.125 and the seller closes the deal.

Now there any be several markets for this gold. The markets may be in New York, London, Tokyo, and Hong Kong. One approach is to have the data base used for recording and display the bids to be centrally located in a city such as Chicago. Or alternatively, each city may have its own data base and the system communicates amounts itself in a fully distributed fashion. Again the tradeoff is time to transact and time to respond. In this type of market, the quicker I can respond the better closure on the bid/asked price one can obtain.

A third example of a distributed database is that which can be used for the establishment of a multimedia communications system, where sessions are established and users are sending images and other multimedia objects from one point to another.

In this case, the system is used to set up calls and to maintain the multimedia sessions. The session set up algorithm identifies all of the participants in the session and knows their locations on the network. When a member of the session transmits a packet to a local node, the local node accesses the overall database which knows the status of all participants in the session. It then generates a set of additional packets, customized as is necessary and transmits them to the desired nodes. It must assure that there is certain synchronization in the packet transfer and that all of the packets are synchronously transmitted both interpacket and interuser on a single packet.

In these three example there are certain characteristics that arise that a distributed database must reflect. These characteristics are:

o Reduced access time due to the combination of reduced communications time and reduced I/O time due to more efficient loading per processor.

o Real time updating of data records and assurance of timing factors so that bid and asked numbers are not reversed in time.

o Overall synchronicity within and between packet transmissions to assure end to end synchronicity for display and processing purposes.

6.2.4 Distributed Database Issues

We can now begin to discuss some of the structural elements of a distributed database. We shall use the relational database system for the representation of database in this environment. Approaches using the other schema are also possible and are left to be developed in the problems. The discussion in the section is based upon the work by Ceri and Pelagatti.

Recall that a relation, R, is a table that store data. The data is stored in columns called attributes, Ai. The number of entries in the table of the relation is called the tuples in the data entry.

In the normal non-distributed environment, we have a collection of relations that generate the overall database. The global schema is this collection of data as if we had no distributed environment. Now we ask ourselves, how do we split this up into several pieces so that we may locate the data in multiple locations. We do so in two steps:

o Fragment: In this step we take the global schema and generate separate disjunct schemas call fragments and named R1,....,Rn.

o Allocate: We assume that the database is to be distributed to k locations, L1,...,Lk. We then must map the fragments onto these locations. Many fragments may be mapped onto several locations.

We can deal now with the first issue of the allocation of the databases to many locations and focus first on the issue of reading of such databases. Consider the database shown in Figure 6.x. It is composed of many sub elements. We desire to distribute this database over many locations. Let us take the example of the 800 number database that is used in the telephone business. This database is a conversion data base that converts the 800 number into a real dialable number of the type xxx-yyy-zzzz. Rather than having this database at one central location and having all customers dial that location, we can break this database apart and distribute it at many locations. The breaking apart function is the fragmentation task, and the placing of these parts at multiple locations is the allocation task.

Figure 6.x Distributed Database Fragmentation and Allocation

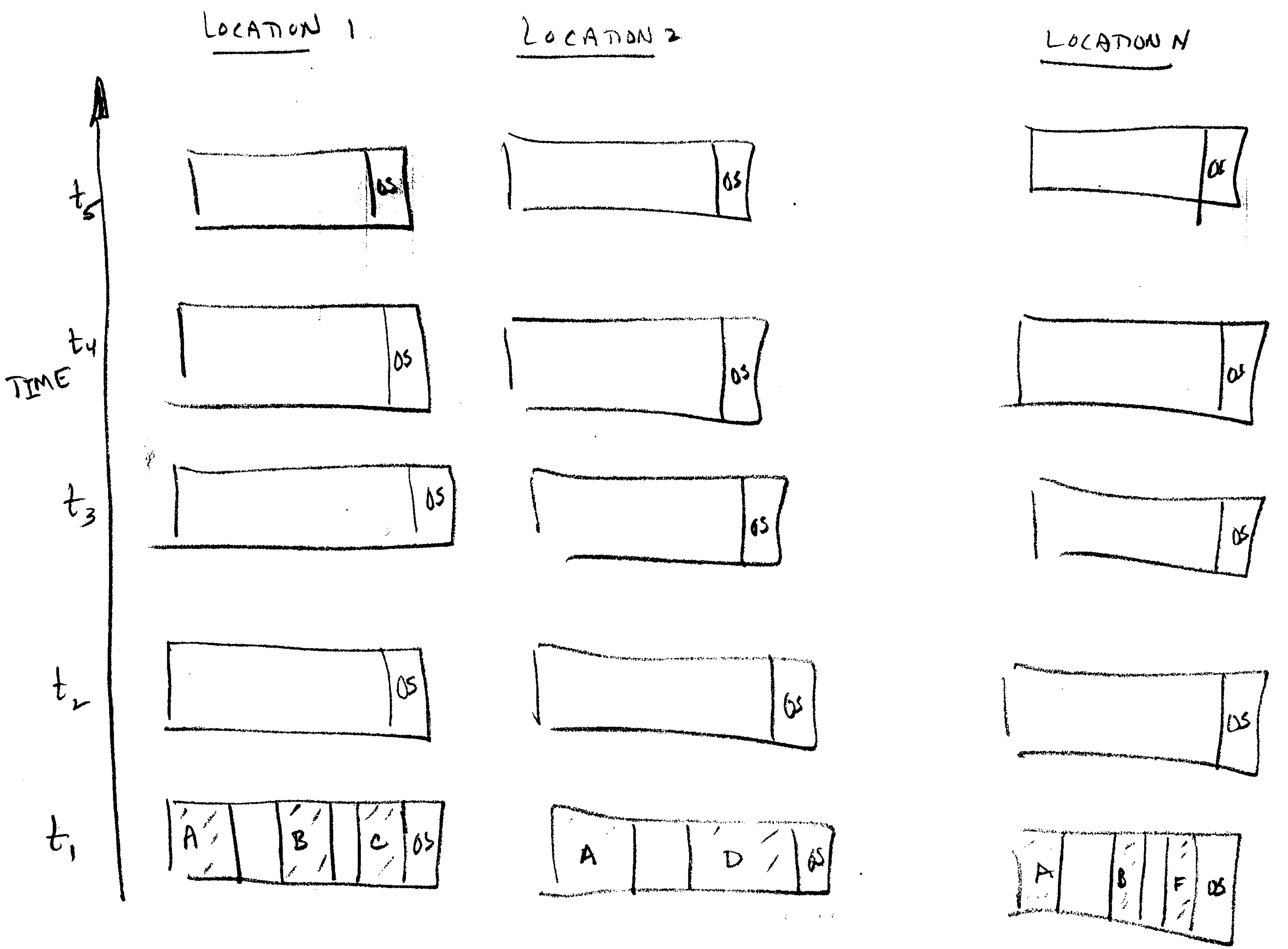


Figure 8. X Distributed Memory Management

In fragmenting a database, we are considering the the need to break it apart into smaller segments that can be accessible in some particular fashion. In the 800 number case we may segment by other areas of the country, by the specific application, by the type of business or other reason. When we fragment, we than may want to assign or allocate the fragments to multiple locations.

Thus if we define D as the total database, we can define D as:

$$D = \{D1, D2, \ldots, Dn\}$$

where Dk is a specific fragment. We can then define Li as the ith location. Let Si be the assignment vector for the ith location;

$$Si = \{Si1, Si2, \ldots, Sin\}$$

where :

$$Sik = \begin{cases} 1 \text{ if } Dk \text{ is at } Li \\ 0 \text{ otherwise} \end{cases}$$

Then we have the database at Li being the following:

$$D(Li) = D^T Si$$

where T is the transpose operator, and we represent it as:

$$D(Li) = \{Si1D1, \ldots, SinDn\}$$

The optimal segmentation and allocation problem can now be stated as follows. How do we choose the best fragmenting and allocation scheme to either maximize or minimize some specific performance criteria. One approach is to define the cost of a scheme and then to find the fragmentation and allocation that minimizes the cost. Consider the following cost equation. Let CTOT be the total cost of the scheme. The cost equals:

$$CTOT = CDB + CPRO + CCOM + CLAB$$

where:

CDB is the cost of the actual storage databases at each location and duplicated as needed at each location.

CPRO is the cost of the processing power needed at each location to manage the database entry and control.

CCOM is the communications cost of the system as needed for the users to access each of the fragmented databases.

CLAB is the labor cost per access, summed over all access. It is a measure of the response time of the system in providing a unit access to a datafield.

In Figure 6.x we have depicted the tradeoffs available in each of these four cost parameters. Let us discuss some of the qualitative tradeoffs.

Figure 6.x Cost Tradeoff parameters: Fragmentation and Allocation

o Cost for Memory: As we fragment the database, we then allocate it to many locations. The cost for memory is dependent on how much we place at each location. At one extreme, we can fragment and place only parts at one and only one location. At the other extreme we can fragment and place all parts at all locations. The latter will be an upperbound onmemory costs and the former a lower bound. Depending on the cost of memory the cost may be significant. The memory cost can be made lower by a slower memory storage system, however, that will increase the labor cost element.

o Cost for Processing: As we have done for memory, the processing cost is dependent upon two factors, the number of accesses per location and the amount of memory controlled per location. If we have few locations then we are dominated by access, if we have many locations but significant duplication of the database, them we have significant access processing.

o Cost for Communications: This factor depends on how many locations there are and how many locations have the data base elements. If there are many locations and many data base elements then the communications costs are low. If there is only one location then we would expect high communications cost.

o Cost of Labor: The labor costs are based on how many people are required for a volume of work. The people are determined by the number of units and the holding time per person. If the overall processing time is dominated by the response time of the human then this is the lower bound. If, however, the response time is dominated by the system delays, then this is a factor. For example, if we have log communications delays, long access times etc, we have large labor costs.

Conceptually, we can conceive of an optimization criteria that says the if this cost is to be minimized, we could then determine the optimum fragmentation and allocation to reach the minimum cost element. Generally this is not readily determined and suboptimum approaches are performed. We demonstrate several of these in the problems.

We can now consider the simple second issue that relates to the real time updating of databases. This is the issue of writing or reading and writing databases. In the previous analysis we have developed an approach to the selection of how to distribute the database based upon the reading of the files. In this area we shall focus on the issues of actively changing the contents of the database. There are two major issues that are key to understanding the active change of a distributed database. The first issue is that of recovery from faults and the second is the concurrency of a database use. Recovery relates to the issue of haw to deal with system errors and how to restart the database after such a failure. The issue of concurrency relates to the need to establish the concept of a transaction, which we shall do shortly, and show that transactions are key elements in read write actions.

Figure 6.x depicts the issue of recovery of the database after a failure. The distributed database is composed of three levels of elements. At the highest level is the data agents that control the flow of the actions of the database. The agents have a define root agent that provides an overall centralized control element. At the second level is the distributed transaction monitors DTM that provide for the communications amounts the different locations. The DTM provide control of the elements in the third level, the Local Transaction managers, LTM. The DTM elements communicate amounts each other and also sent the action signals to the LTM.

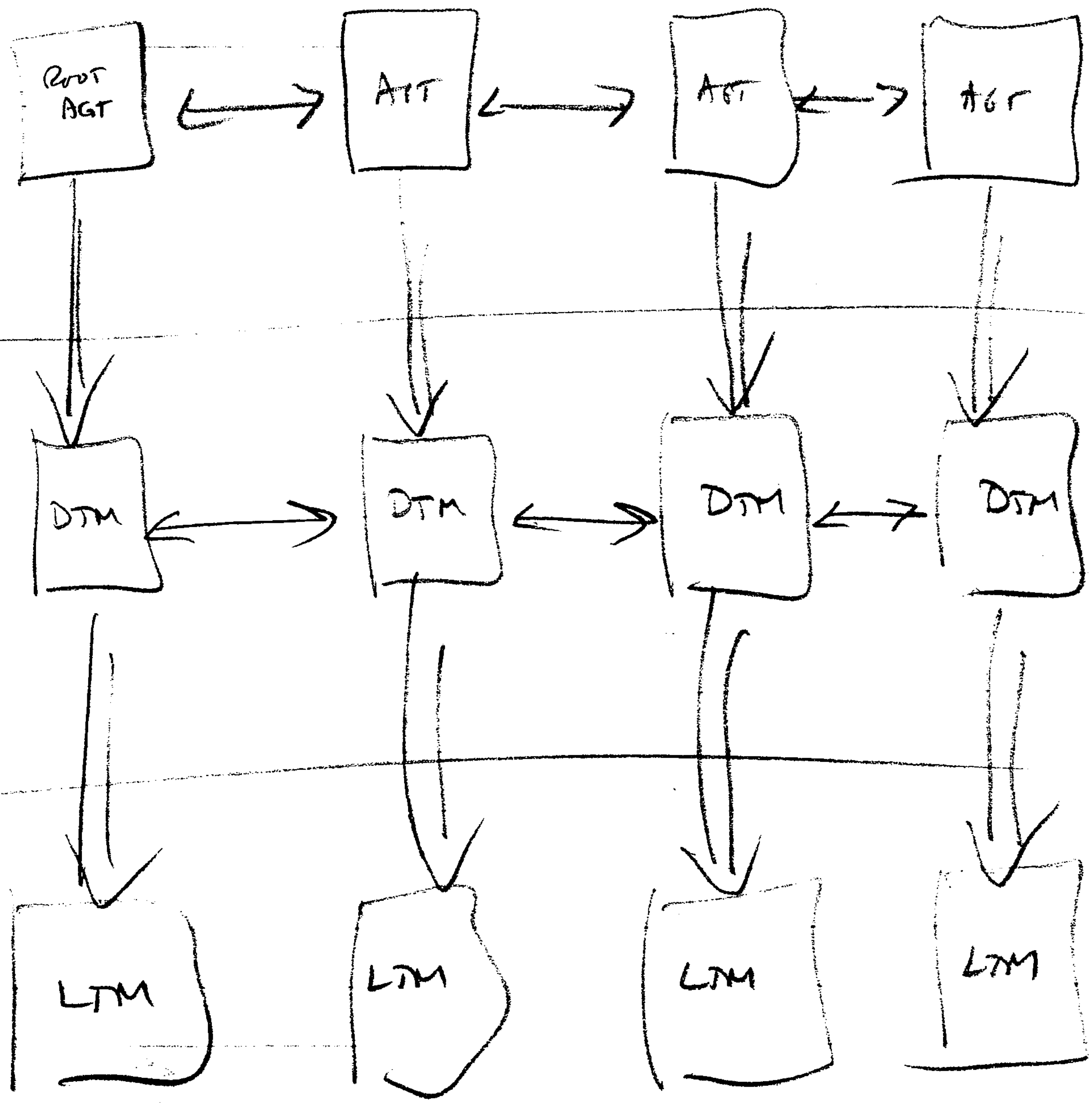
The recovery issue relates to a transaction that is occurring that may be halted in mid stream. If this is the case, it is necessary to remember that it was not completed and possible, but not necessarily, where it left off. It is the function of the root agent, in conjunction with the TMs to keep logs of the status of the transactions. The complete transaction has to be defined and the progress of the transaction is followed through a log process, If a fault occurs, then the log is used as a means to determine if the system is recoverable.

Concurrency is the issue that relates to ensuring that one event does not occur before another that id dependent upon the first. We would not want to post money to an account before we checked the validity of the depositing check, and if posted allow the customer to withdraw the amount not yet checked.

Figure 6.x depicts the issue of concurrency control. As we had done with recovery, concurrency depends on the three layer architecture. Now however, we need to have the agents talk with one another and with the dTMws. We perform the concurrency control through a locking mechanism, combined with a structured and time tagged transaction sequence. The transaction is a complex set of reads and writes, that must be defined a priori. By tagging the transaction sequence as a whole, and providing locking mechanism throughout the database schema, we can assure that improper reads and writes do not occur. The cost may occur in the efficiency of the database. access time.

Figure 6.x Concurrency Control

Concurrency



- Lock _ shared
- Lock - exclusive
- Unlock

{ ◦ Local - S, E, U
{ ◦ Dist -

Ref Ceri & Pelagatti

We can now define a transaction as an atomic unit of access to the database. This transaction is complete by completing all of the elements of the access or otherwise it is not complete. We can define Tj as a transaction. We further define a transaction as a combination of reads and writes to a data base. Let W denote the write function or operation and R the read operation. Let Dk represent the data item on which the operation is to be made. We call Dk a dat item and say that Dk is part of relation Rj. A data item is that set of a relation that is materially affected by an operation of a read or write.

We can now define a transaction as an ordered tuple of W and R operations as follows:

$$Tj = \{ Rj(Dk), Wj(Dm), \ldots\ldots, Rj(Dp) \}$$

That is Tj is a collection of ordered reads and writes, at locations and on data elements k,m,...,p.

Transactions have four basic characteristics that must be observed:

o Atomicity: A transaction is the totality of all of its parts and cannot be broken down into sub elements.

o Durability: Any system that performs a transaction must assure the operators that the results of the transaction endure.

o Serializability: When there are several transactions executing simultaneously, them the result must be the same as if the transactions occurred in a hourly serial manner. That is if there are three transactions T1, T2, and T3, then if we have:

T1 = [W1,W1,W1,R1,R1]

T2 = [W2,W2,R2,R2]

T3 = [W3,R3,R3]

We can now define a SCHEDULE, S, as a sequence of W/R commands and one serial schedule is the one;

S = {T1,T2,T3}

Another schedule is a sequence of mixed versions of Wi and Rk. Any admissible sequence form a schedule must be serializable, or equivalent to the first S.

o Isolatable: This means that incomplete transactions must not be effected or cannot effect other transactions.

We have seen that we ensure the necessary elements of a transaction are preserved by ensuring that the use of locks, time tags and the other elements that we discussed are used in a distributed database environment.

In the system at any one time there are many transactions that are occurring. That is there are Ti, Tj, ..., Tq. This clearly shows the problem of the distributed database. The issues are as follows:

o Concurrency: This relates to changing data elements at different places at different times, when there is a need to assure that a sequence of changes be made in certain order. For example Ti may have a sequence or ordered reads and writes on a set of data elements. Tk may have another set of reads and writes. However there may be an operation in Tj that may proceed an operation in Tk that will alter the way Tk is carried out. A simple example will give the case.

Let us assume that we wish to reserve a sat on an airflight. However another person at another city wants to reserve the seat on the same flight. Let us assume that you get to the counter in New York one minute before the contending person does in San Francisco. Hopefully, the system is first come first serve and you get that last seat.

Let us first examine the transaction that goes into reserving the seat.

o A read is made into the database to determine if a seat is available.

o A write is made into the database to reserve the seat on the plane.

o A read is made into the database to check the credit card.

o A write is made into the database to charge the card.

o A write is made to a data base to issue the ticket.

Thus the transaction is:

T1 = {R1(D1),W1(D1),R2(D2),W2(D2),W1(D3)}

Here we have W1 being a read at location 1 which is New York, and W2 being Phoenix. The second transaction is:

T2 = {R5(D1),W5(D1),R2(D2),W2(D2),W5(D3)}

Now these transactions are occurring almost simultaneously but the reservation system is in New York (1) and San Francisco (5). Let us see what could happen in this case.

o New York checks and shows available.

o San Francisco checks and shows available.

o San Francisco books seat.

o New York books seat.

o New York checks credit

o San Francisco checks credit

o New York charges credit

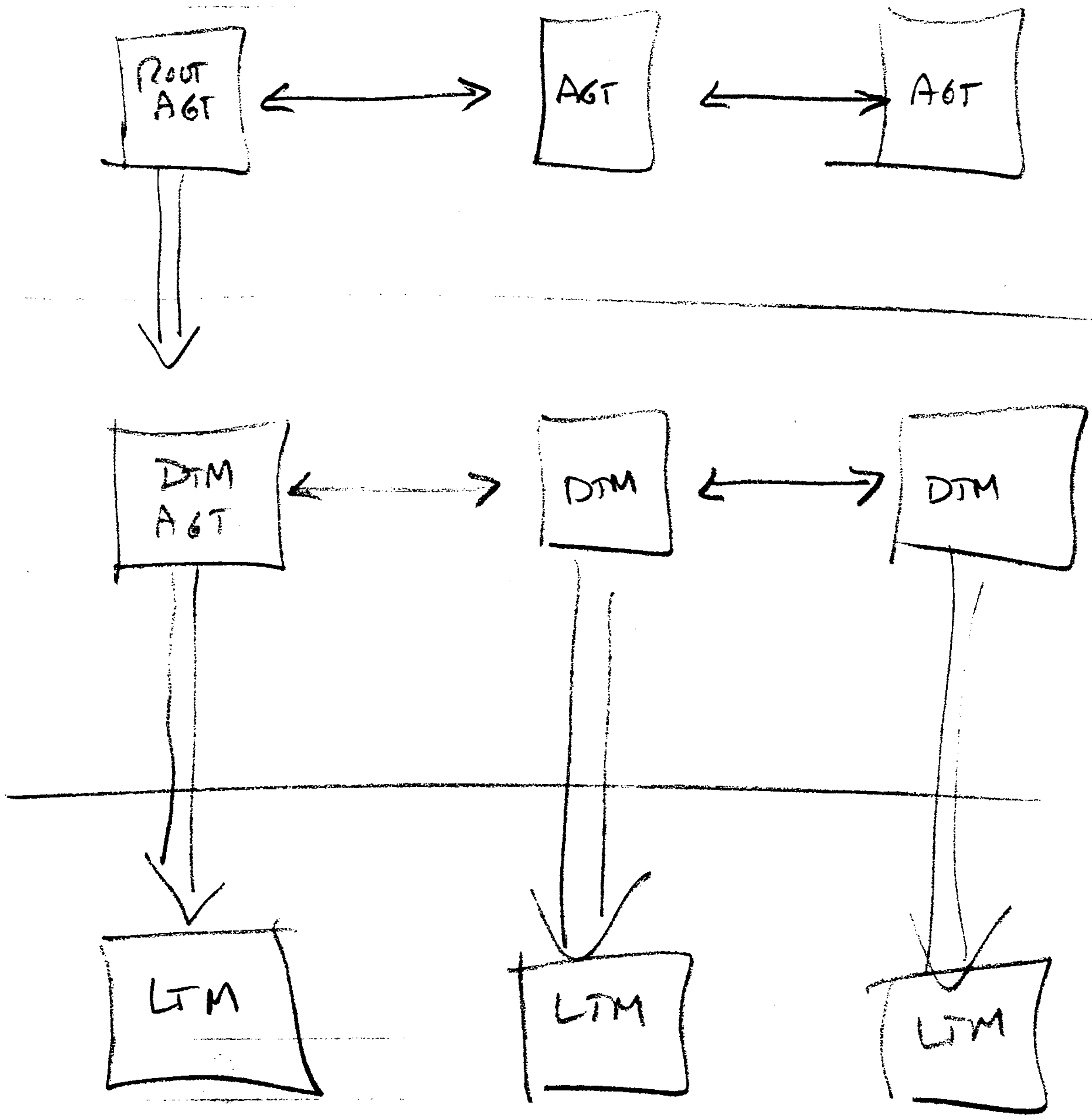
o San Francisco charges credit

o New York gets default on ticket print.

o San Francisco gets ticket

o You get very mad !!!!!

Recovery



- Begin
- Commit
- Abort
- Create

L — B, Co, A, CR
D — B, Co, A, CR

Fig

Ref Ceri & Pelagatti

Even though the dat bases update instantly, we still have not kept the integrity of the overall transaction. Admittedly this was a bad way to create a transaction, it however, is typical of what can happen in large scale transaction systems.

The solution of the concurrency problem is time stamping the total transaction and ensuring that all dat node controller have the ability to update each other simultaneously.

6.2.5 Multimedia Databases

The multimedia database consist of several media that contain a complex interconnection of video, voice data, text, image and other forms of information. As we discussed in the context of the reactions database, there can be significant connections between many of the different files. For example, consider a voice file that contains certain information on a particular patient record. This file may be associated with a video file of the patients echo cardiogram. It is necessary to ensure that the files are related to one another, that they can be tracked down and also synchronized in time and space. These types of requirements are not typical for alphanumeric records and files.

6.2.5.1 Physical Access

Let us consider the case of accessing the record of patients in a hospital. There are four types of record that may be of concern for the development of a certain type of procedure. These records are:

o Patient records stored in the Relational database disks.

o Radiological images stored in a CD juke box storage system and managed by a PACS type data manager.

o Text records stored in an image based CD system similar to the image files and front ended by a separate database manager.

o A voice storage system that is the storage vehicle for voice messages that annotate the image files.

o A video angiography of the patient's hear stored on a CD rom disk.

We can consider the multimedia data object as a meta-object which is the concatenation of the set of four separate data objects. Let us define these objects as follows:

o OR is the data object associated with the patients files

o OI is the data object associated with the patients radiological records. Note that this may be just x-rays, all x-rays, x-rays and MRI data, or a concatenation of many of these elements.

o OT is the collection of text records that relate to this file. It may include the write ups on the current x-rays prepared by the attending radiologist or any other set of elements.

o OV is the voice samples of the consulting radiologist or other related physician.

We can now defining the compound data object associate with the patient as OP and definite it as:

OP = {OR,OI,OT,OV}

where {.....} stands for an appropriate concatenation. We shall discuss the structure of this concatenation in detail latter in this section.

We can now see how we can relate these compound data objects t the physical system design by examining the overall system architecture. Figure 6.x depicts the overall architecture of this multimedia storage system.

Figure 6.x Multimedia Storage for Medical Applications: Physical Access

(a) Separate Object Access Mechanisms

(b) Single Front Ended Access Mechanism

In this figure, we depict the separate storage elements and assume for this case that they are generally located in close proximity. There are two extreme strategies for developing physical access to these records. The first assumes that each is a stand alone system and the end user mus establish all connections to related records.

6.2.5.2 Logical Access

The logical access issues for multimedia database are significantly more complex than for the alphanumeric databases. The SQL type languages for physical database are based upon the simpler structure of the records and files and make the use of simple syntax possible. Queries in a multimedia context are much more complicated. In an alphanumeric system it is possible to ask the question "How many patients in 1989 have had acute endocarditis?". The statement is parsed into the following elements:

- o How many: Count the number and the answer is an integer.
- o Patients: Count on the patient records
- o 1989: Condition on the year
- o endocarditis: Condition on the disease.

Consider now the type of question we may query the data base with in a multimedia format. "Which patients have shown a colcystitis diagnosis that the physician has expressed concern in the extent of the diagnosis and also the scans are not fully determinate?".

Let us examine this query and compare its elements to the query for an alphanumeric data base query.

6.2.6 Distributed Multimedia Databases

6.2.7 MMDB Design Factors and Issues

6.3 Conclusions

This chapter developed the concepts of databases and further developed concepts for both distributed database and multimedia databases. The database element is key to the storage and retrieval of multimedia files. Knowing the standard data elements in a typical record file is necessary but not sufficient for the development of a multimedia database. In Chapter 4 we had developed an understanding of the types of files that we could develop in a multimedia environment and studied to types of storage media available. That allowed us to develop the times statistics necessary for the operations of a data file retrieval. In this chapter, we have extended that to the database element, understanding the way the information can be stored and what languages can be used in its retrieval.

6 Distributed Data Buses

Chapter ~~8 Distributed Environments~~.

8.1 Environment Factors.

8.1.1 Network Requirements

'Connecting RPC-Based Distributed Systems Using Wide-Area Networks'
R. van Renesse, A.S. Tannenbaum, H. van Staveren, J. Hall in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Internetworking LANs and WANs'
B.S. Stuck in
'Localnet' 86'
Online Publishers, 1986

'Chapter 4 : Directories and Dictionaries' in
'Building an Open System'
Jacob Slonim, Abraham Schonbach, Michael A. Bauer, Lachlan J. MacRae,
Keith A. Thomas
Van Nostrand Reinhold, 1987

'Capacitated Facility Location'
A. Kershenbaum in
'Current Advances in Distributed Computing and Communications'
edited by Y. Yemini
Computer Science Press, 1987

'Distributed Computing System Architectures : Hardware'
M. Bozygit in
'Distributed Operating Systems, Theory and Practice'
edited by Y. Paper, J-P. Banatre and M. Bozygit
Series F : Computer and Systems Sciences, Vol 28, NATO ASI Series, 1987

'Part2/ Chapter 6 : Networks and Interconnection Structures' in
'An Introduction to Distributed and Parallel Processing'
J.A. Sharp
Computer Science Texts, Blackwell Scientific Publications, 1987

'The Distributed System POOL'
L. Gerlach, K.T. Malowaniec, H. Schdeig, R. Spurk in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'On the Adequate Support of Communication Interfaces in Distributed Systems'
J. Nehmer in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'A Distributed Process Execution and Communication Environment for High-Performance Application Systems'
W. Schröder in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

8.1.2 Session Requirements

'Chapter 3 : Acquisition - Environment'
G.F. Mullery in
'Distributed Systems , Methods and Tools for Specification, an Advanced Course'
M.W. Alford, J-P. Ansart, G. Hommel, L. Lamport, B. Liskov, G.P. Mullery, F. B.
Schneider
Lecture Notes in Computer Science, Springer-Verlag, 1985

'Distributed Transaction Processing and the Camelot System'
A. Z. Spector in
'Distributed Operating Systems, Theory and Practice'
edited by Y. Paper, J-P. Banatre and M. Bozygit
Series F : Computer and Systems Sciences, Vol 28, NATO ASI Series, 1987

'Process management in a Distributed Operating System'
S. J. Mullender in

'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

8.1.3 End User Requirements.

'Reliable servers in the JASMIN Distributed System'
P. Uppaluru, W.K. Wilkinson, H. Lee in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

8.1.4 Call Processing Requirements.

'REX : A Remote Execution Protocol for Object-Oriented Distributed Applications'
D. Otway, E. Oskiewicz in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Chapter 2 : Concepts and Tools of Communication',
'Chapter 5 : Call Processing Requirements' in
'Distributed Computing Systems :
Communication, Cooperation, Consistency'
CORNAFION (group name)
Elsevier Science Publishers, 1985

'Part 1, Section 2 :Second Example : Mutual Control of Logical Clocks' in
'Networks and Distributed Computations,
Concepts, Tools, and Algorithms'
Michel Raynal
Computer Systems Series, the MIT Press, 1988

8.2 Databases.

8.2.1 Database Structures.

'Chapter 6/ Section 6.1 : Distributed Database Systems/
Some Introductory Concepts' in
'An Introduction to Distributed and Parallel Computing'

J.M. Crichlow
Prentice Hall, 1988

'Chapter 1 : An Overview of Database Management' in
'Advanced Database Techniques'
D. Martin
MIT Press Series in Information Systems, 1986

'Chapter 5 : DBMS Techniques and Database Architectures' in
'Advanced Database Techniques'
D. Martin
MIT Press Series in Information Systems, 1986

'Data Sharing and Synchronization Facilities'
A. Endrizzi, A. Kontogiorgos in
'Distributed Data Sharing Systems :
Proceedings of the Third International Seminar on Distributed Data Sharing
Systems'
edited by F.A. Schreiber, W. Litwin
North-Holland

'Architecture of the Network Data Management System'
W. Staniszkis, W. Kaminski, M. Kowalewski, K. Krajewski, S. Mezyk, G. Turco in
'Distributed Data Sharing Systems :
Proceedings of the Third International Seminar on Distributed Data Sharing Systems'
edited by F.A. Schreiber, W. Litwin
North-Holland

8.2.2 Database Access

'Tight Database Cooperation in a Server-Workstation Environment'
U. Deppisch, V. Obermeit in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Chapter 5/ Section 5.1 : Servers in the Client-Server Network Model/ File Servers' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow

Prentice Hall, 1988

'Database Access in Open Systems'
S. Pappe, W. Effelsberg, W. Lamesdorf in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Chapter 7, Section 4 : Distributed Databases - Query Processing' in
'Building an Open System'
Jacob Slonim, Abraham Schonbach, Michael A. Bauer, Lachlan J.
MacRae,
Keith A. Thomas
Van Nostrand Reinhold, 1987

'Selection in a Database : a Complete Discussion' in
'Advanced Database Techniques'
D. Martin
MIT Press Series in Information Systems, 1986

'An Authorization Mechanism for a Generalized DMBS',
M. Adiba and F. Azrou in'Distributed Data Sharing Systems :
Proceedings of the Third International Seminar on Distributed Data
Sharing
Systems'
edited by F.A. Schreiber, W. Litwin
North-Holland

'Query Decomposition in PRECI*'
S.M. Deen, R.R. Amin, M.C. Taylor in
'Distributed Data Sharing Systems :
Proceedings of the Third International Seminar on Distributed Data
Sharing
Systems'
edited by F.A. Schreiber, W. Litwin
North-Holland

'Concurrency Control Issues in Distributed Heterogeneous Database
Management Systems'
V.D. Gligor, R. Popescu-Zeletin in
'Distributed Data Sharing Systems :
Proceedings of the Third International Seminar on Distributed Data
Sharing
Systems'
edited by F.A. Schreiber, W. Litwin

North-Holland

'Part1, Section 1 : First Example : a Data Transfer Protocol' in
'Networks and Distributed Computations,
Concepts, Tools, and Algorithms'
Michel Raynal
Computer Systems Series, the MIT Press, 1988

'Chapter 3 : the LOCUS Distributed Filesystem'
B.J. Walker and S.H. Kisser in
'The LOCUS Distributed System Architecture'
G.J. Popek and B.J. Walker
Series in Computer Systems, the MIT Press, 1985

'Specification and Implementation of a Standard for Remote Database Access'
S. Pappe, W. Lamersdorf, W. Effelsberg in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Which Kinds of OS Mechanisms Should be Provided for Database Management?'
P. Christmann, Th. Härder, K. Meyer-Wegener, A. Sikeler in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

8.2.3 Distributed Databases.

'Chapter 12 : Distributed Information Systems' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

'Chapter 2 : A Model for Distributed Databases' in
'A Paradigm for Concurrency Control Protocols for Distributed Databases'
A. El Abbadi
Ph.D. Thesis at the Department of Computer Scienec, Cornell University, 1987

'On Centralized versus Distributed Database Systems'
D.M. Dias, P.S. Yu, B.T. Bennett in

'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Chapter 6 / Sections 6.2, 6.3, 6.4, 6.6: Distributed Database Systems/ The Case for Distribution, The Distribition Problem and Pattern, Queries and Updates in DDBS, Some Example Systems' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

'Chapter 3 : Data Representation, Packing, and Protection' in
'Advanced Database Techniques'
D. Martin
MIT Press Series in Information Systems, 1986

'Chapter 7 : Relational DBMS ORACLE (Product Review of Version 4 : April 1985)' in
'Advanced Database Techniques'
D. Martin
MIT Press Series in Information Systems, 1986

'ADDS - Heterogeneous Distributed Database System'
Y.J. Breitbart, L.R. Tieman in
'Distributed Data Sharing Systems :
Proceedings of the Third International Seminar on Distributed Data Sharing
Systems'
edited by F.A. Schreiber, W. Litwin
North-Holland

'MDRSM : A Relational Multidatabases Management System'
K.K. Wong, P. Bazex in
'Distributed Data Sharing Systems :
Proceedings of the Third International Seminar on Distributed Data Sharing
Systems'
edited by F.A. Schreiber, W. Litwin
North-Holland

'The Apollo DOMAIN Distributed File System'
P.H. Levine in
'Distributed Operating Systems, Theory and Practice'

edited by Y. Paper, J-P. Banatre and M. Bozygit
Series F : Computer and Systems Sciences, Vol 28, NATO ASI Series, 1987

8.2.4 Distributed Database Issues

'Fundamental Issues in Distributed Database Systems'
M.J. Muller, G. Thomas
Technical Memorandum, Bell Communications Research, 1987

'A Technical Comparison of Distributed Heterogeneous Database Management Systems'
S. Bhalla, B.E. Prasad, A. Gupta, S.E. Madnick
M.I.T. Industrial Liaison Program Report, 1987

'Data Distribution in Manufacturing Systems'
S. Jablonski, T. Ruf, H. Wedekind, G. Zoertlein in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Chapter 6 / Section 6.5 : Distributed Database Systems/ Partition Failures' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

'Chapter 7, Sections 3, 6, 7 : Distributed Databases/ Conceptual Design Issues, Physical Design Issues, Database Administration' in
'Building an Open System'
Jacob Slonim, Abraham Schonbach, Michael A. Bauer, Lachlan J. MacRae,
Keith A. Thomas
Van Nostrand Reinhold, 1987

'Chapter 6 : Optimal Database Implementation Techniques' in
'Advanced Database Techniques'
D. Martin
MIT Press Series in Information Systems, 1986

'Index Selection in a Distributed Data Base',
E. Barucci, E. Grazzini, R. Pinzani in

'An Authorization Mechanism for a Generalized DMBS',
M. Adiba and F. Azrou in
'Distributed Data Sharing Systems :
Proceedings of the Third International Seminar on Distributed Data Sharing Systems'
edited by F.A. Schreiber, W. Litwin
North-Holland

'Concurrency control Issues in Distributed Heterogeneous Database Management Systems'
V.D. Gligor, R. Popescu-Zeletin in
'Distributed Data Sharing Systems :
Proceedings of the Third International Seminar on Distributed Data Sharing Systems'
edited by F.A. Schreiber, W. Litwin
North-Holland

'Communication Models for Distributed Computation'
D.K. Gifford in
'Distributed Operating Systems, Theory and Practice'
edited by Y. Paper, J-P. Banatre and M. Bozygit
Series F : Computer and Systems Sciences, Vol 28, NATO ASI Series, 1987

'Part2, Section 2 : Mutual Control of Logical Clocks'
'Networks and Distributed Computations,
Concepts, Tools, and Algorithms'
Michel Raynal
Computer Systems Series, the MIT Press, 1988

'Part1, Section 1 : First Example : a Data Transfer Protocol' in
'Networks and Distributed Computations,
Concepts, Tools, and Algorithms'
Michel Raynal
Computer Systems Series, the MIT Press, 1988

'Part 4, Chapter 12 : Data Flow Computing' in
'An Introduction to Distributed and Parallel Processing'
J.A. Sharp
Computer Science Texts, Blackwell Scientific Publications, 1987

'Which Kinds of OS Mechanisms Should be Provided for Database Management?'
P. Christmann, Th. Härder, K. Meyer-Wegener, A. Sikeler in
'Experiences with Distributed Systems'

Use class example of I/O mgt. | 1. Sockets 2. ~~Arch~~ Req. Arch. etc

CHAPTER 7

Distributed Operating Systems

The control of the flow of information, requests and the overall management of the session process is generally beyond the control of the typical functions of the communications systems that we had developed in Chapter 5. This management can be viewed as a major function of the operating system of a standard operating system of a stand alone computer and more importantly the function of a distributed operating system of a fully distributed environment.

Philosophy – OS is designed to meet reqmnts

There are two extreme approaches that can be taken to viewing the overall operational control and management of a large distributed system. The first approach states that the control can be fully distributed in the layer architectures of the separate elements of the network, and thus follow all of the ISO layered commands. Frequently this works well within a purely record based and generally homogeneous network of multimedia elements. The second approach states that a form of central control, although implemented in a distributed fashion is a means to the same end. This latter approach allows for the integration of multiple system elements, even though they may not be fully ISO compatible. Moreover, it allows for the introduction of new and innovative multimedia interfaces that the record based ISO level may not anticipate.

In this chapter, we focus on the development of a multimedia distributed operating system that ensures that the sessions are managed properly and that handles all of the other functions of an operating system. These functions include such elements as file and record management, process and session management, device and I/O management and an overall network management function.

As with the other chapters, our focus is to build on existing paradigms and then to expand the concepts to a fully distributed and multimedia environment. Moreover, we shall also develop the sizing and performance analyses that are necessary for the development of an overall system architecture.

7.1 Operating Systems

The operating system is that software that allows all of the different processors to be be utilized to their best degree. In particular, the operating system will allow for control of the memory of the system and the files that are to be managed by the processors. It also allow for the overall management of the processes that are running on the system. Finally, the operating system allows for the control of the input and output devices on the system. In a distributed environment, the operating system can be extended to include all of the processes in the system and in turn be the manage of the overall files in the system. As we discussed in the development of the distributed database environment, the distributed operating environment, builds upon the stand alone system.

In this section, we first develop an understanding of the stand alone operating system and then we develop the overall concepts of the distributed operating system. In particular we develop a system called, MEDOS, standing for the Media Distributed Operating System. This system has been developed for use in a fully distributed multimedia environment.

7.1.1 OS Structures

An operating system is simply a collection of software that, at one side controls the hardware and other external assets of the computing environment, and at the other side, provides a ready means for the user of the resources to incorporate them into the overall application. Thus we can view the operating system from two directions, first that towards the resources managed, and second towards the end users need to manage the overall set of resources. There are generally five major unctions of an operating system as we understand it today. These are:

- o Process Management
- o Memory Management
- o I/O Management
- o Device management
- o File Management

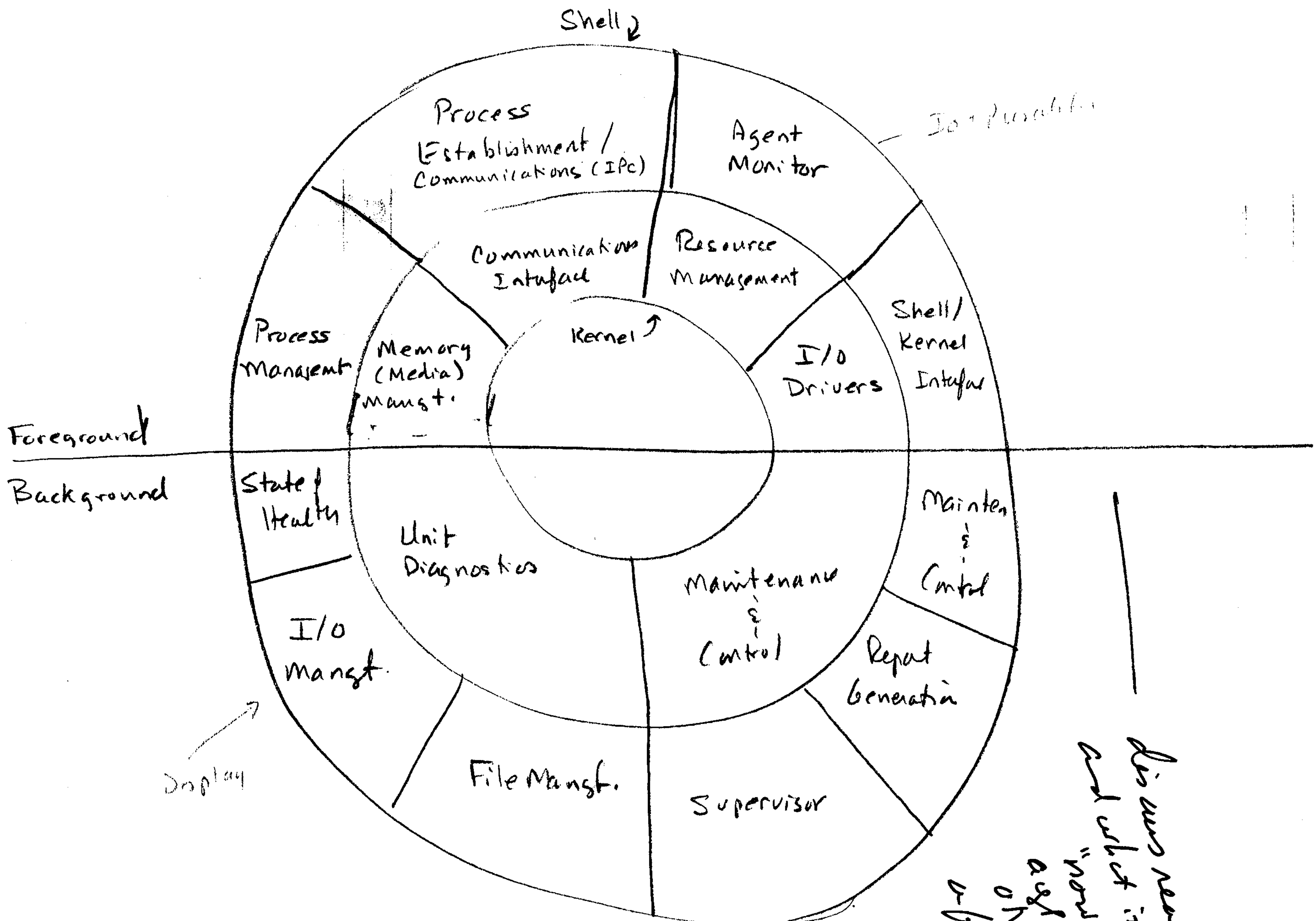
These five major functions relate to typically a single stand alone environment that focuses on the need to manage the resources for a single machine. As we have discussed in the development of multimedia communications, the primary element is that of the session. In contrast, in the environment of the single machine, even with multiple users and multiple applications, the primary focus is the process (an program in execution). There are significant differences in the ability to meet the needs of a process as compared to a session. In this section, we shall review the structure of current operating systems, and expand the concept to the area of a fully distributed operating systems focused on multimedia applications.

Operating systems have evolved from the simplest ones that were basically the operator themselves, through batch systems, and upwards to today's systems that have the capability to perform multi-processing and multi-tasking. The development of such common operating system environments such as UNIX and MS-DOS are responses to the end users needs to manage the computing resources of the specific location. Expanding the resource management to multiple locations has been achieved through communications channels that we have discussed in the last chapter.

To provide an oversight, we can view operating systems to fall in four general categories. These are a local operating system (LOS), a networked operating system (LOS), a distributed operating system (DOS) and a multi-media distributed operating system (MEDOS). Our objective in this section is to develop the structure of a MEDOS and to do so by showing how it is a natural evolution of the LOS environment. Before continuing, let us first define in some detail each of these separate operating system environment, with emphasis on the requirements of the end user that they satisfy.

Before comparing the different architectures for the operating system environments, let us first consider a canonical environment of an operating system. Figure 7.x shows the canonical structure for the operating system. It consists of six levels. These levels are:

Figure 7.x Canonical Operating System Structure

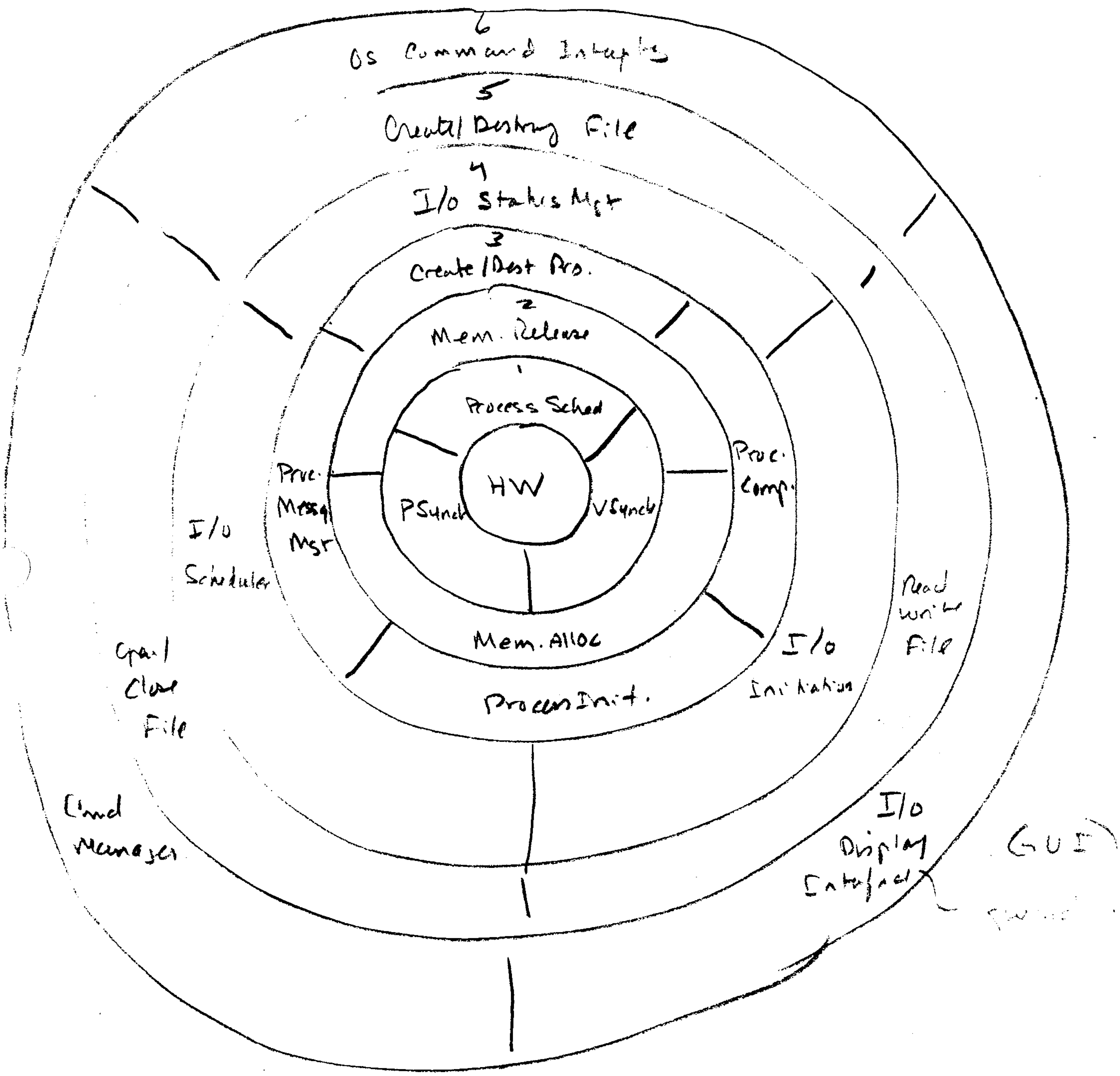


discuss real time and what it is — "now" and time otherwise a fault is sent.

GUI in PC world

Figure 8.17 Operating System Structure: Canonical Example

(Madnick & Donovan Op Sys p 19)



o Hardware: This is the innermost level and represents the totality of all hardware that the operating system could manage as resources to the end user and the application. Typical of such resources could be the processor or processors involved in the computation, the memory units, printers, displays and entry devices for the end user.

o Generics: This typically is microcode that is written specifically for each device and may also typically be provided by device vendors for the local access to interface and manage the individual devices. For the most part, the device drivers and fault monitors are part of the generic elements.

o Kernel: This is the first part of the operating system. The kernel element, for our purposes, is the layer of operating systems functions that directly interface with the devices and hardware. In this canonical structure, all kernel functions of the operating system are hardware directed, as compared to the shell functions that are applications support directed. Thus memory management is a kernel layer function whereas file management is a shell layer function.

o Shell:The shell is the second layer of the operating system and is that layer that looks out towards the user. It provides for the user interface and also allows for direct access by the higher layers to all the service provided by the operating system. Access to the shell layer is through a set of commands that we call primitives. Primitives have a defined syntax and can be used by the end user to not only access all the operating system facilities but to manage them in sokme detail.

o Services: The services layer is provides for special support functionality to the end user application. Typical service layer functionality is that of a windowing environment that evokes the functionality of the operating system. X Windows is a typical service layer functionality that supports the capabilities in UNIX base system. Other services could include the ability to perform sessioning, the ability to mail in the network, the ability to perform communications service between various nodes ijn the network and other such services. The service layer can also be thought of as the OSI layer interface at the applications layer.

Typical services that can be supplied by the system are as follows (see Figure 7.x):

o Communications: This service allows for the connecting of a local environment to any other local environment. This service usually utilized the ISO based layered approach to establish the communications link.

o File: This service attends to the need of the end user in accessing and maintaining their overall files. In a distributed environment the file service extends to sharing in a common and seamless fashion all of the important file elements.

o Directory: This service allows access to the location and names of all files on the system. If we consider a file in the extended sense of UNIX then we can see that the directory provides the overall access mechanism for any readable and/or operable entity or agent on the system.

o Mail: This service allows for the sending of data elements from one user to another on the system.

o Session: This is and advanced service that allows for the establishment and maintenance of a session based service. We shall be discussing this at length.

o Gateway: This service allows for the interconnection of multiple users to other networks. It extends the communications service by providing a level of protocol translation and conversion.

o Presentation: This service consists of means and methods to present information on the end users console. An example, as we have discussed is the X Windows service that is used for bit mapped presentation.

o Print: This service allows for the output in a hardcopy format of any of the data or processed files in the system.

These services are only examples of the many that can be generated at the services layer. The services are accessed via Service primitives that allow for specification of the details of the service elements. In turn the services are generated by using a collection of the Shell Primitives.

o Applications: This is the layer that include the set of programs that make up the end user application. The applications may be one or many, and they may be associated with one or many end users.

We can now use this canonical model and explain each of the four types of operating systems in terms of the canonical form. The reader should note that we have extended the models for the operating system environment beyond what may generally appear in the literature. The classic text by Madnick and Donovan provide an excellent review of operating systems in the mid-1970's. The more recent texts by Tannenbaum and by Deitel provide excellent updates to the environment.

Consider now the different types of systems:

o Local Operating System (LOS)

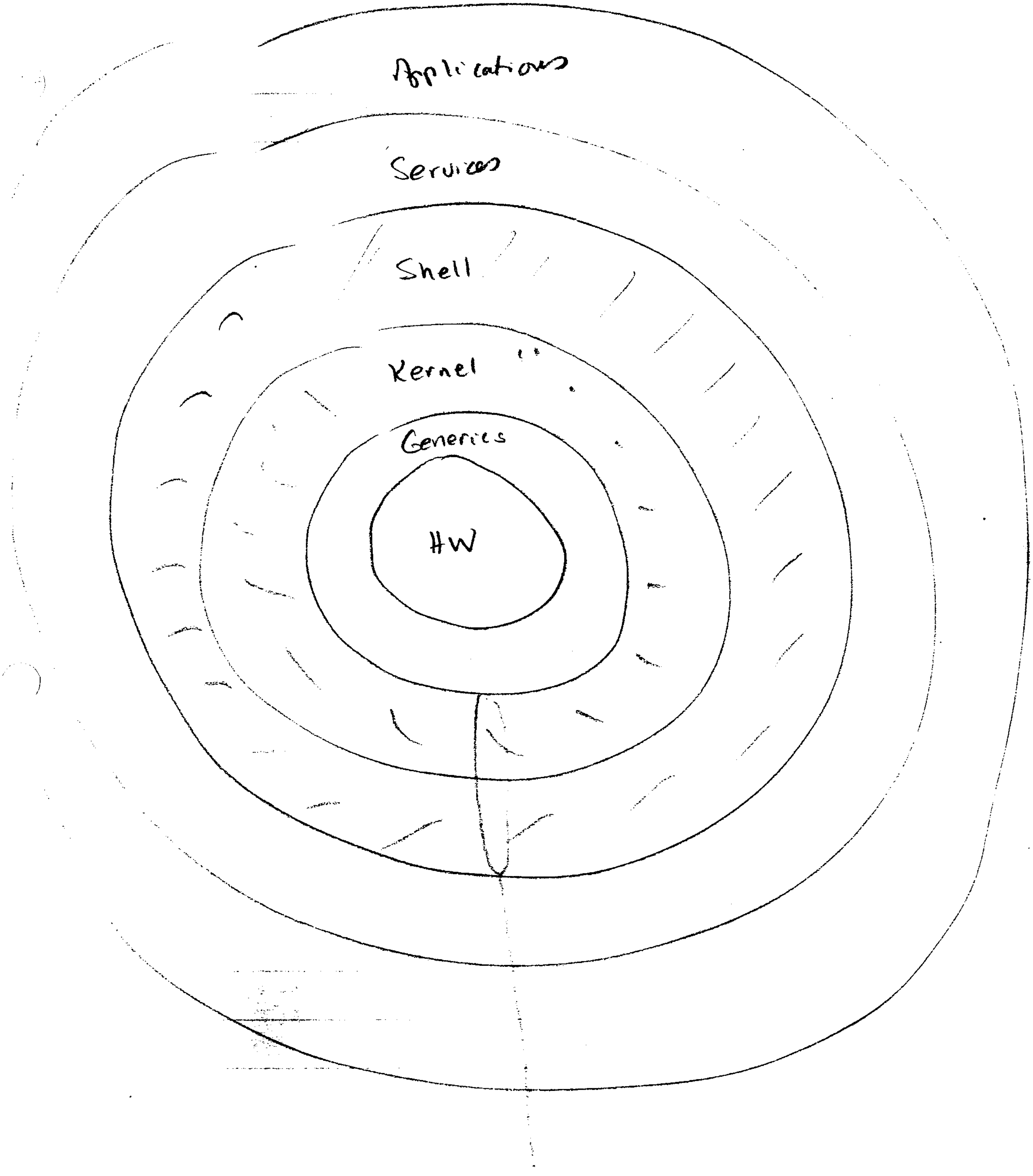
A LOS is the result of the evolution of the end users need to readily access many of the system resources and the systems designers needs to allow for maximum flexibility in explaining both the applications run on the machine and the number of users utilizing the machine resources. From the end users perspective, there is a need to access files that are stored in the system and to manipulate those files for the purpose of a specific application. We allow the access to the LOS through a user interface that is a command interpreter. The commands that allow the user to manipulate files are similar to commands that permits all of the functions in the LOS. We shall call these commands the operating system primitives, or primitives for simplicity.

The LOS focuses on the local environment and its resources. It allows fore the input and output of the end users commands and data and also manages all other system resource I/O. In a similar fashion to file management, it manages the system memory, and if necessary can interact with the management of processes to permit certain memory to be brought to faster processing locations for better optimization of applications run time. Typical of this latter approach is a software controlled cache function.

To the user of an MS-DOS system, the commands such as dir (for directory), path, and file, are typical of the file management commands readily available.

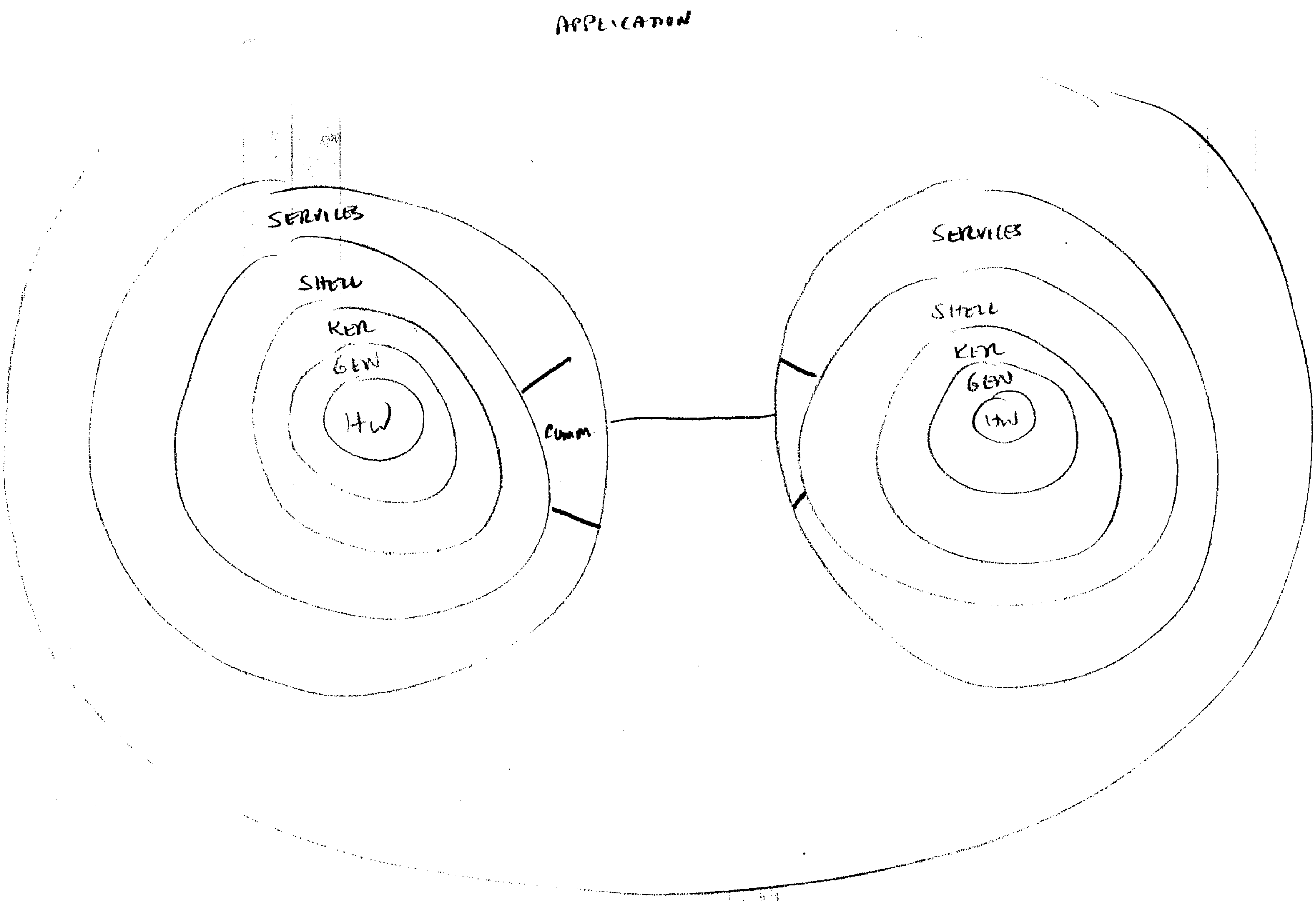
Figure 7.x depicts the structure of an LOS operating an application that may share one or more machines. The application is bound between the two machines by the services layer communication service. This means that the machines are totally separate and are self contained. Only in an abstract sense are the applications shared. In reality they are fully resident in each machine.

Figure 7.x LOS Canonical Structure



OPERATING SYSTEM

LOS



APPLICATION
SERVICES
SHELL
KER
GEN
HW
COMM.
SERVICES
SHELL
KER
GEN
HW

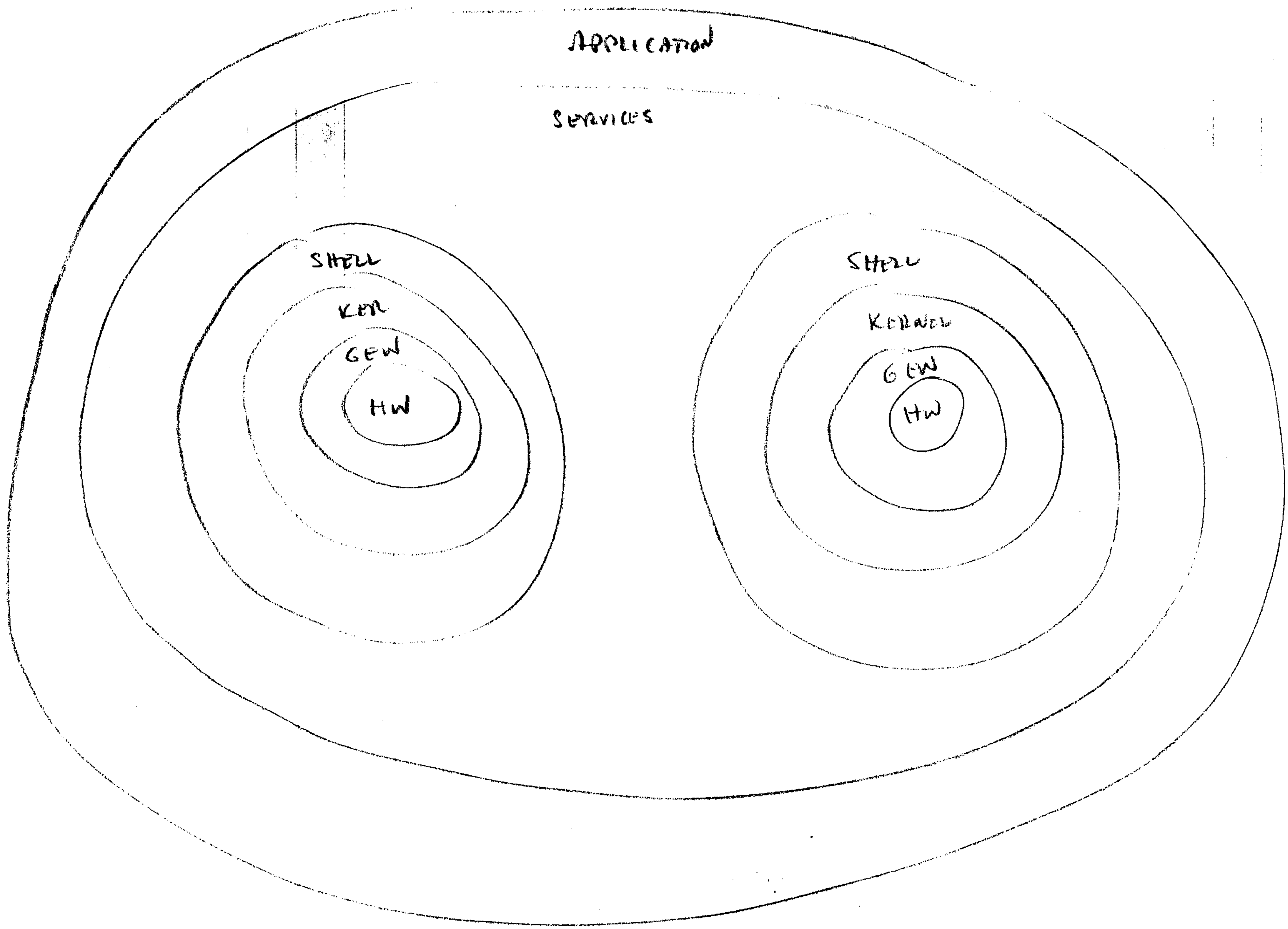
o Network Operating System (NOS)

A network operating system is the next step up in complexity from the local OS. In the local OS environment, all of the resources are local and are owned and operated by the single OS. In the network OS case, there are multiple locations, but all of the resources are locally owned and controlled. Thus if a user desires to use the resources at another location, that user must log onto those resources through the network.

We can see in Figure 7.x that in the NOS case, the OS elements are still discrete even though the users may be sharing the applications and now Rtes services layer. The NOS structure allows access to all elements through a shared network communications protocol and also a set of other shares service elements.

Figure 7.x NOS Canonical Structure

NOS

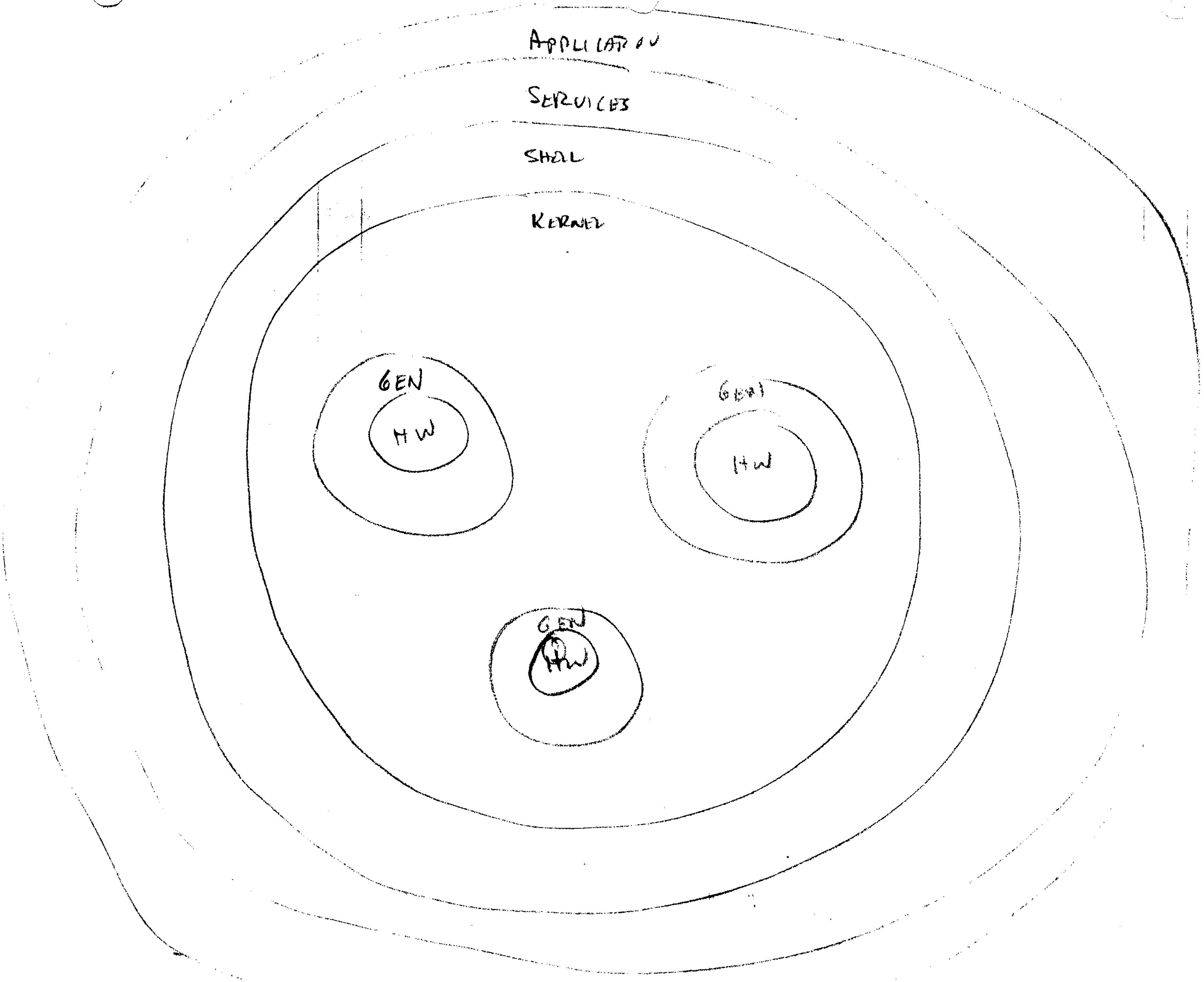


o Distributed Operating System (DOS)

In a distributed operating system,only the hardware and its associated generics are still associated on a logical basis with the individual sites. In this case the operating system functions of both the shell and kernel are shared amounts all the users. In a fully implemented DOS environment, no user will be able to identify the specific resource being utilized at any time. The sharing of all of the resources and their access is through the common distributed shall/kernel combination. Figure 7.x shows how such a DOS environment can be envisioned. One could conceive of the day when a fully distributed system is developed with sharing of even the generics and hardware. Such is not the issue in the present discussion.

Figure 7.x DOS Canonical Structure

<u>DOS</u>



o Media Distributed Operating System (MEDOS):

We have discussed the concept of a distribute operating system and the focus has been on the system that has standard data elements and storage. In a multimedia environment, the hardware the lies in between the system is more complex and hardware elements are multimedia in character. Thus the system must combine all of the elements that we have discussed previously.

7.1.2 OS Elements

Operating Systems have been developed to provide serves to a wide variety of machines and in a large set of environments. Typically there are four major functions of an operating system (see Madnick and Donovan):

- o Memory Management
- o File management
- o Process management
- o I/O management

We have also included an fifth function (see Tannenbaum):

- o Device management

Very simply, these five functions perform the following tasks.

o Memory Management: This function manages the reel and non-real time memory of the system. It allocates space in memory and moves date to and from different parts of the systems memory. As we look towards multimedia applications, memory management becomes even more complex than that for standard digital data. The nature of the storage media changes and the use of real and virtual storage becomes more complex.

o File management: A file is defined as a collection of dat that has a specified name (see Deitel). In contrast to memory management, which manages the physical resources of the storage devices, file management manages the allocation of data to memory that is used in the context of specific applications. As with memory management, file management for multimedia applications is more complete because of the need to create files of often unbounded and disparate media.

o Process management: A process is a program in execution. The management of processes entails the allocation of system resources to ensure the effective execution of all processes. The implementation of interprocess control (IPC) for the coordination and swapping of processes in execution is also a key ingredient of this function.

o I/O Management: The management of the various input and output devices is a key element of this function. This includes the management of printers, display devices and other elements.

o Device management: Device management typically refers to internal elements oft the computation process. It may often be bundled into the management of I/O elements.

These five areas are the classical elements of the operating system structure. However, in Figure 7.17, we have displayed the canonical structure of an operating system. This canonical structure shows two major divisions. The first division is that between the shell and the kernel. As we have stated before, the shell is that part of the operating system that relates outwardly to the end user, and specifically to the services layer. The kernel looks inward to the devices. The second division is in terms of foreground and background process. The foreground represents real time or near real time processes that operate in the OS. The background processes are no real-time processes.

The division of an operating system into these two divisions is an important one from an operational perspective. The shell/kernel division allows the designer to focus on the directional emphasis on interfacing and functionality. The foreground/background partition focuses on the real time nature of the associated processes.

In the canonical structure, we have shown sixteen separate processes that are used to define the canonical operating system. Theses represent those that are normally a part of the standard five elements plus a set that we have added for the purpose of expanded beyond the local OS to a fully distributed environment. We shall now discuss the contents of each of the processes.

- o Shell
    - o Foreground

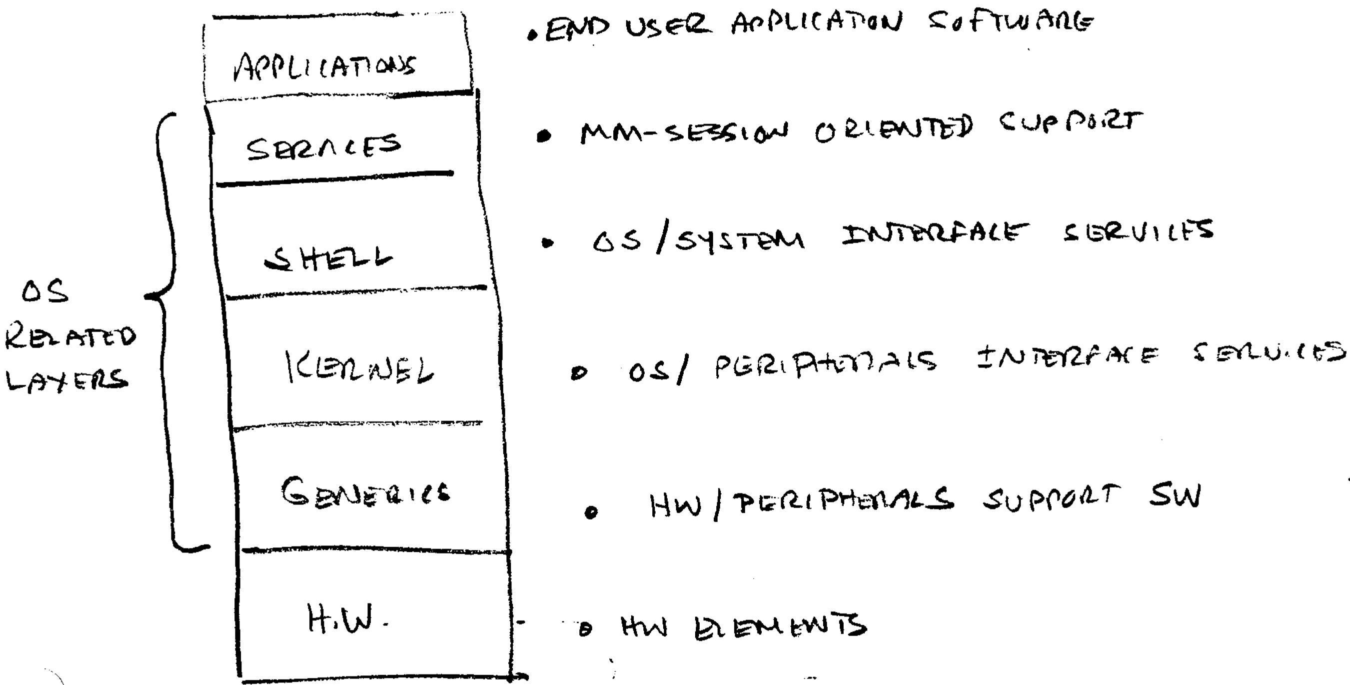
        - Process management
        - Process establishment
        - Agent Monitor
        - Shell/Kernel Interface
    - Background
        - State of Health
        - I/O management
        - File Management
        - Supervisor
        - Report Generation
        - Maintenance and Control
- Kernel
    - Foreground
        - Media(Memory) Management
        - Communications Interface
        - Resource Management
        - I/O Drivers
    - Background

o Unit Diagnostics

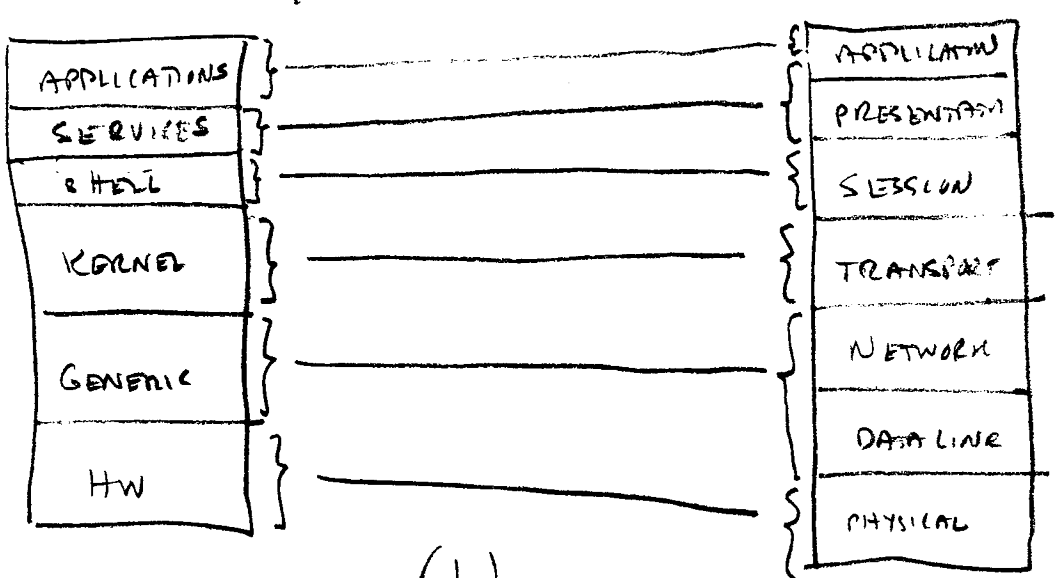
o Maintenance and Control

Figure 7.17 Operating System Structure: Canonical Example

Figure 8.20 OS Elements



(a)
Element Function



(b)
ISO Relationships

Figure 7.18 Operating System Types

discuss sockets versus session layer functions

Discuss TLI functions

We can now use the discussion of the canonical operating system and show how existing operating systems map into the canonical model. In Figure 7.19, we have presented three of the more common operating systems in use, VMS, UNIX and MS-DOS.

Figure 7.19 Example OS: (a) VMS, (b) UNIX, (c) DOS

7.2 OS Details

Now that we have defined the overall canonical model for an operating system, and we have provided a comparison to specific existing operating systems, we can now demonstrate how each of the key processes is developed with module specification diagrams. Figure 7.20 reiterates the overall canonical architecture

Figure 7.20 OS Elements

Figure 7.21 OS Modules

Figure 7.22 details the modules for the Agent Monitor process and the Media Management process. These module specification diagrams portray the dataflow and the module objects that need for the implementation of the process.

Figure 7.22 OS Module State Diagrams

We can now detail the processes for the canonical for by comparing them to existing operating systems. Figure 7.23 shows the module descriptions for similar modules for the operating systems that we have discussed.

Figure 7.23 Detailed OS Structures: (a) VMS, (b) UNIX, (c) DOS

7.3 Distributed OS

The operating system structure that we developed in the last section was a generic structure that can apply to all possible types of operating systems. However, as we have seen in the examples of existing operating systems that are local in nature, many of the elements of the canonical model are not needed in the system. What distinguishes a distributed operating system from a local system is the need to ensure that all of the physical resources are treated as one set of local resources and not just as a collection of separate local sets. Thus, for a distributed operating system, there is a need for significantly increased communications management amounts the physical resources.

Let us begin by considering a simple example in the area of memory management. In a local operating system, we can partition the memory and manage it from a single local point. Let us assume that the system has both RAM memory and Disk memory and that we page memory from disk into RAM for processing purposes. Let us assume that main memory (eg RAM) has capacity of C_M and that their are N processes that are being run simultaneously.

Let P_i represent the ith process. Let Pi take memory space MPi and let the dat requested for Pi take space MMi. Thus using an appropriate swapping algorithm, we can use a performance measure of the system and the algorithm that measure the efficiency of the usage of main memory and the efficiency of the time efficiency of use. We call this the machines time-bandwidth product. It is defined as:

$$(7.x)\ BT = (MEM_{eff} * MIPS) \quad * \quad (Time_{eff} * T_{cycle})$$

where:

(7.x) MEMeff = the percent of main memory fillable on average by the swapping algorithm.

(7.x) MIPS = the processing rate of the machine in instructions per second.

(7.x) Time eff = the percent of a cycle time that is used for processing as compared to all other factors such as data transfer etc.

(7.x) T cycle = the average instruction cycle time

Now the T eff can be calculated as follows. If T load is the total time to load a main memory cell, and T proc is the total processing time for the command of that process in that cell then we define T eff as:

$$(7.x)\ T_{eff} = \frac{T_{proc}}{T_{proc}+T_{load}}$$

Now generally we have :

$$(7.x)\quad T\ load << T\ proc$$

because the memory access time is significantly short. The access time of the memory is generally short and is dependent mainly upon the I/O characterisics of the memory device. In a distributed environment however, the access time is increased due to the communications associated with the memory call plus the nomad memory I/O access time. Specifically we have:

(7.x) T load = T I/O + T comm

Now when we design the paging algorithm, or consider the need for virtual memory, even in one machine, we must include the communications effects into the analysis. Frequently:

(7.x) T comm >> T I/O

Let us calculate some of the numbers in these equations to give the student a feeling for the dimensions of the problem. The speed of light is about one foot per nanosecond. If we have a distributed system with separation of 500 miles on average, then this is $2.5\ 10^{6}$ feet and equals a delay of 2.5 msec. If the I/O to the memory unit works at microsecond speed, then we have the inequality indicated with three orders of magnitude difference. In addition if we have a 2 MIP machine, and we page in 100 instructions per second, the processing time is 50 microseconds. Clearly T proc is greater than T I/O but much less than T comm. It will be this factor that we shall see dominating the DOS performance.

Table 7.x depicts the mapping of the detailed operating systems functions as described against the canonical model and shows how they relate to the higher level functions commonly used in operating system descriptions. We shall now use the more common

structure to discuss the key features that must be considered in a distributed operating system as compared to a local operating system. Each of these issue relate to the problems that we have discussed in the above example. That is, it is now possible to use other memory devices and other processors that are connected more loosely and with greater delays than generally anticipated.

Table 7.x DOS Functional Inter-Relationships

Function	Process Mgt	Comm Mgt	Device Mgt	I/O Mgt	Memory Mgt	File Mgt	Ev Mg
SHELL							
FOREGROUND							
Process Mgt							
Process Estb							
Agent Montr							
SK Interfc							
BACKGROUND							
State of Hlth							
I/O Mgt							
File Mgt							
Supervisor							
Rpt Gen							
M&C							

KERNEL

FOREGROUND

Media Mgt

Comm Interfc

Resource Mgt

I/O Drivers

BACKGROUND

Unit Diagnosts

M&C

Let us now consider all of the major elements of the DOS and detail their key performance factors and compare these against a standard LOS.

o PROCESS MANAGEMENT

The process management function is directed towards the overall management of the processes that are operational in the system. It looks outward towards the processes that are part of the overall applications program and ensures that they are scheduled and interlaced in an effective manner. Process management takes the form of managing the multiple users in the system and matches the users and their needs with the overall set of system resources. A process is taken in and out of the main memory and assigned system resources depending upon the needs of the process, the priority of the end user and the availability of the resources.

Local System Performance

In a local operating system environment, the issues of process management relate to the following items.

- o Process Establishment:
- o Interprocess Communications:
- o Process Monitoring:
- o Process Messaging:

o Process Scheduling:

o Process Termination:

Distributed Systems Design Factors

In a distributed environment, we include all of the above items plus several that are in addition to these. Moreover, we need to expand the functionality of the separate elements of the process management function to ensure that processes are properly managed across several resources. Process management in a distributed environment may take on several different aspects. In the local environment, we have a set of processes that can be existing on a single processor. They are managed by the operating system of the local processor. In a distributed system, we can envision several scenarios:

o Single process per Processor/Multiple Processors: In this case there is a single process that is operational at a single processor and it must communicate with other processes at other processors.

o Multiple Processes per Processor/Multiple processors: In this case we have multiple processes that are operational at each processor but that they all must communicate between themselves.

o Single process per Multiple processor/MP: In this case we have a single process that is shared amounts several processors. These processors are separated by many local distances and thus have the dynamics associated with separate locations.

o Multiple processes per Multiple Processors?MP: In this case we have multiple processes, all communicating amongst themselves, distributed in multiple processors.

Process management in a distributed environment is increased in complexity by the sharing of resources that are at disparate locations but the advantage is the ability to share resources and to dynamically reallocate resources amongst many users.

o DEVICE MANAGEMENT

Device management is also known as processor management. It is the physical complement of the process management that we have just discussed in the above are a process management. The processor management function basically relates to the assignment of processes to specific systems resources or processors. The process management function relates to the intra-process management functions, namely how does on manage the process itself and in relation to their processes. The processor management function relates to how wen manage the processes in relation to their outside environment. The processor management function relates to the overall need of the system to perform the following types of task:

o Job scheduling of the processes to the processors.

o Traffic control of tasks between and amongst the processors.

o Synchronization of the processors amongst themselves.

Device management for local systems and distributed systems is very similar. The allocation of resources can be managed on a centralized or a distributed fashion. In either of these cases the specific algorithms may be static or dynamic, and in turn may be time driven or event drive. Thus the type of processor manager may be characterized by the tuple:

{Control, Timing, Response}

where the control is central or distributed, the timing is static or dynamic, and the response is event or time driven. Thus there are the following types of processor management systems:

{Static, Dynamic, Event} etc.

This yields eight possible configurations of process management. We develop several algorithms for the management under these domains in both the examples in this section and the problems.

In a distributed environment, the same set of variables apply, and moreover the same set of algorithmic devices or artifacts are applicable.

o COMMUNICATIONS MANAGEMENT

o I/O MANAGEMENT

The I/O management entails both the I/O to devices and peripherals as well as I/O to the end user. We shall focus on both of this elements. The physical device I/O is controlled by a set of device drivers that talk directly to the device and are controlled by the management portion of the I/O element. The end user I/O are interfaced by what we term primitives, which are command sets that use a defined syntax and protocol and allow the user to interact with the operating system directly.

o MEMORY MANAGEMENT

Memory management provides the management of the physical memory of the system. There are typically many forms of memory that are used in a commuting system. These range from the local RAM memory storage, to local disk and local tape memory. Against these physical storage media is the allocation of the memory that the process recognizes as important and the memory as allocated and managed by the operating system. The process frequently has need to access many elements of the physical memory and it then needs to changes and redirect that memory as it is processed. The operating system must assist the process in its process management and memory management functions in allocating this memory to be effectively used by the system.

Typically we have a system with a fixed amount of memory that is immediately accessible by the process in operation. Let us defined each process by Pi and let the total memory be given by MT.

Let the set P{I,k} represent the process set at time k, and let this equal:

$$P\{I,k\} = \{\ Pi : i \quad I \text{ and } kT< t < (k+1)T\ \}$$

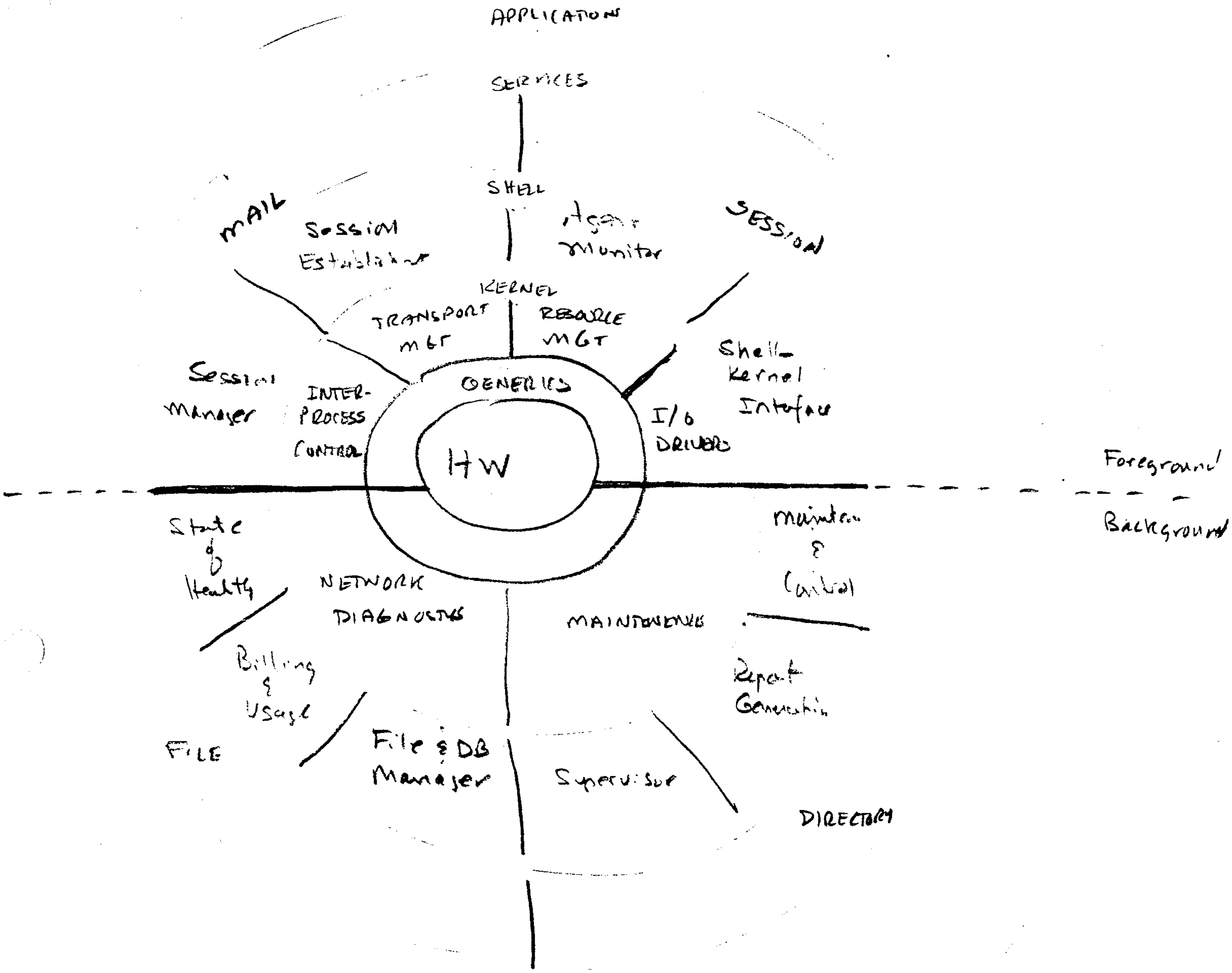
o FILE MANAGEMENT

o EVENT MANAGEMENT

7.4 Distributed OS Example

Figure 7.24 MEDOS Architecture

Figure 8.24 MEDOS Architecture



(a) Overall

Figure 7.25 MEDOS Modules

8.24b

Shell Architecture

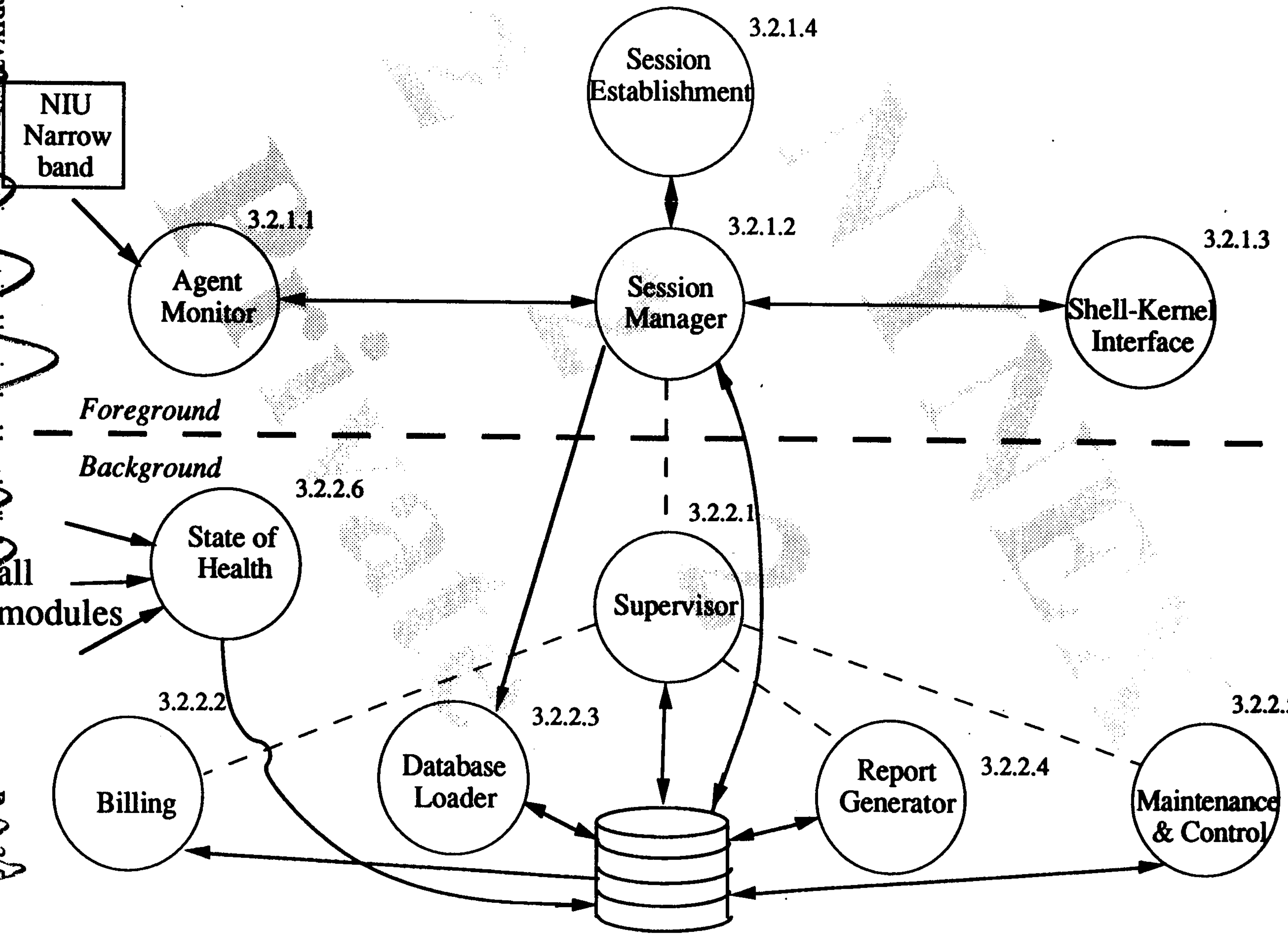


Figure 3.2

8.24(b)

Kernel Architecture

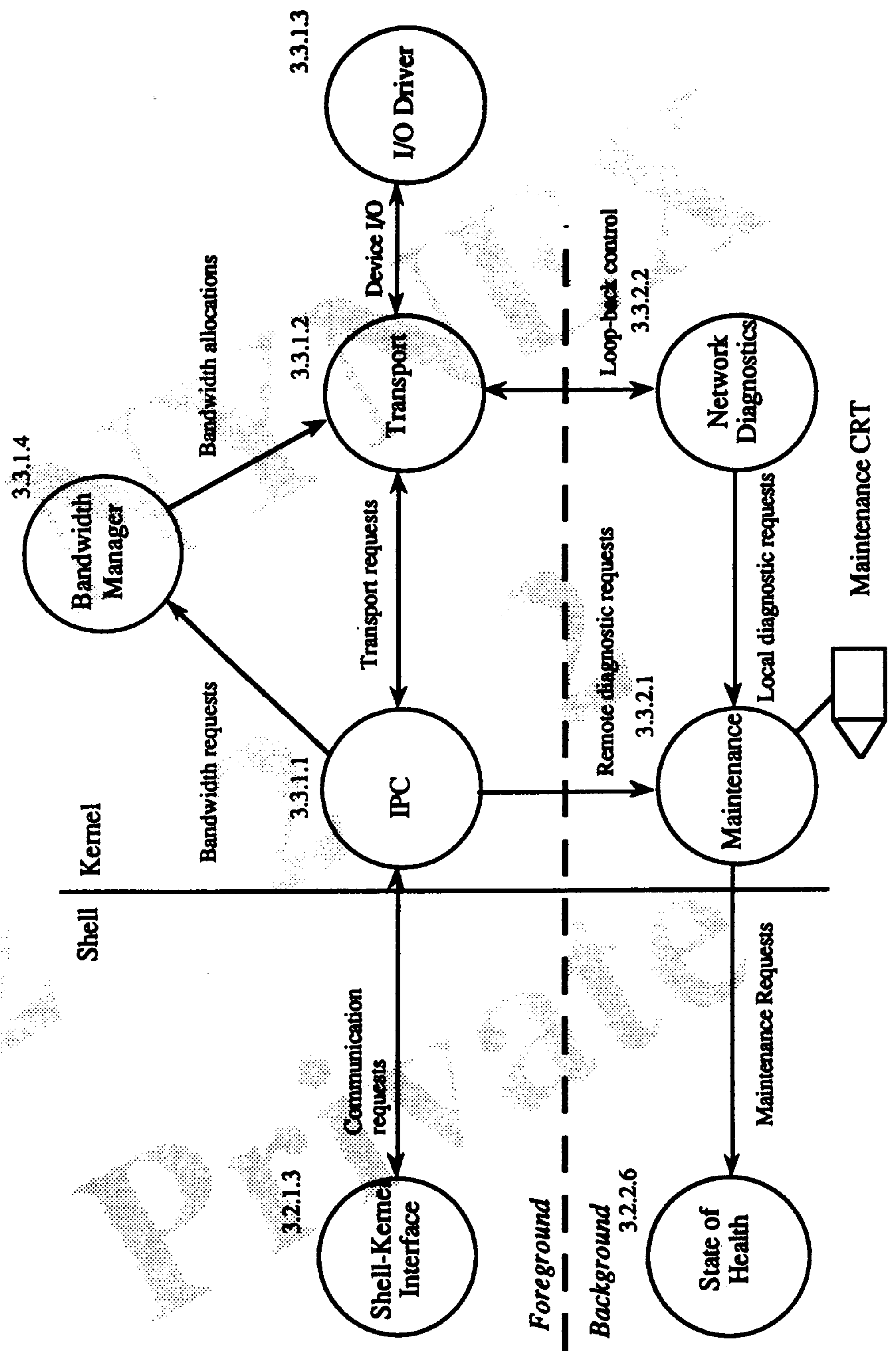


Figure 3.13

Figure 7.25 MEDOS State Diagrams

Figure 8.25 MEDOS Modules Example (Agent Monitor)

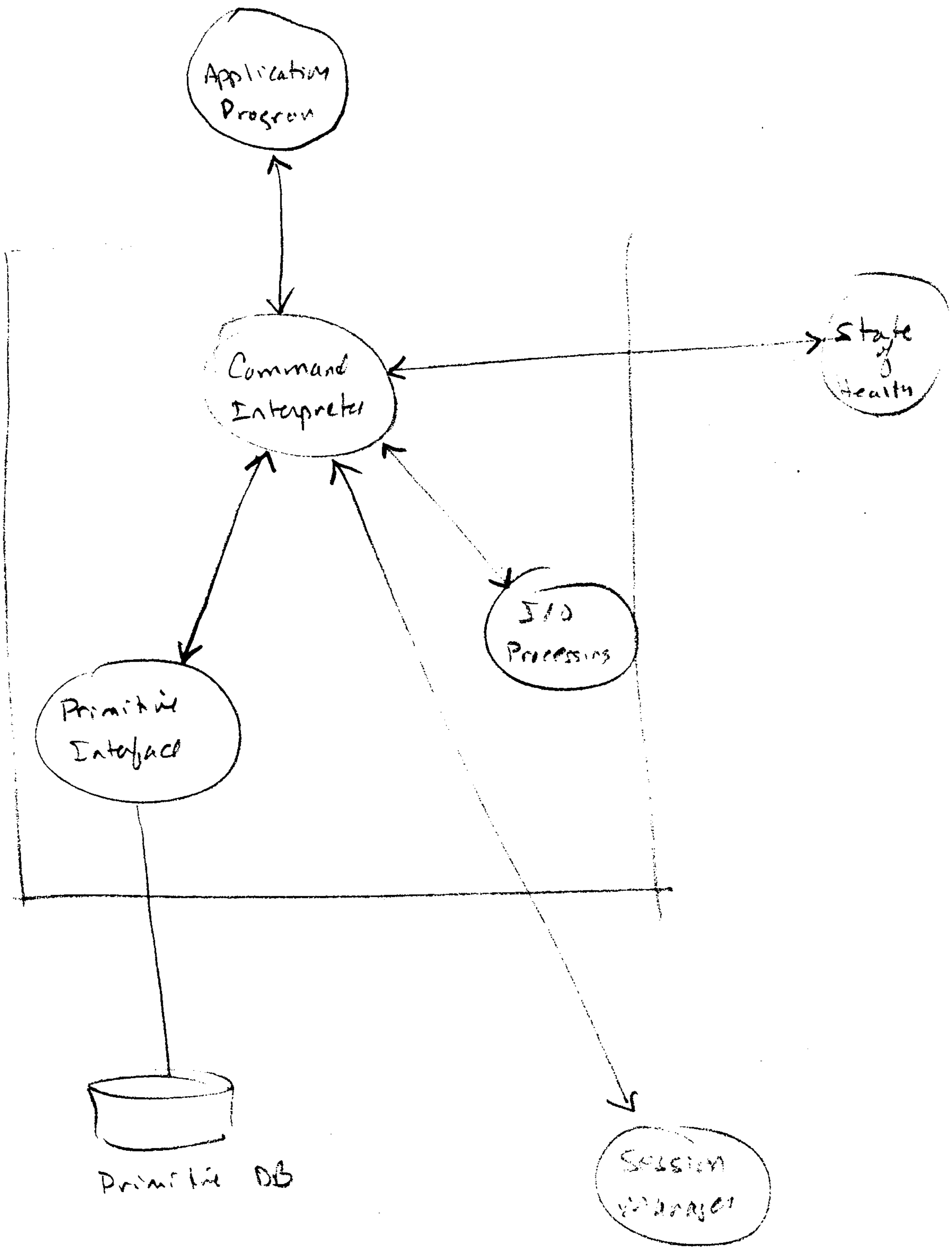
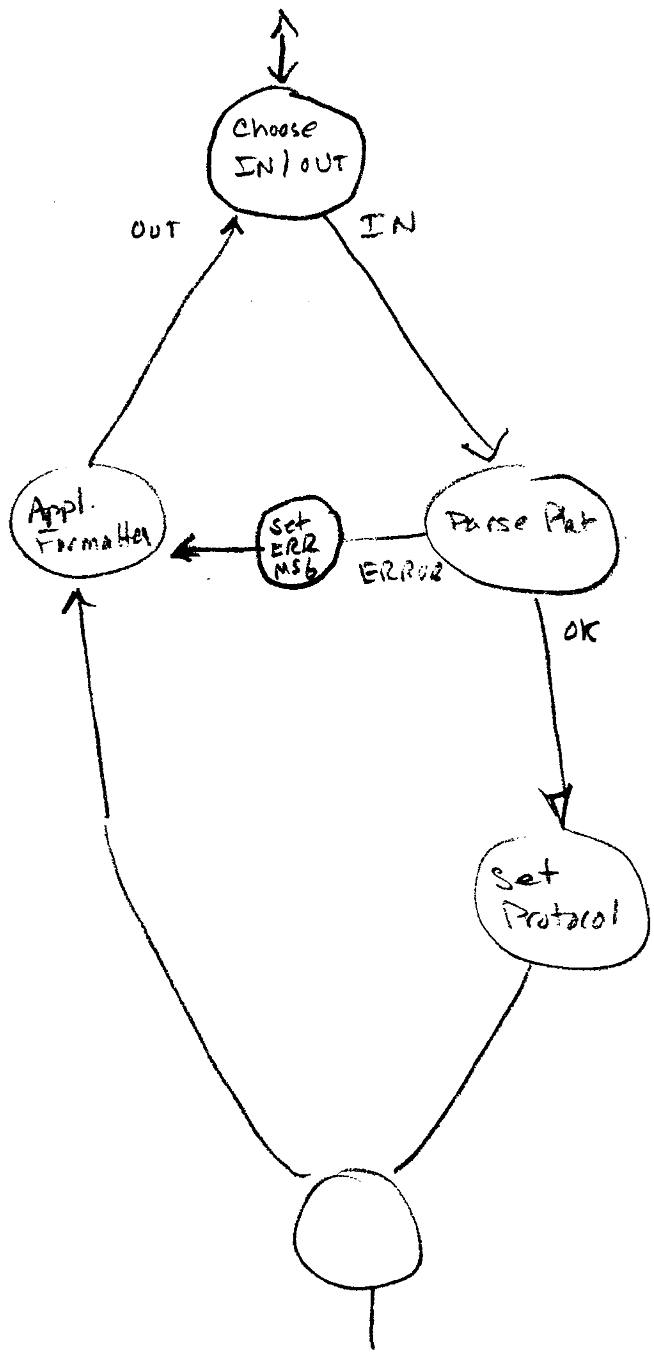


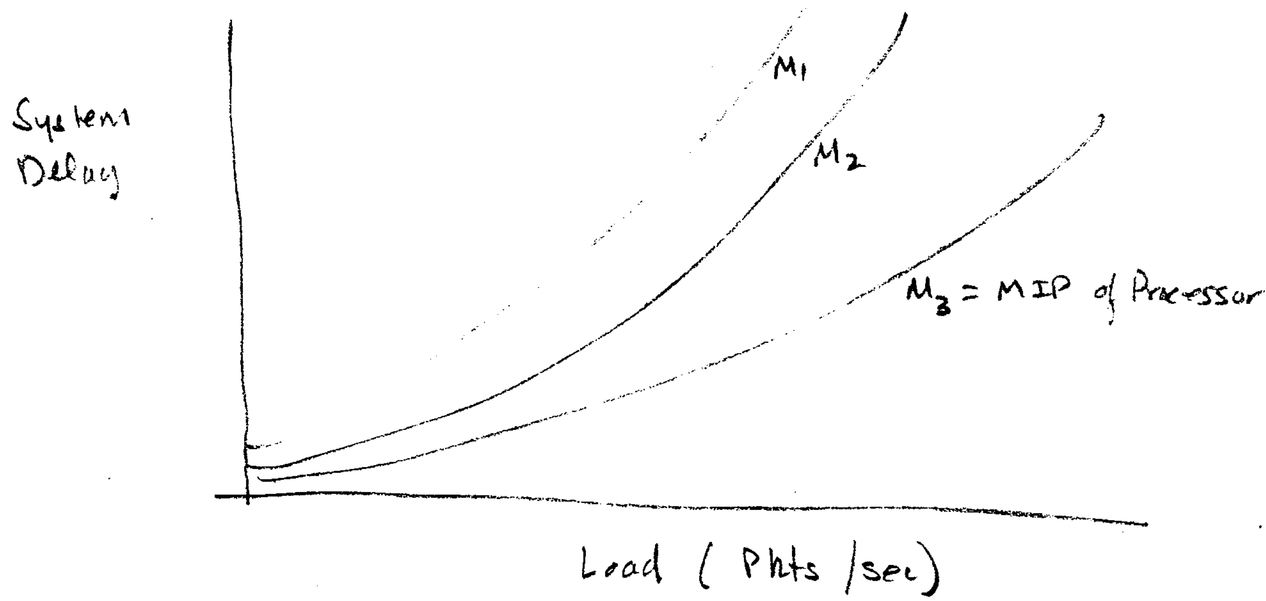
Figure 8.25 MEDOS State Diagrams / Agent Module / Command Interp

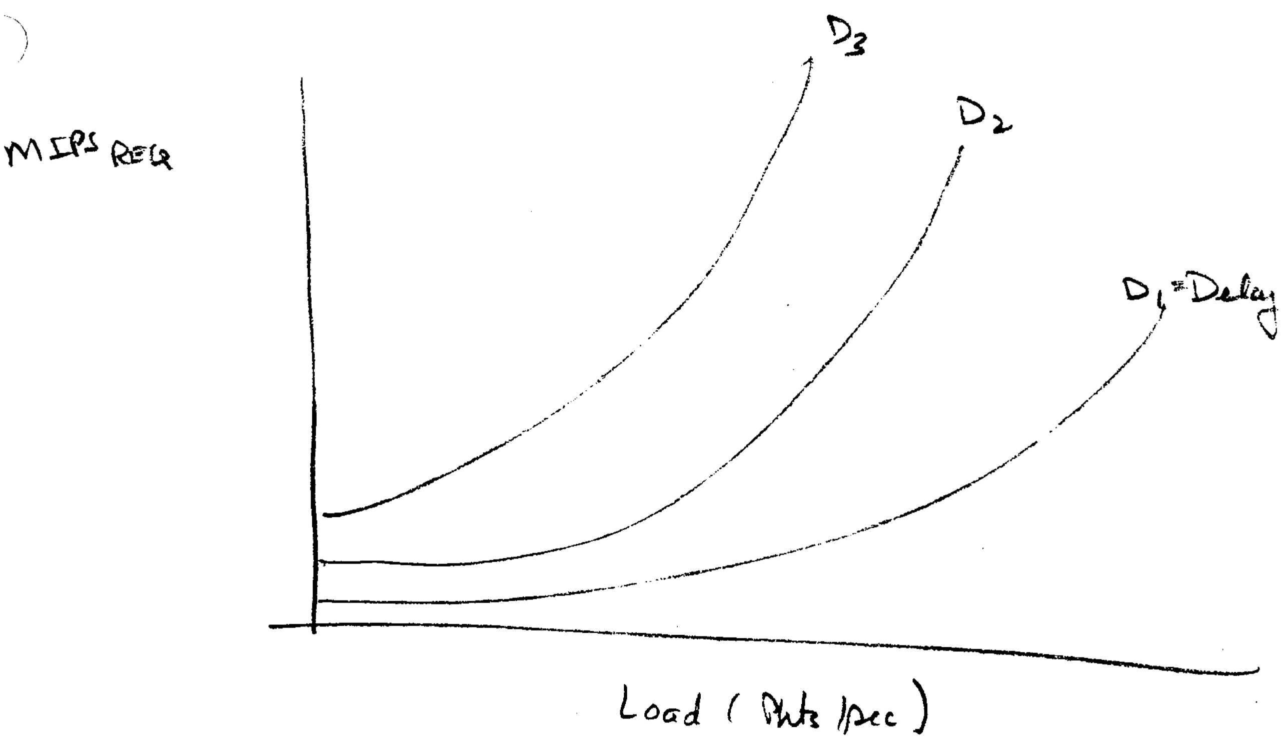


7.3.6 Distributed OS Performance Factors

Figure 7.26 MEDOS Performance Figures

Figure 8.26 MEDOS Performance Figures





7.5 Distributed Processors

In any distributed environment there are not only issues of the distributed data base and the distributed operating system, but there is also the issue of the distributed processors and their interconnection. This section addresses some of the issues of distributed processors.

7.5.1 Processor Types

7.5.2 Processor Performance

7.4.3 Processor Interconnection

7.5 Distributed Processes

As we recall from our earlier discussions a process is a program in execution. A program calls upon all of the resources of the operating system, the databases and the processors. Thus there is a significant different in a multimedia distributed environment for the functioning of a set of distributed processes. This section explores some of these issues.

7.5.1 DP Types

7.5.2 DP Architectures

Figure 7.27 Layered Architectures

Figure 7.28 Module Architectures

7.5.3 DP Interfaces

Figure 7.29 Distributed Interfaces

7.5.4 DP Performance

Figure 7.30 Block Diagram of Key Factors: (a) Loading, (b) Transport, (c) Distributed OS, (d) Processing

7.6 Conclusions

This chapter developed the overall distributed environment and discussed the key elements of that environment. What should be clear after developing these many issues is that there still are many open questions in distributed environments and these questions are driven by the availability of new technology that allows for interconnection at extremely high data rates. In the past, as we have discussed in the presentation of the ISO model, we have found that the data rates were slow and that it was necessary to provide a significant overlay of protocols to assure end to end performance. A distributed environment in that world was limited to slower acting systems. In the world of unlimited bandwidth as presented by fiber links, a fully distributed architecture can be developed.

edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

7.0 Distributed
8.3 Operating Systems

8.3.1 OS Structures

'Chapter 13 : Distributed Computing Systems' in
'Linked Local Area Networks', second edition
A.J. Mayne
John Wiley & Sons, 1986

'Chapter 2 : Introduction to Operating Systems' ,
'Chapter 4 : Introduction to Distributed Operating Systems',
'Chapter 7 : Process-Based Distributed Operating Systems',
'Chapter 8 : Object-Based Distributed Operating Systems in
'Design of Distributed Operating Systems -
Concepts and Technology'
P.J. Fortier
Intertext Publications, McGraw-Hill, 1986

'Chapter 2 : Basic Concepts'
M.W. Alford, L. Lamport, G.P. Mullery in
'Distributed Systems , Methods and Tools for Specification, an Advanced Course'
M.W. Alford, J-P. Ansart, G. Hommel, L. Lamport, B. Liskov, G.P. Mullery, F. B.
Schneider
Lecture Notes in Computer Science, Springer-Verlag, 1985

'Chapter 3 : Acquisition - Environment'
G.F. Mullery in
'Distributed Systems , Methods and Tools for Specification, an Advanced Course'
M.W. Alford, J-P. Ansart, G. Hommel, L. Lamport, B. Liskov, G.P. Mullery, F. B.
Schneider
Lecture Notes in Computer Science, Springer-Verlag, 1985

'Which Kinds of OS Mechanisms Should be Provided for Database Management?'
P. Christmann, Th. Härder, K. Meyer-Wegener, A. Sikeler in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

8.3.2 OS Elements.

'Chapter 5 : Design and Implementation of Open Systems' in
'Building an Open System'
Jacob Slonim, Abraham Schonbach, Michael A. Bauer, Lachlan J. MacRae,
Keith A. Thomas
Van Nostrand Reinhold, 1987

'Which Kinds of OS Mechanisms Should be Provided for Database Management?'
P. Christmann, Th. Härder, K. Meyer-Wegener, A. Sikeler in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'A Distributed Process Execution and Communication Environment for High-Performance Application Systems'
W. Schröder in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

8.3.4 Distributed OS Issues

'Chapter 4/ Sections 4.2.1, 4.2.2, 4.2.3 : Operating Systems for Distributed and
Parallel Computing/
Distributed Operating Systems/
Some Problems, Interprocess Communication, Resource Sharing' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

'Chapter 6 : Introduction to the Problems of Distributed Computing Systems' in
'Distributed Computing Systems :
Communication, Cooperation, Consistency'
CORNAFION (group name)
Elsevier Science Publishers, 1985

'Chapter 5 : The Process Model',
'Chapter 6 : The Object Model' in
'Design of Distributed Operating Systems -

Concepts and Technology'
P.J. Fortier
Intertext Publications, McGraw-Hill Book Company, 1986

'Reliability Issues in Distributed Operating Systems'
A. S. Tanenbaum, R. van Renesse in
Proceedings of the IEEE Sixth Symposium on Reliability in Distributed Software and Database Systems, 1987

'The CHORUS Distributed Operating System : Some Design Issues'
M. Rozier, J. Legatheaux Martins in
'Distributed Operating Systems, Theory and Practice'
edited by Y. Paper, J-P. Banatre and M. Bozygit
Series F : Computer and Systems Sciences, Vol 28, NATO ASI Series, 1987

'Part 2, Chapter 7 : Designing a Distributed Processing System' in
'An Introduction to Distributed and Parallel Processing'
J.A. Sharp
Computer Science Texts, Blackwell Scientific Publications, 1987

'Distributed System Construction : Experience with the Conic Toolkit'
N. Dulay, J. Kramer, J. Magee, M. Sloman, K. Twidle in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Observations on Building Distributed Languages and Systems'
R.D. Schlichting, G.R. Andrews, N.C. Hutchinson, R.A. Olsson, L.L. Peterson in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Experiences with the Development of a Portable Network Operating System'
U. Hollberg, B. Mattes, A. Schill, H. Schmutz, B. Schöner, R. Staroste, W. Stoll in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

8.3.5 Distributed OS Examples

'Chapter 4/ Sections 4.2.4, 4.2.5 : Operating Systems for Distributed and Parallel
Computing/
Distributed Operating Systems/
Example Systems, Comparison of Examples' in
'An Introduction to Distributed and Parallel Computing'
J.M. Crichlow
Prentice Hall, 1988

'Experience with the Accent Network Operating System'
R.F. Rashid in
'Networking in Open Systems'
edited by G. Goos and J. Hartmanis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Chapter 1, Section 2 : Case Studies of Distributed Computing Systems', in
'Distributed Computing Systems :
Communication, Cooperation, Consistency'
CORNAFION (group name)
Elsevier Science Publishers, 1985

'The CHORUS Distributed Operating System : Some Design Issues'
M. Rozier, J. Legatheaux Martins in
'Distributed Operating Systems, Theory and Practice'
edited by Y. Paper, J-P. Banatre and M. Bozygit
Series F : Computer and Systems Sciences, Vol 28, NATO ASI Series, 1987

'Chapter 1: The LOCUS Architecture' in
'The LOCUS Distributed System Architecture'
G.J. Popek and B.J. Walker
Series in Computer Systems, the MIT Press, 1985

'Distributed System Construction : Experience with the Conic Toolkit'
N. Dulay, J. Kramer, J. Magee, M. Sloman, K. Twidle in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Observations on Building Distributed Languages and Systems'
R.D. Schlichting, G.R. Andrews, N.C. Hutchinson, R.A. Olsson, L.L. Peterson in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis

Lecture Notes in Computer Science, Springer-Verlag, 1987

'Experiences with the Development of a Portable Network Operating System'
U. Hollberg, B. Mattes, A. Schill, H. Schmutz, B. Schöner, R. Staroste, W. Stoll in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Accomodating Heterogeneity'
J. Zahorjan, E.D. Lasowska, H.M. Levy, D. Notkin and J. Sanislo in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'Process management in a Distributed Operating System'
S. J. Mullender in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

'A Distributed Process Execution and Communication Envionment for High-
Performance Application Systems'
W. Schröder in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

8.3.6 Distributed OS Performance Factors.

'A Measurement Study of Load Balancing Performance'
S. Zhou, D. Ferrari in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Analysis of Three Dynamic Distributed Load-Balancing Strategies with Varying
Global Information Requirements'
T. L. Casavant in

'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'A Performance Study of Dynamic Load Balancing in Distributed Systems'
A. Thomasian in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Dynamic Load balancing in a Distibuted System Using a Decentralized Algorithm'
A. Hác , X. Jin in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Which Kinds of OS Mechanisms Should be Provided for Database Management?'
P. Christmann, Th. Härder, K. Meyer-Wegener, A. Sikeler in
'Experiences with Distributed Systems'
edited by G. Goos and J. Hartmannis
Lecture Notes in Computer Science, Springer-Verlag, 1987

8.4 Processors.

8.4.1 Processor Types.

'Communications Processors' in
'Datapro'
Datapro Research Corporation, April 1988

'Part2, Chapter 5 : Distributed Systems' in
'An Introduction to Distributed and Parallel Processing'
J.A. Sharp
Computer Science Texts, Blackwell Scientific Publications, 1987

8.4.2 Processor Performance.

'Allocation and Relocation of Processes in Distributed Computer Systems'
C.N. Nikolau, G. Kar, D.F. Ferguson, G. Leitner in
'Current Advances in Distributed Computing and Communications'
edited by Y. Yemini
Computer Science Press, 1987

8.4.3 Processor Interconnect.

'A Simple Analysis of Expressnet'
P. Jacquet, P. Mühlethaler in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'A 100Mb/s Optical Token Ring Network Suitable for High-Speed Inter-Processor Communications'
T. Hoshiko, Y. Kimura, S. Tomita in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'An Interconnection Topology for Fault-Tolerant Multiprocessor Systems'
A. Ghafoor, T.R. Bashkow, I. Ghafoor in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'An Interconnection Network Supporting Ralational Join Operations'
B. Menezes, D. Brant, D. Loewi, A. Dale, R. Jenevein in
'1987 IEEE 7th International Conference on Distributed Computing Systems'
edited by R. Popescu-Zeletin, G. Le Lann, K.H. Kim
Computer Society Press of the IEEE, 1987

'Design Issuses in Networking Incompatible Operating Systems'
N.D. Johnson in
'Localnet' 86'
Online Publishers, 1986

'Part2, Chapter 6 : Networks and Interconnection Structures' in
'An Introduction to Distributed and Parallel Processing'
J.A. Sharp
Computer Science Texts, Blackwell Scientific Publications, 1987

CHAPTER 9

Conclusions

In this book, we have developed a view of multimedia communications that combines the structure of the underlying images and the storage and transport media along with a careful understanding of the end user and their interactions with the different forms of media. Unlike the world of computer communications where the computer can be trained to perform certain tasks that are required for proper performance, the human in multimedia application is operating in a mode of total creativity. This creativity results in complex and often unpredictable interactions with the other human users. This interaction is the essence of the multimedia communications theory.

We have tried in this book to present a canonical structure to view the multimedia multiuser environment, focusing on the concept of the session. We have not focused on the psychological inferences and actions of the end user in great detail. This is the artistic side of multimedia communications and expands beyond just knowledge engineering or even epistemology.

9.1 Key Issues

This book has developed a philosophy of multimedia communications that tries to separate the "artistic" element from the engineering element. It may be criticized that such a separation is difficult if not impossible, but our intentions were to gain a

better focus on the system ideas by quantifying the basis elements and not serializing them. Thus the approach has been alone the more common engineering lines, understanding that the system definition and design are the key characteristics.
We can now review the overall key issues that we have developed in this book and present what has been developed in context.

Multimedia communications, as we have seen, deals with the handling, manipulation, processing and transmission of complex images that are highly interactive in nature. The insensitivity factor focuses on the "conversational" mode of communications that we have discussed.

- o The Overall System Approach
- o Characterization
- o Performance and Sizing
- o Feel and Form versus Function and Formalism
- o New Paradigms of Presentation
- o Displaced Working Environments
- o End User as Design Element
- o Sessions
- o Distributed Environments

9.2 Future Trends

The evolution of multimedia communications will be enhanced by the development and deployment of broadband communications as well as the development of smarter and more sophisticated end user terminals. The most difficult problem is the issue of how do we effectively interface with the human end user. Specifically what common paradigms are most effective in what environments.

The trends that impact this area of multimedia communications are quite simple to articulate at this stage but they may change as we enter into a time of rapid technological change. Specifically, if we look at the model of the multimedia environment, and we map prognosticated changes in each of these areas against the separate elements of the multimedia domain, we can carefully determine the impact of the multimedia environment and determine the returns to that environment.

Let us focus on the specific multimedia areas and discuss the trends in each of those areas that will impact the development of an improved and more advanced multimedia communication system.

- o Source Characterization
  - o Image Compression
  - o Speech Recognition
  - o Speech Generation
  - o Print Technologies

- o Scanning and Entry Technologies
- o Character Recognition
- o Image Element Abstraction

o End User Interface

- o Enhanced Entry Mechanisms
- o High Resolution Displays
- o To HDTV or Not to HDTV
- o Alternate Sensory Inputs (Is Touch in the Future?)
- o Optimal Interface design Methodologies

o Memory and File Storage

- o Terabit memory Storage
- o Multimedia Data Base Access Systems
- o Multimedia Object Creation and Dissemination
- o Standard versus Non Standard Files
- o Nano Second Access Times
- o Cached High Density Files

o Communications

- o Gigabit Data Channels
- o Local Versus Long Distance Data Transport
- o Local Network Architectures
- o Local versus Distributed Intelligence
- o Connectionless versus Connection Based
- o Shared versus Private Networks

o Databases

- o Multimedia Database Accesses
- o File and Database optimization
- o Concurrency in Multimedia Forms
- o Recovery of Multimedia Forms
- o Multimedia Database Allocation Algorithms

o Operating Systems

- o Distributed Multimedia Applications
- o Multiple Platform Applications

These trends provide for a brief introduction of some of the options that are available to the user and developer of multimedia communications networks. They present a view of what is possible but do not present what will actually occur. We have developed some of these issues based upon what is difficult to do in toady's environment but new technology will most certainly alter this view.

9.3 Major Research Directions

The discussion in the past section turns on views of future trends that are natural follow on to what we have discussed in the body of the text. In this final section of the book we wish to share some of the future research directions that are open and whose solutions provide the most fruitful ground for future work.

- Multimedia Database Structures
- End User Interface Language Design
- WAN Communications Architectures
- Distributed Operating System Optimization
- Overall System Optimization Performance
- Multi Sense Interfaces

9.4 Summary

This book presented a vision of multimedia communications as more tan just an amalgam of separate parts, disembodied from the ultimate user. It presents a holistic view that recognizes th impact of the user on the services and the underlying technology. Multimedia communications is a concept that is in its infancy but that also offers the user a significant increase in performance, functionality and the ability to disembody themselves from the medium of communications.

As was stated earlier in the book, Marshall McLuhan stated that the introduction of a new medium changed not only how we presented information, but ultimately what was considered knowledge. This is both a challenge and a warning. The technological concepts presented in this book provide for a significantly different view of media or presentation technology. It raises the question of how best to interface the end user to a network that responds in near instantaneous speed at almost all data rates. It, in effect, introduces a plethora of new media, thus challenging and warning us simultaneously.