

**Efficient Transfer and Storage of Image Data For Distributed
Development of Biomedical Imaging Tools**

by

Roger C. L. Tam

B.Sc., University of British Columbia, 1994

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
Master of Science

in

THE FACULTY OF GRADUATE STUDIES
(Department of Computer Science)

we accept this thesis as conforming
to the required standard

The University of British Columbia

August 1997

© Roger C. L. Tam, 1997

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of COMPUTER SCIENCE

The University of British Columbia
Vancouver, Canada

Date 97/08/13

Abstract

The central idea behind a medical informatics testbed is the sharing of geographically distributed health information resources. Medical image data is one of the most important resources that can be shared between clinical and research sites. In this thesis, we build an infrastructure for the efficient transfer and storage of medical image data to support distributed development of biomedical imaging applications. The Digital Imaging and Communications in Medicine (DICOM) standard provides interoperability in our testbed. In addition, properties of DICOM are used to help the biomedical imaging tool developer process the images. Performance analyses are done to measure the efficiency of various layers of our implementation.

Contents

Abstract	ii
Contents	iii
List of Tables	v
List of Figures	vi
Acknowledgements	viii
1 INTRODUCTION	1
1.1 Motivation	4
1.2 Clinical Environment	6
1.3 Research Environment	7
1.4 Thesis Statement	10
2 MODEL	12
2.1 Introduction	12
2.2 Digital Imaging and Communications in Medicine (DICOM)	14
2.3 Testbed Model	21

3	Implementation	26
3.1	DICOM Service-Object Pair Classes	26
3.2	Central Test Node Facilities (MIR Libraries)	31
3.3	ALI DICOM Support Suite	33
3.4	Applications	36
4	Performance and Results	39
4.1	ATM Network Performance	39
4.2	DICOM Performance	42
4.3	Application Performance	43
4.4	Evaluation of Model	46
5	Related Work	51
6	Conclusions and Future Work	54
6.1	Summary and Conclusions	54
6.2	Future Work	55
	Bibliography	57

List of Tables

3.1	SOP Classes for Radiology	27
3.2	SOP Classes for ALI PACS	28
3.3	SOP Classes for Oral Biology	29
3.4	SOP Classes for BIT Development Lab	30
4.1	CTN Facilities Performance Statistics	42
4.2	Imager-to-PACS Simulation Results for Image Size 525350	45

List of Figures

1.1	Diagram of Network Connections	2
1.2	Model of Distributed Biomedical Imaging Tool Development	3
1.3	Processing Pipelines for Clinical and BIT Research Environments	8
1.4	Image Information Models for Clinical and BIT Research Environments	9
1.5	Requirements for Transfer and Storage of Image Data for Distributed BIT Development	10
2.1	Zeroth Level Model of DICOM Integration	13
2.2	DICOM Protocol Stacks	14
2.3	(a) DICOM Model of the Real World (Simplified) and (b) DICOM Information Model (Simplified)	17
2.4	Image Information Object Definition (IOD)	18
2.5	DICOM Service Exchange	19
2.6	SCU/SCP Roles of Retrieve Operations	21
2.7	DICOM SCU/SCP Roles of BIT Development Testbed	22
2.8	Model for Integrated Query/Retrieve/Filter Application	23
2.9	Model for Data Re-Organization	24
3.1	Calling Hierarchy of DICOM Applications	32

3.2	Calling Structure of CTN Facilities	32
3.3	Calling Structure of ALI Support Suite	34
4.1	Scanning and 3D Reconstruction	46
4.2	Segmentation of a 2D CT Slice	47

Acknowledgements

Many thanks to Dr. Peter Cahoon and Dr. Gerald Neufeld for supervising this thesis. Thanks also to Dr. Kellogg Booth for his invaluable advice. Funding and equipment for this project were provided by a Natural Sciences and Engineering Research Council Industrial Postgraduate Scholarship, ALI Technologies, and IBM.

ROGER C. L. TAM

The University of British Columbia

August 1997

Chapter 1

INTRODUCTION

This thesis discusses a collaborative project that is part of a larger initiative to develop a testbed that will serve as a prototype for a distributed health care information and communication system. The central motivation for this testbed is to facilitate the sharing of limited resources between geographically distributed clinical and research locations by using a high-bandwidth computer network. Each site connected to the network requires resources only available at other sites or has resources required by other sites. The three general categories of resources under consideration are hardware, software and personnel having domain-specific knowledge. An example of a hardware resource is the medical scanners located in the hospitals; the scanners are the primary source of medical image data. An example of a software resource is the image processing software located in Computer Science. An example of domain-specific experts is the radiologists at the hospitals.

In our testbed, the sharing of a resource is accomplished by distribution of the data produced by that resource. There are a number of software applications currently being developed on the testbed, each suited for distributing certain types

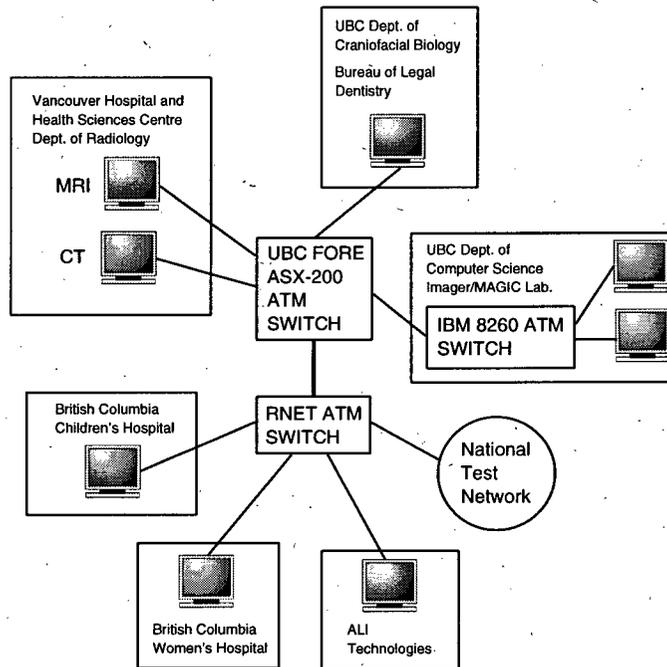


Figure 1.1: Diagram of Network Connections

of data. Examples of such applications include programs for the transfer and storage of patient information, video and voice links for teleradiology, and three-dimensional shared workspaces that allow a number of users across the network to simultaneously view and manipulate models in virtual space [CG96, KC95].

Figure 1.1 shows a diagram of the network connections in our testbed. The hospitals, which include Vancouver Hospital and Health Sciences Centre, British Columbia Children's Hospital and British Columbia Women's Hospital, are the sources of patient information such as image data from medical scanners. They are also the primary source of medical expertise. ALI Technologies is a company that

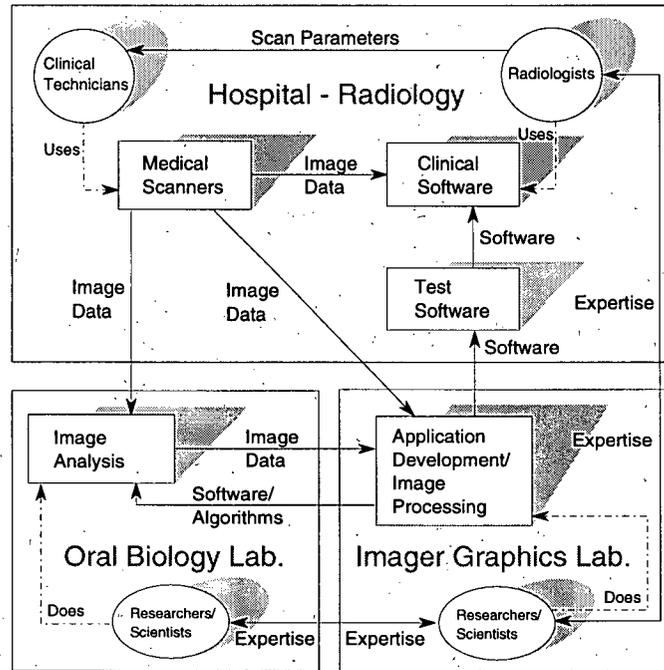


Figure 1.2: Model of Distributed Biomedical Imaging Tool Development

specializes in picture archiving and communication systems (PACS). ALI provides certain sites on the network with hardware and software for storage and retrieval of medical images. Engineers at ALI also provide technical knowledge in areas such as constantly evolving standards in medical informatics. The Oral Biology site, which includes the Craniofacial Biology Laboratory and Bureau of Legal Dentistry, is a source of data and expertise in areas such as craniofacial reconstruction and forensic dentistry. The Imager Computer Graphics Laboratory in Computer Science houses specialized graphics workstations and software to process image data from the other nodes, and provides technical knowledge in computer-related areas.

1.1 Motivation

One of the primary roles of the Imager Laboratory on this testbed is to develop new biomedical imaging applications. Examples of such projects include customization of a three-dimensional mesh generator (Voxview3D) for use in model building in Oral Biology, computed tomography (CT) image segmentation filters for volume rendering in Radiology, and a finite element model of the human ventricular system for BC Children's Hospital. Using a client-server model of distributed processing, we can view the Imager Lab as an application development server that takes image data and domain-specific information from other nodes as input, and returns new applications and/or processing algorithms as output. (Note that applications developed for users at a certain site need not be installed at that site; the network allows many applications to be used remotely.) Figure 1.2 shows a high-level diagram of this model, with Radiology and Oral Biology as examples of clients. Implementing this model requires building an infrastructure that integrates the many hardware and software components from these heterogeneous clinical and research environments. This is a large and complex problem. The solution requires finding a suitable way to divide the problem. The model shows that information flowing between sites can be in the form of image data, software or expertise. By focussing on one of these types of data, we can narrow down the context to form a manageable subproblem.

In this thesis, we focus on the distribution of image data, because images occupy the largest component of network bandwidth and require the most organization for storage. The amount of data is rapidly growing, thus creating the need for an effective management system. The principal objective of this project is to design and implement a system to provide efficient image data transfer and storage to support distributed research and development of biomedical imaging tools (henceforth

abbreviated BIT). An "efficient" design should maximize transfer speed, interoperability and adaptability, while satisfying the individual requirements of each site.

The original method of transferring image data to the Imager Lab involved a disorganized series of actions. If a clinical site, such as Radiology, wanted to send data, a user at the site would contact a user at the Imager Lab via the telephone or e-mail, and give out information, such as a patient's name or ID, to identify the data set. The BIT developer would then login to the hospital computer system, and search the patient directories for the correct data set, then use FTP to retrieve the data. Often, there were follow-up calls to obtain information such as scan parameters. The users at clinical sites do not use FTP to initiate the transfer process, because most are unfamiliar with system details such as UNIX file systems and commands, and find it very inconvenient to switch away from clinical software to perform file transfers. Besides being inefficient, the original method of data transfer required the remote BIT researchers to be given login access to hospital systems, which potentially compromised the privacy of all patients, including those whose data sets are not involved in a BIT research project. Other sites that need to send data to the Imager Lab, such as research labs in Oral Biology and Children's Hospital, have users that can use FTP to send images to the Computer Science anonymous FTP site. However, images from these research sites usually vary much more in format than those from clinical sites, and it was often quite time-consuming for the BIT researcher to try to figure out how to process and convert the images to a usable format.

Before development of the testbed, image storage was also disorganized. Patient data at the Imager Lab was distributed over a number of file systems, with varying directory structures. This often caused problems when a user tried to locate

a particular data set. In addition, many images were stored in the original formats from the source site, and it was sometimes difficult to recall how a particular format should be processed and converted, unless auxiliary files describing the data were created and kept with the data.

Clearly, a new system for the transfer and storage of image data was needed. Before producing a model for the new system, we needed to identify the needs of different users in each environment.

1.2 Clinical Environment

The clinical environment is centred in hospital radiology departments. There are a number of medical scanners that produce images in various formats, each designed to be processed differently. The scanners are connected, either directly or via a local area network, to workstations that are used for scanner control and/or processing and display of the images. The radiologists use the software on image review workstations to control the parameters for image display, thereby enhancing the image features they want to focus on. Currently, the radiology departments on our testbed are mostly using commercially developed packages sold by scanner manufacturers or third-party vendors. For example, the Radiology Department at Vancouver Hospital (UBC site) uses the General Electric Advantage Workstation system to review magnetic resonance imaging (MRI) images. Because such systems are designed for clinical purposes, the user-interface hides lower-level details such as internal image formats (*e.g.*, header size and location, encoding methods, proprietary formatting, etc.) and image processing algorithms (*e.g.*, conversion from stored image values to displayed pixel values).

In a radiology department, there are two groups of users that are usually the

most involved in biomedical imaging research. Radiologists, the first group, are the physicians that direct the acquisition of medical images and make diagnoses with the aid of such images. Clinical technicians, the second group, are responsible for physically operating the scanners. In providing image data for other sites on the testbed, these users have a number of primary requirements. Firstly, they would like to be able to activate transfer processes to remote sites with minimal disruptions to clinical operation. Most users in a clinical environment do not normally need to know much about the computers they are using beyond functional familiarity with commercial software. Therefore, any attempts to integrate clinical workstations into the testbed should take advantage of the built-in image archiving facilities of existing software whenever possible. This would minimize the need for users to learn new software, and prevent users from having to switch back and forth between programs. Fortunately, most newer commercial image review workstations have at least some image archival capabilities. Another primary concern of clinical users is patient confidentiality. Users from remote sites should not be able to access locally stored patient data without authorization. Also, data stored at remote sites should be well-protected from access outside of the testbed.

1.3 Research Environment

The research environment comprises a number of laboratories, with the Imager Lab as the primary BIT development site. There are research labs in Oral Biology and BC Children's Hospital that also work with medical image data, but they do not usually develop new applications. In contrast to the clinical user, the BIT developer wants to start with images that are as "raw" as possible, because the latter stages of processing for clinical display often render the images unsuitable for

use in developing new applications. Figure 1.3 shows the data processing pipelines used by the clinical and BIT research environments to generate images suitable for their respective purposes.

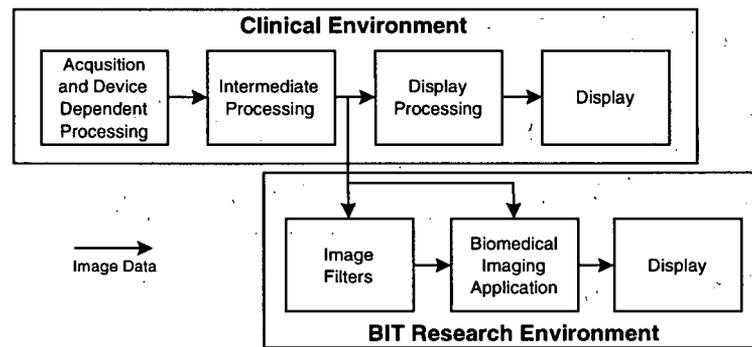


Figure 1.3: Processing Pipelines for Clinical and BIT Research Environments

In the BIT pipeline, there is a filtering stage that is used for purposes such as header extraction and/or processing, image format conversion, and feature extraction. Which filters are used depends relatively little on the main imaging application, but mostly on data properties such as the source modality (CT, MRI, etc.), scan parameters, image dimensions and image encoding methods (*e.g.*, bits per pixel, byte ordering, compression methods). In order to promote modularity and code reuse, these filters are not integrated into the main application until the end stages of development. The researcher would like to integrate the filtering process into the image retrieval process. The advantage of an integrated process would be image data arriving to the researcher in a “ready-to-use” form.

The BIT developer also needs organized storage of image data, both in the raw and filtered forms. Ideally, data storage should be centralized, with one database for all the sources of images. Users from remote sites that need to send images to

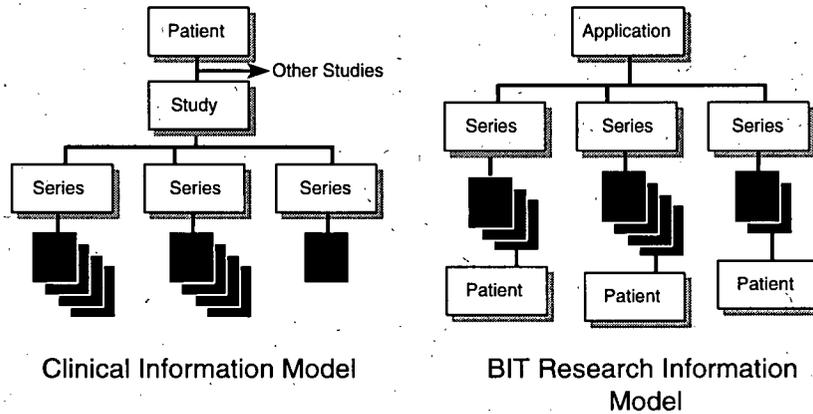


Figure 1.4: Image Information Models for Clinical and BIT Research Environments

the Imager Lab should be able to do so directly to the image storage server, which would allow the BIT developer to retrieve the data at his convenience. Centralized storage also provides for better security because of easier access control. To allow direct transfer from other sites, the communications sub-system of the database should be interoperable with hospital software, as well as data transfer software in other research labs such as Oral Biology.

The research environment uses a different image information model from that in the clinical environment, and it would be very beneficial if the database contained a facility to re-organize data according to the research information model. Figure 1.4 shows the two different information models. The clinical information model is patient-centred, and is based on how various real-world entities in radiology operations are related. The BIT research information model is designed to be application-centred, with the idea that data sets can be grouped according to which imaging application requires them for testing. The goal of this re-organization is to allow the BIT developer to specify to the database a certain application identi-

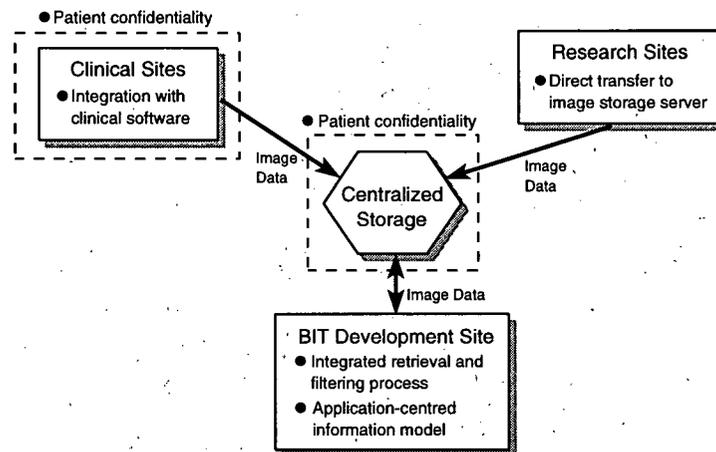


Figure 1.5: Requirements for Transfer and Storage of Image Data for Distributed BIT Development

fication, and be given a list of data sets for that application as output. Note that the user does not care about the internal storage structure of the database, as long as the interface allows any data set to be marked and identified as belonging to a certain group.

1.4 Thesis Statement

Figure 1.5 summarizes the requirements for the transfer and storage of image data to support distributed development of BIT's. The key to a system that satisfies all of these requirements lies in using a common image transfer protocol and storage format. This thesis shows that the medical informatics standard known as Digital Imaging and Communications in Medicine (DICOM) 3.0 is an appropriate protocol and image format on which our testbed can be built. While the use of DICOM to facilitate horizontal integration in a medical informatics testbed is not new, this

work makes contributions in showing how DICOM can be vertically integrated into a research environment for developing BIT's. More specifically, we show how particular properties of DICOM are used to satisfy the individual requirements of each site.

Chapter 2 of this thesis presents a model for the system that we have implemented. The concepts and relevant properties of DICOM 3.0 are explained. An application profile for each site is done to determine which parts and/or properties of DICOM are needed and how they should be applied.

Chapter 3 of this thesis discusses the implementation of the system. The main modules of the software developed for this testbed are described, along with an overall view of the use of each program.

Chapter 4 discusses the performance of the testbed. The results of performance analyses of the Asynchronous Transfer Mode (ATM) network, DICOM protocol and application efficiency are summarized. Then an evaluation of how well the implementation meets the goals of the systems are given, along with an example of an application being developed on the testbed.

Chapter 5 discusses related work, and compares and contrasts our work to that of others.

Chapter 6 presents the conclusions, and gives future directions for this project.

Chapter 2

MODEL

2.1 Introduction

This chapter presents the model for the system that we have implemented for efficient image data transfer and storage. It introduces the design rationale and basic concepts of the DICOM standard, and details the properties that make it suitable for our purpose. This chapter then gives an abstract view of how DICOM is used to provide an image transfer platform that can be integrated into each site to support BIT development. Figure 2.1 shows the zeroth level diagram of the model.

In the clinical environment, DICOM provides compatibility with clinical software. Most recent clinical image review stations (*i.e.*, those produced within the last two to three years), such as GE's Advantage Workstation, feature at least a minimal set of DICOM-compliant picture archival functions. In most cases, this means that the system converts images from its proprietary format to DICOM, provides a communications front-end that can send and receive DICOM messages, and integrates access to these functions into the user-interface. With the proper con-

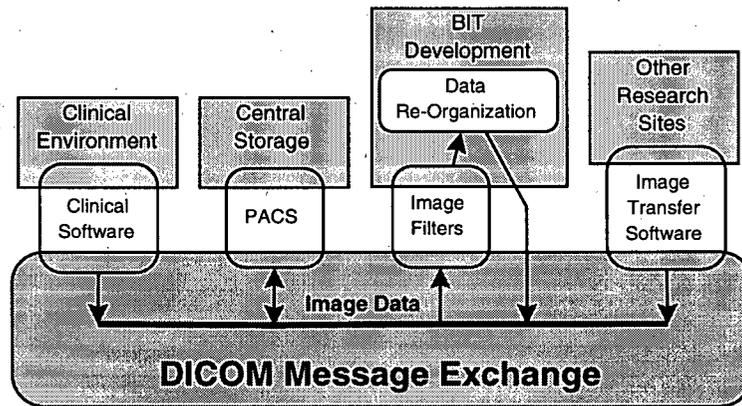


Figure 2.1: Zeroth Level Model of DICOM Integration

figuration, such a system can communicate directly with a PACS back-end that is responsible for responding to image storage and retrieval commands and interfacing with storage media. A DICOM 3.0 compatible PACS back-end (ALI UltraPACS) is used as the central storage unit in our testbed.

In the Imager Lab BIT development site, various software filters are used to pre-process images into a suitable form for BIT testing. DICOM facilities are used to integrate the filtering process into the image retrieval process. Properties of DICOM are also used in software to re-organize data to conform to the BIT research information model (Figure 1.4).

The other biomedical imaging research sites, such as Oral Biology, primarily want to be able to send images directly to the PACS. Since most users at these sites are familiar with the UNIX operating system, and there is no requirement for integration with clinical software, simple command-line DICOM applications have been written to allow communication with the PACS.

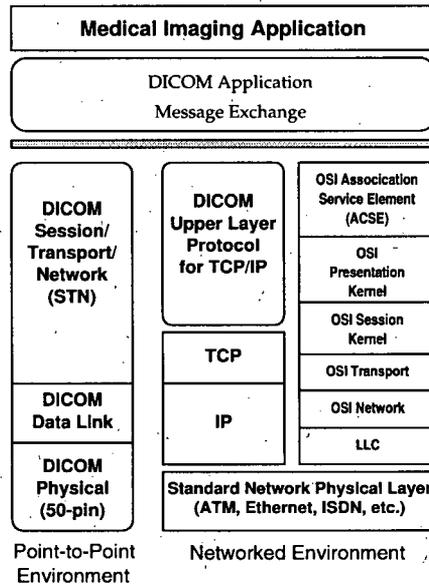


Figure 2.2: DICOM Protocol Stacks

2.2 Digital Imaging and Communications in Medicine (DICOM)

A general definition of DICOM is that it is a standard for the communication of medical images and associated information. The standard specifies a communications protocol that consists of a number of layers below the application level. Figure 2.2 shows the DICOM protocol stacks for point-to-point, TCP/IP and OSI network connections. In addition, DICOM also specifies an image format for medical data.

The current DICOM standard evolved over a decade from earlier standards. In 1983, a joint effort between the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) was initiated to develop a standard for interfacing between digital medical imaging equipment (*e.g.*, CT, MRI,

ultrasound, etc.) and other devices, such as image processing workstations or film printers. In 1988, version 2.0 of the ACR-NEMA standard was published. However, by this time many users wanted a standard that supported robust network communication. ACR-NEMA Version 2.0 was designed for point-to-point connections, and supported network communication only at a low level. For example, a user could send an image to a remote device, but the standard did not specify what the device would do with the image. As networks grew larger, this became increasingly more problematic. In a large network, the number of devices and the distances between them can make it very difficult for any two devices trying to communicate to know precisely what each other's capabilities are. This is especially true if some devices are dynamically configured to handle different tasks. This gave motivation for the ACR-NEMA committee to develop a high-level communications platform for medical data transfer.

Major revisions and additions to the ACR-NEMA standard using object-oriented design resulted in DICOM 3.0, finalized in 1994. DICOM contains a number of major enhancements to ACR-NEMA 2.0. These improvements, as specified in Part 1 of [Ame94], include:

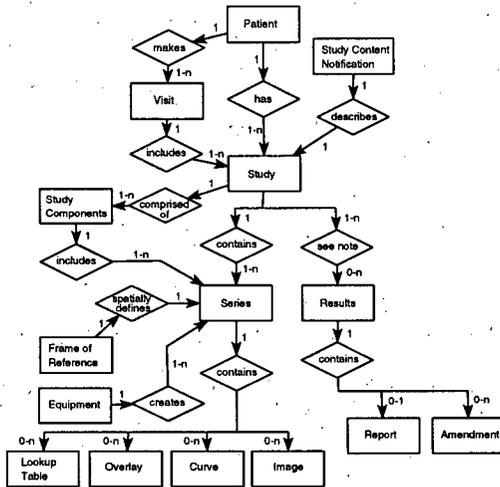
- A Network Interface Unit (NIU) was added to support operation in a networked environment using standard protocols such as TCP/IP and OSI. In our testbed, TCP/IP is used as the underlying network protocol, and in the protocol stack the NIU is referred to as the DICOM Upper Layer Protocol for TCP/IP. Figure 2.2 shows how this layer lies between the TCP and DICOM Application Message Exchange layers.
- DICOM specifies how standard-compliant devices react to commands and data being exchanged. Previous versions only dealt with the transfer of data, and

relied upon the end-applications to know what to do with it. DICOM specifies the semantics of commands and associated data.

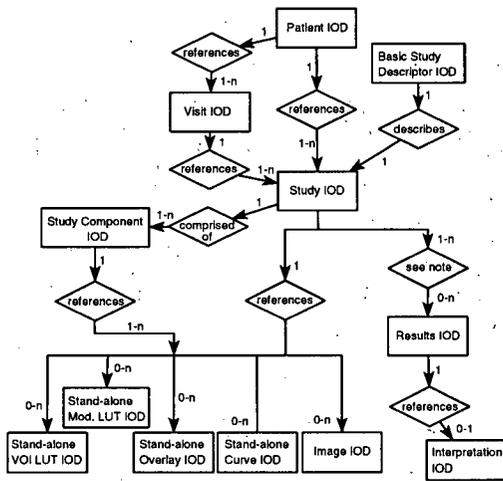
- DICOM specifies how an implementor claiming conformance to the standard must structure a Conformance Statement to select specific options. It also specifies levels of conformance, instead of only specifying a minimum level of conformance as in previous standards. This gives us a way to unambiguously state the DICOM capabilities of each application or device.

DICOM is based on models of real entities, relationships and operations in a radiology department. Figure 2.2(a) shows a simplified version of the model of real-world objects and relationships used by the designers of DICOM. For example, the model contains the relationships “Patient has Study”, “Study has Series” and “Series has Images”, which reflect real-world relationships. In DICOM, an abstraction of a real information entity (*e.g.*, CT Image) is called an *Information Object*. Figure 2.2(b) shows a simplified version of the information model used for DICOM, and shows how various Information Objects are related to each other. For example, the Patient Information Object references the Study Information Object, which in turn references the Image Information Object. Each Information Object Definition (IOD) consists of a description of its purpose and a set of *Attributes*, which are properties of the object. An IOD consisting of a single information entity is called a *Normalized IOD*, and an IOD that is a combination of information entities is called a *Composite IOD*. Figure 2.4 shows a Composite image IOD, consisting of five information entities: Patient, Study, Series, Equipment and Image. Examples of Attributes include Patient Name, Study Date and Image Type.

DICOM also defines operations, called *services*, that can be performed on objects. For example, “store image” is a service that can be performed on an



(a)



(b)

Figure 2.3: (a) DICOM Model of the Real World (Simplified) and (b) DICOM Information Model (Simplified)

Note: the relationship between the study and result is complex, involving other real-world objects (*e.g.*, the interpreting physician) that are not modelled by the standard.

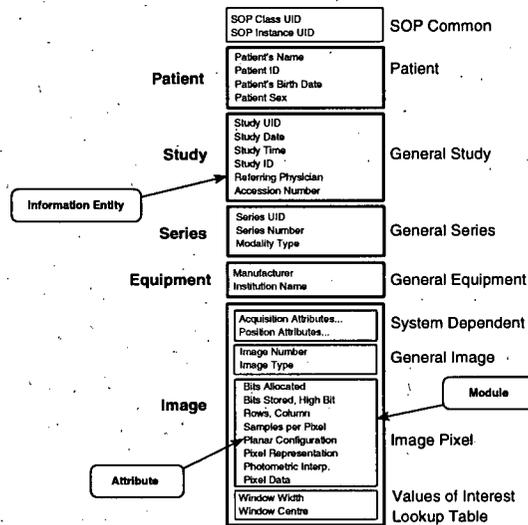


Figure 2.4: Image Information Object Definition (IOD)

instance of an image IOD. Other examples of services that are permissible on image Information Objects include “find”, “get” and “move”. The combination of an IOD and one or more such services forms a *Service-Object Pair (SOP) Class*. For example, DICOM contains a CT Image Storage SOP Class, which is the combination of a CT image IOD and the “store” service. The operations allowed on Information Objects for a certain SOP Class are called the *Service Elements* of that class. For a particular SOP Class, a device may serve either as a *Service Class Provider (SCP)* or a *Service Class User (SCU)*. In the “store image” example, an SCU would initiate a request to store an image, and an SCP would respond to the request.

Figure 2.5 shows a diagram of how DICOM services are exchanged over a network. The part of each process responsible for communications is called an *Application Entity (AE)*. Each Application Entity has an *Application Entity Title* that has to be used when setting up communication between processes. The connec-

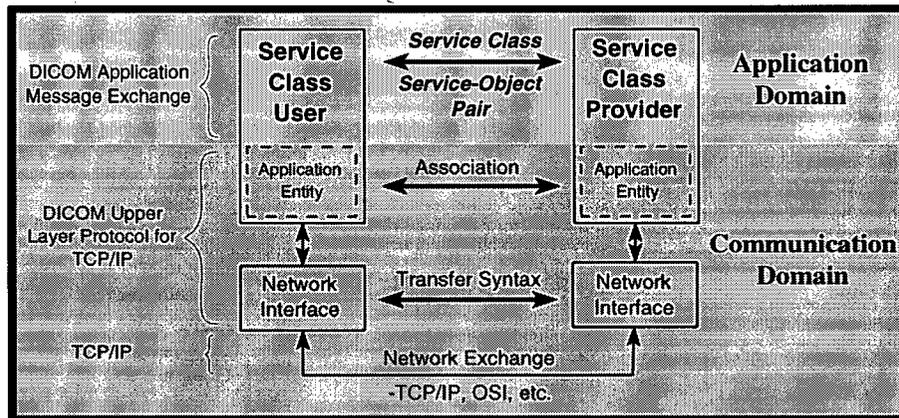


Figure 2.5: DICOM Service Exchange with the corresponding layers of the protocol stack on the left

tion between two Application Entities is called an *Association*. Before information exchange can occur between two DICOM applications, their respective Application Entities must negotiate for an Association. The initiating partner of the Association proposes the SOP Classes to be used and the SCU/SCP role for each class. The receiving side decides whether it will accept the Association, and if so, which SOP Classes it will accept. After the negotiation process, each side knows exactly what the other's capabilities and limitations are, and can begin the transmission of SOP instances.

DICOM groups SOP Classes into a number of general categories called Service Classes that are named as follows:

- Verification Service Class
- Storage Service Class
- Query/Retrieve Service Class
- Study Content Notification Service Class

- Patient Management Service Class
- Study Management Service Class
- Results Management Service Class
- Print Management Service Class

This project only deals with the first three Service Classes. The first, Verification Service Class, is intended for testing connectivity. It only contains one SOP Class, simply called the Verification SOP Class. An SCU of this class would send a C-ECHO request to a remote SCP that, if functioning properly, would issue a C-ECHO response. (The "C-" indicates a service that is applicable only to Composite IOD's.) The second Service Class, called the Storage Service Class, provides basic support for the transfer of images between DICOM applications. It uses the C-STORE Service Element that defines the behaviour of the sending SCU and receiving SCP. The third Service Class is called the Query/Retrieve Service Class, and contains SOP Classes to access and move images based on simple search criteria (*e.g.*, get all of the images for a particular patient). This Service Class uses the C-FIND, C-MOVE and C-GET Service Elements. C-FIND is used to query for a set of images matching one or more particular identifiers. Each search key can be at one of four levels (Patient, Study, Series, or Image). The C-MOVE and C-GET services are used to initiate data retrieval. The Query/Retrieve Service Class utilizes the Storage Service Class for the underlying data transfer processes. Figure 2.6 shows the SCU/SCP roles in a retrieval operation. A Retrieve Service SCU acts as an Storage Class SCP in order to receive data. Accordingly, a Retrieve Service SCP acts as a Storage Class SCU.

There are two primary differences between a C-MOVE and C-GET operation. The first deals with the number of Associations. C-STORE sub-operations

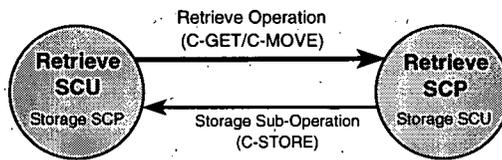


Figure 2.6: SCU/SCP Roles of Retrieve Operations

performed in a C-GET are done under the same Association as the C-GET. In a C-MOVE, the C-STORE sub-operations are done under a separate Association. The second difference is that a C-MOVE can be used to initiate a C-STORE to a third party Application Entity, whereas a C-GET operation can only initiate data transfer to the original C-GET SCU.

2.3 Testbed Model

This section gives an abstract description of how DICOM is used to support distributed BIT development in our testbed. The requirements of each site are used to determine which Service Classes and SCU/SCP roles are needed. The Vancouver Hospital Radiology Department is used to represent all clinical sites, and Oral Biology is used to represent all BIT research sites other than the Imager Lab. This section also presents models for the integration of the query/retrieval and image filtering processes, and the re-organization of data to fit the BIT research information model.

The SCU/SCP roles of each site must be appropriately determined to facilitate and regulate data flow. Figure 2.7 shows the Service Classes and SCU/SCP roles at the Radiology, Oral Biology and Imager Lab sites. Software in Radiology will serve as an SCU of the Storage Service provided by the PACS. Radiology only

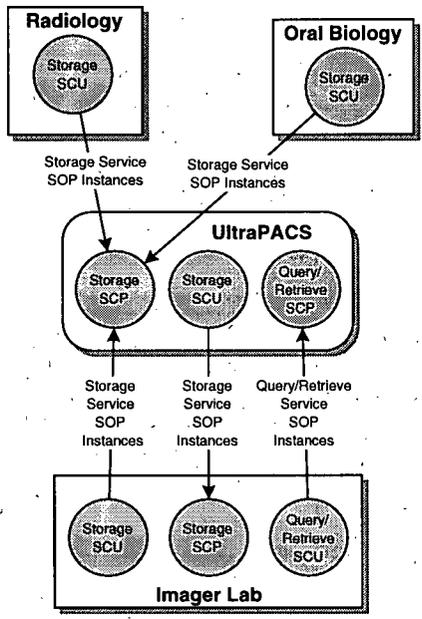


Figure 2.7: DICOM SCU/SCP Roles of BIT Development Testbed

needs an SCU, and not an SCP, for the Storage Service Class, because it only acts as a source, and not a sink, for image data. It is important for the PACS to be able to handle all image modalities from the hospitals. The particular IOD's and SOP Classes required are discussed in Chapter 3, which details the implementation of the system. The Radiology site is designed not to provide Query/Retrieve Service Class SCP's for remote SCU's on the testbed, because of the security requirements. Non-clinical users should not be able to query the hospital's local database and retrieve sensitive data not related to BIT development projects. This configuration gives Radiology staff control over which images are released to other sites.

Oral Biology also requires a Storage Service SCU to partner with the PACS' SCP. This will allow researchers at this site to send images directly to the central storage device. The PACS is configured not to support the Query/Retrieve Ser-

vice Class for Oral Biology and similar sites, so that the central image server is protected from unauthorized access. Consequently, Oral Biology has no Storage or Query/Retrieve Service SCU's.

The Imager Lab makes use of both the Storage and Query/Retrieve Service Classes on the PACS. It requires both an SCU and SCP in the Storage Service Class, in order to send and receive images from storage. It also needs a Query/Retrieve Service SCU, so that it can find and retrieve data sets from the PACS. The PACS is a Storage Service SCP for all testbed sites, and is a Storage Service SCU and Query/Retrieve Service SCP for the Imager Lab only.

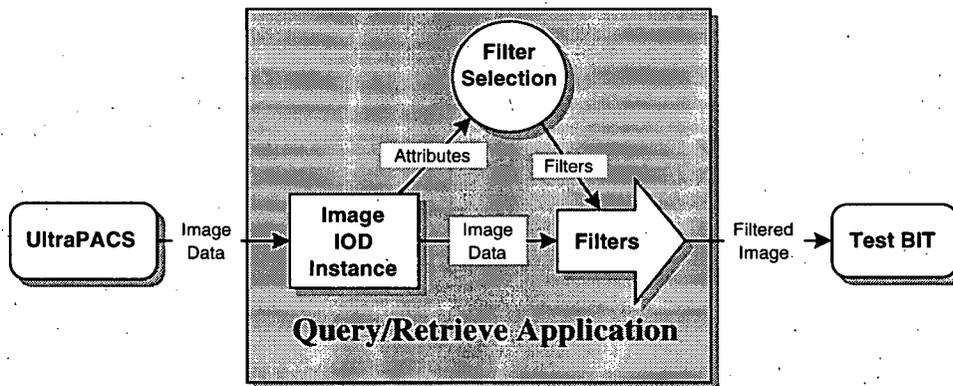


Figure 2.8: Model for Integrated Query/Retrieve/Filter Application

The model in Figure 2.7 satisfies the requirements in the clinical and research environments, with the exception of the Imager Lab BIT development site. As discussed in Chapter 1 and Section 2.1, the Imager Lab requires software filters for pre-processing of the data. These filters can be integrated as shown in Figure 2.8. As each image is received, its Attributes are read by the Query/Retrieve Application. One or more filters are then chosen based on the values of the Attributes. The filter selection process can be automatically, semi-automatically or manually performed,

but is always directed by values of the DICOM Attributes and the target BIT. The filters are then applied to the image, which results in an image ready for input into the biomedical imaging application.

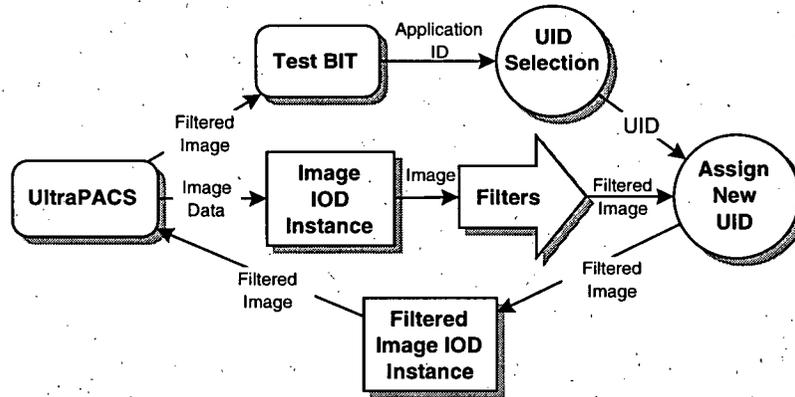


Figure 2.9: Model for Data Re-Organization

The Imager Lab also requires a means to re-organize data to fit the application-centred information model. Figure 2.9 shows the model we use to achieve this goal. Most Composite image Information Objects have an Attribute that uniquely identifies the study to which the image belongs. This Attribute is called the Study *Unique Identifier* (UID), as shown in Figure 2.4. The main idea in our model is to take each image after it has gone through the filters, and alter or add the Study UID to mark the image as belonging to the test data set for a certain BIT. This filtered image can then be sent back to the PACS. The primary benefit gained from this process is that it results in a ready-to-use data set in the database that can be found and retrieved by specifying the Study UID for the application that requires it. When more than one data set is required for an application, as is often the case, each data set can remain distinct from others with the same Study UID by using the Series UID, which is at one level below Study.

Another requirement of the BIT developer discussed in Chapter 1 is the standardization of methods to extract important image properties (*e.g.*, source modality, scan parameters, image dimensions and encoding methods). Use of the DICOM image format solves this problem. We have written programs to allow the user to view all Attributes within an image Information Object. This gives the BIT developer an easy way of obtaining information that is often hidden from the clinical user and would otherwise be time-consuming to find out.

Chapter 3

Implementation

This chapter discusses the implementation of our testbed, based on the model presented in Chapter 2. The level of DICOM conformance at each node is specified by listing the SOP Classes supported by applications at that site. The calling hierarchies of main libraries and modules are detailed, along with an appropriate functional description of each important sub-system. Finally, the applications themselves are discussed.

3.1 DICOM Service-Object Pair Classes

While the model diagramed in Figure 2.7 (Service Class SCU/SCP Roles) determines the direction of image data flow and location of storage, the exact DICOM capabilities and limitations of each Application Entity can only be specified by listing all of the SOP Classes and SCU/SCP roles that it supports. Determination of these classes was the first step in the implementation phase of this project. Note that although an SCU of a particular SOP Class is conceptually different from an SCP, the two roles are often implemented in the same application if both are required at

the same site.

SOP Classes for Radiology	
SOP Class Name	SCU/SCP Role
Verification SOP Class	SCU & SCP
CT Image Storage	SCU
MR Image Storage	SCU
Ultrasound Image Multi-Frame Image Storage	SCU
Ultrasound Image Storage	SCU

Table 3.1: SOP Classes for Radiology

Table 3.1 shows the SOP Classes supported by applications at the Vancouver Hospital Radiology Department. The Verification SOP Class is useful for testing connectivity, especially when new Application Entities are added across the network. Configuration parameters such as TCP port numbers can be checked using the Verification SOP Class. This site should have the ability to initiate and respond to C-ECHO requests, so it acts as both an SCU and an SCP for this SOP Class. The other SOP Classes are Storage Service SOP Classes, one for each imaging modality (*e.g.*, ultrasound) or variant thereof (*e.g.*, multi-frame ultrasound). Currently, there are three source modalities (CT, MRI, and ultrasound) on the network. This number is expected to grow as small local networks in the hospitals spread out to reach other modalities. In terms of the BIT development testbed, Radiology only needs SCU's for the Storage Service Class, because it does not receive images from other testbed sites. As explained in Section 2.3, Radiology is designed not to support Query/Retrieve Service Class SCP's for remote testbed SCU's in order to maintain the security of patient information. All of the SOP Classes in Radiology have been implemented in commercial software packages from various vendors. Those packages that have Query/Retrieve Service SOP Classes built-in are configured not to accept Associations from non-clinical sites.

SOP Classes for ALI PACS	
SOP Class Name	SCU/SCP Role
Verification SOP Class	SCU & SCP
Computed Radiography Image Storage	SCU & SCP
CT Image Storage	SCU & SCP
MR Image Storage	SCU & SCP
Ultrasound Image Multi-Frame Image Storage	SCU & SCP
Ultrasound Image Storage	SCU & SCP
Nuclear Medicine Image Storage	SCU & SCP
Secondary Capture Image Storage	SCU & SCP
Stand-Alone Overlay Storage	SCU & SCP
Stand-Alone Curve Storage	SCU & SCP
Stand-Alone Modality LUT Storage	SCU & SCP
Stand-Alone VOI LUT Storage	SCU & SCP
Patient Root Query/Retrieve Information Model - FIND	SCP
Patient Root Query/Retrieve Information Model - MOVE	SCP
Patient Root Query/Retrieve Information Model - GET	SCP
Study Root Query/Retrieve Information Model - FIND	SCP
Study Root Query/Retrieve Information Model - MOVE	SCP
Study Root Query/Retrieve Information Model - GET	SCP

Table 3.2: SOP Classes for ALI PACS

Table 3.2 lists the SOP Classes for the PACS, which acts as the central image server. As in Radiology, both SCU and SCP roles are supported for the Verification SOP Class. ALI's UltraPACS supports all 11 Storage Service SOP Classes in the current DICOM standard. Thus, it can receive and store images of any source modality on our testbed, plus any that may be added in the future. The PACS is an SCU in the Storage Service SOP Classes because it needs to send out images to satisfy retrieval requests. It is also an SCP in these classes because it needs to receive images for storage. There are also six Query/Retrieve Service SOP Classes (Patient Root and Study Root Query/Retrieve Information Models - FIND, MOVE and GET). The Patient Root Query/Retrieve Information Model means that search keys can be at any one of the four levels (Patient, Study, Series and Image), with the condition that at least one of the keys must be at the Patient level. The Study

Root Query/Retrieve Information Model means that the search keys must contain at least one at the Study level, with the rest at any level below Patient. An SCP in the Query/Retrieve classes handles query and retrieval requests from Imager Lab application entities. Note that most implementors, including the Imager Lab, use the MOVE service as the standard retrieval method; the GET service is supported here in case an AE that does not support MOVE is added to the network.

SOP Classes for Oral Biology	
SOP Class Name	SCU/SCP Role
Verification SOP Class	SCU & SCP
Computed Radiography Image Storage	SCU
CT Image Storage	SCU
MR Image Storage	SCU
Ultrasound Image Multi-Frame Image Storage	SCU
Ultrasound Image Storage	SCU
Nuclear Medicine Image Storage	SCU
Secondary Capture Image Storage	SCU
Stand-Alone Overlay Storage	SCU
Stand-Alone Curve Storage	SCU
Stand-Alone Modality LUT Storage	SCU
Stand-Alone VOI LUT Storage	SCU

Table 3.3: SOP Classes for Oral Biology

Table 3.3 shows the SOP Classes for Oral Biology. As in the other sites, there is an SCU and an SCP for the Verification SOP Class. All 11 Storage Service SOP Classes are supported in the SCU role. It is an SCU in these SOP Classes for sending images directly to the PACS. There are no Query/Retrieve Service SOP Classes.

The SOP Classes for the Imager Lab BIT development site are presented in Table 3.4. The Imager Lab supports all of the Storage SOP Classes in the PACS (SCU and SCP), as well as an SCU for each of the six Query/Retrieve SCP's provided by the PACS. This allows users to store, query and retrieve medical images

SOP Classes for BIT Development Lab	
SOP Class Name	SCU/SCP Role
Verification SOP Class	SCU & SCP
Computed Radiography Image Storage	SCU & SCP
CT Image Storage	SCU & SCP
MR Image Storage	SCU & SCP
Ultrasound Image Multi-Frame Image Storage	SCU & SCP
Ultrasound Image Storage	SCU & SCP
Nuclear Medicine Image Storage	SCU & SCP
Secondary Capture Image Storage	SCU & SCP
Stand-Alone Overlay Storage	SCU & SCP
Stand-Alone Curve Storage	SCU & SCP
Stand-Alone Modality LUT Storage	SCU & SCP
Stand-Alone VOI LUT Storage	SCU & SCP
Patient Root Query/Retrieve Information Model - FIND	SCU
Patient Root Query/Retrieve Information Model - MOVE	SCU
Patient Root Query/Retrieve Information Model - GET	SCU
Study Root Query/Retrieve Information Model - FIND	SCU
Study Root Query/Retrieve Information Model - MOVE	SCU
Study Root Query/Retrieve Information Model - GET	SCU

Table 3.4: SOP Classes for BIT Development Lab

of any modality in the database. The Patient Root Query/Retrieve Classes are used to retrieve the raw, unprocessed data because hospitals use a patient-centred information model, and send data sets to the PACS one patient at a time. The Study Root Query/Retrieve Classes are used to retrieve the filtered, ready-to-use images because these data sets have been re-organized to the application-centred information model.

The SOP Classes listed in the above tables unambiguously specify the DICOM data transfer capabilities of each node. For the exact expected behaviour of a particular SCU or SCP (*e.g.*, Association negotiation, error handling, etc.), the reader is referred to Part 4 of [Ame94]. However, these classes do not reveal what a site's capabilities are in terms of processing DICOM images. For example, in order to build an application that can integrate the retrieval and filtering processes based

on the model shown in Figure 2.8, routines must exist that can create and manipulate DICOM Composite image Information Objects and be able to read and set values of individual Attributes within an object. Although practically all DICOM applications manipulate Information Objects to some extent, the routines implemented in Storage and Query/Retrieve SCU/SCP's can only access a small subset of an image Information Object's Attributes. A complete set of access functions based on an object's IOD must be implemented to order to do Attribute-based processing to any large extent. This means although the Imager Lab has SCU/SCP's for all 11 Storage SOP Classes, this does not mean we have the ability to process all of these image types. In fact, currently we only have routines that can process the four image types from Radiology. Therefore, the routines developed for this testbed can be divided into two distinct but related groups: one to facilitate communication by implementation of the SOP Classes, the other for DICOM Composite image Information Object manipulation and processing. The following sections discuss how these routines were built.

3.2 Central Test Node Facilities (MIR Libraries)

All of our DICOM-related applications, with the exception of the software in Radiology, have been built on top of two sets of libraries, one called the Central Test Node facilities, from the Mallinckrodt Institute of Radiology (MIR), the other called the DICOM Support Suite, from ALI Technologies. The ALI libraries are higher on the calling structure, as shown in Figure 3.1.

The MIR libraries were developed for demonstration purposes at the 1993 Radiological Society of North America Annual Meeting. The application software was called the Central Test Node (CTN) because it provided a site that vendors

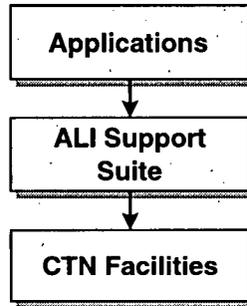


Figure 3.1: Calling Hierarchy of DICOM Applications

can use to test their applications for DICOM compliance via the Internet. The MIR subroutine libraries were called *facilities*. The libraries have since been regularly updated and released for public use, and are considered to be the *de facto* standard for an implementation of DICOM 3.0. There are four main CTN facilities: DCM, DUL, MSG and SRV. Figure 3.2 shows their calling structure.

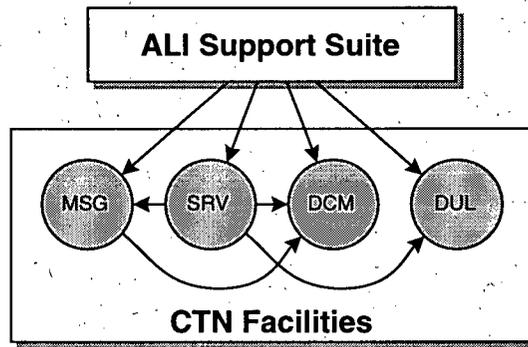


Figure 3.2: Calling Structure of CTN Facilities

The DCM facility provides the basic means for manipulation of DICOM Information Objects. There are functions for the addition and removal of Attributes, as well for the extraction and modification of Attribute values. These routines are

generic in nature, and rely on the calling functions to provide information specific to an object's IOD. For example, to change the value of the "Echo Time" Attribute in an MR image Information Object, the calling routine has to pass the identification, or *tag* value (0018,0081), the *value representation* (decimal string) and the length of the data element to the DCM function. The DCM facility also has functions for the import and export of DICOM byte streams using several transfer syntaxes.

The DUL facility is an implementation of the DICOM Upper Layer Protocol for TCP/IP (Figure 2.2), and uses Berkeley sockets for network access. The MSG facility provides functions for parsing and construction of structured command objects. The SRV facility implements DICOM message exchange, as defined in Part 7 of [Ame94], and includes functions for sending and receiving commands and data. The SRV facility uses the MSG facility to translate messages into DCM objects, and then calls DCM functions to export the objects into the DICOM byte stream. This byte stream is passed to the DUL facility and submitted to the network.

While an implementor can develop applications that directly call functions within the CTN facilities, this is usually not the best approach. The reason lies in the fact that the MIR libraries are low level, very complex and have a cumbersome functional interface.

3.3 ALI DICOM Support Suite

The ALI DICOM Support Suite library acts as simplified interface to the CTN facilities, and hides most of the details in the MIR implementation while providing all of the functionality that applications require. The Support Suite was originally implemented on the NeXTSTEP operating system (version 3.3), and was ported to Silicon Graphics (SGI) IRIX (version 6.2) for this project. There are 11 modules

in the Support Suite: diappent, dierr, dif, difutil, dilist, dinet, diobj, dircv, dirq, diservice and dps. Their calling structure is shown in Figure 3.3.

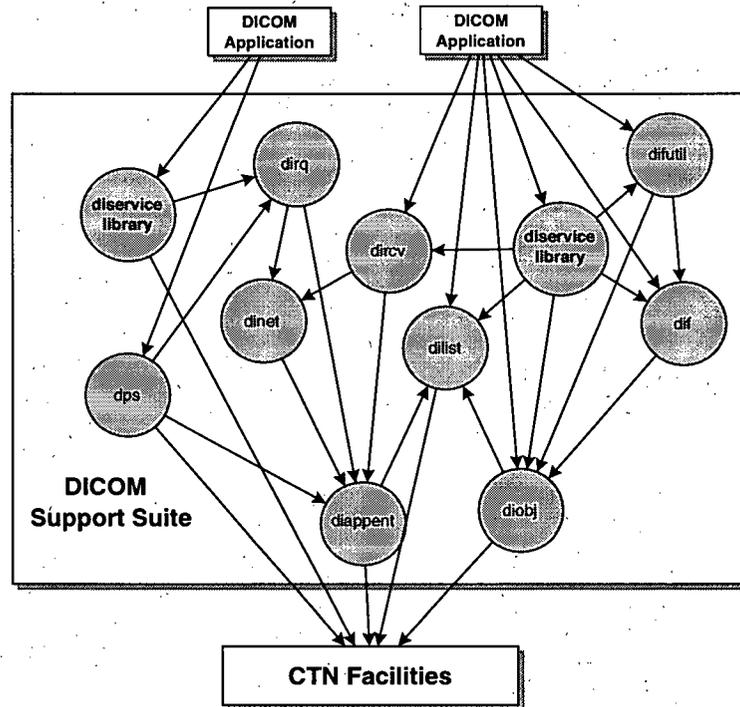


Figure 3.3: Calling Structure of ALI Support Suite

Note: the dierr module is called by all other modules, and was left undrawn as not to clutter the diagram.

The dinet module interfaces with CTN network communication routines, and provides support for network initialization and termination. The dirq module contains routines that sends out-going requests to form Associations. The dircv module is the complement of dirq, and handles in-coming Association requests. Diappent defines a structure for Application Entity objects, and provides easy access to commonly used configuration parameters (e.g., Application Entity Title, TCP port, etc.). This structure is required by almost all of the service-related modules to

identify the originator and recipient of DICOM messages. The dif module supplies routines that read image data from DICOM objects, and provides access to generic image information, such as samples per pixel, image dimensions, etc. Difutil contains routines for manipulating image data, such as scaling or colour to monochrome conversion. The dilist module is used for handling linked lists of data. Dierr defines all error codes for the Support Suite, and provides tracing functions for debugging and trouble-shooting purposes. The dps module performs all the operations needed for printing to DICOM-compliant film printers.

Diobj contains routines for handling DICOM objects, and includes functions for creating and manipulating objects both in memory and in files. This module provides the means to set and access every Attribute value in a DICOM ultrasound, CT or MR image Information Object. (The original ALI diobj module could only handle ultrasound objects; the CT and MR functions were written as extensions to the module by the author.) These access functions are based on the IOD of each object; a pair of "set" and "get" functions are provided for every Attribute in the IOD. This greatly simplifies Attribute-based processing, and facilitates implementation of our integrated Query/Retrieve/Filter application and software for re-organization of data. Because each IOD in the module has its own set of access functions, adding the processing capabilities for other image modalities is simply a matter of writing a new sub-module of "set" and "get" functions. The diobj module will grow as the testbed gets larger and more image types are required.

The diservice library is a group of sub-modules that are primarily responsible for providing the functionality in the Service Classes. This module is where the SOP Classes listed in Tables 3.2 to 3.4 were implemented. There are five sub-modules, each containing functions for a group of SOP Classes:

- **diecho**: allows the user to send and respond to C-ECHO requests as part of the Verification Service Class.
- **distoreSCU**: provides SCU functionality in the Storage Service Class, and allows users to send images to remote DICOM peers.
- **distoreSCP**: provides SCP support for the same SOP Classes as distoreSCU, and allows users to receive images from remote DICOM peers.
- **diquerySCU**: allows users to send Query/Retrieve Service Class requests. The C-FIND, C-MOVE and C-GET services are supported.
- **diquerySCP**: provides SCP support for the same SOP Classes as diquerySCU, and provides functionality to respond to Query/Retrieve Service Class requests.

3.4 Applications

This section describes the applications that have been built for the PACS, Imager Lab and Oral Biology nodes. All applications discussed in this section were developed collaboratively by ALI Technologies and the Imager Lab.

UltraPACS is the name of the image storage and retrieval software installed in the PACS. It is version 3.0 of the commercial release from ALI, and runs on NeXTSTEP 3.3. The main customization for our testbed is that the clinical user-interface has been removed from the system, leaving only the PACS backend. UltraPACS was primarily built using the diservice library.

Ditest is a multi-function, menu-driven application, and is based on an ALI test application originally developed on NeXTSTEP. Ditest now runs on SGI IRIX 6.2, and has the following functionality:

- C-ECHO send and receive: this program allows a user to send C-ECHO's to peer AE's, and gives a report to indicate success or failure. This program also responds to C-ECHO's in a DICOM-compliant manner.
- Composite image Information Object creation: ditest can create CT, MR and ultrasound image Information Objects from raw image data or an existing DICOM image file, and allows users to specify values for certain Attributes when doing so. This is important for converting non-DICOM medical images to the DICOM format. Ditest gives the user a way to package a raw image with all of its important properties into a single file.
- Send images: as an SCU of the Storage Service Class, ditest can send image Information Objects to any SCP supporting the same SOP Classes (in this case, the UltraPACS).
- Receive images: ditest is also an SCP of the Storage Service Class, and can receive image Information Objects. This functionality is used primarily for testing purposes.
- Dump Attributes: ditest can take any Composite image Information Object, and output all of its Attribute values. This is a very useful way for the BIT researcher to find out properties of an image.

Ditest is installed in Oral Biology and the Imager Lab to satisfy the requirements for a Verification SOP Class SCU/SCP, as well as an SCU for each of the various Storage Service SOP Classes.

Query_test is a text-based application that integrates the querying, retrieval and filtering processes. It allows the user to query the PACS by sending one or more search keys with either Patient or Study as the root level. For example, a query can consist three keys: "Patient Last Name = Burger", "Study UID = 123.456.789" and

"Series UID = 987.654.321". The user gets a list of matching data sets in return. He can then select a data set, and send a MOVE request to the PACS to retrieve the data. Query_test acts as a Storage SCP to receive the data. As the data set arrives, the user can choose a number of filter options from the menu. The option selected either specifies a particular processing filter, or allows the application to choose among a group of filters. Query_test reads various Attributes, and selects filters based on those values. Most of the filters in use are image processing filters, such as those for scaling image dimensions, noise removal or feature detection. The application then applies the filters to each image, resulting in a data set that is suitable for input into a particular BIT.

Dcm_change is a command-line application that allows the user to add or change the value of any Attribute in a CT, MR or ultrasound Information Object. This allows the user to re-organize data by changing or adding the Study and (optionally) Series UID's. Shell scripts are usually used to re-assign UID's to large sets of data.

Chapter 4

Performance and Results

This chapter presents an evaluation of how well our model and implementation meet the goals outlined in Chapter 1. An analysis of how various factors affect data transfer speed is given, along with experimental results. In addition, the experience of using the testbed to develop a volume rendering application is used to identify the strengths and deficiencies in our model.

4.1 ATM Network Performance

Networking technology plays an important role in the efficiency of data transfers in our testbed, largely because of the number and size of image files that are moved. For example, an average abdominal CT scan produces 128 cross-sectional slices of dimensions 512×512 . CT images normally have 16 bits per pixel, so the entire data set would be about $128 \times 512 \times 512 \times 2 \approx 67$ Mbytes. Furthermore, the testbed needs to support multiple data transfers and/or real-time applications simultaneously over each link. To increase the available bandwidth, we use fibre-optic based Asynchronous Transfer Mode (ATM) technology, instead of standard 10 megabits

per second (Mbps) Ethernet. Most of the links on our testbed are 100 Mbps ATM links, but these will be upgraded to 155 Mbps before the testbed becomes fully operational. An analysis of factors affecting ATM performance has been performed as part of testbed development, but because the focus of this thesis is at the DICOM level and above, we only present a brief summary of the results here, primarily to put the results of the next section (performance analysis of the DICOM implementation) into clearer perspective. For more detail on ATM performance (*e.g.*, experiment setup, measurement results for each factor, etc), the reader is referred to [TV96].

There are a number of potential bottlenecks that could reduce the effective bandwidth in an ATM network. Among the ones identified to have the strongest effects on our testbed are:

- Transport Overhead - The various transports associated with ATM, such as TAXI (Transparent Asynchronous Transmitter-Receiver Interface), AAL (ATM Adaptation Layer), IP (Internet Protocol) and TCP (Transmission Control Protocol), all require overhead and take up some of the available bandwidth. For example, at the ATM layer itself, five of the 53 bytes per cell are used for header information.
- Buffering Techniques - Changing the send and receive buffer sizes can have a strong effect on the throughput. For example, the sliding-window mechanism of TCP has a significant effect on TCP performance.
- Adapter/Switch Differences - The switches and interface adapter cards used in a network have a definite effect on ATM performance. One important difference between different adaptors is the amount of work being done in hardware versus software. Another important factor is reliability; a slower

adaptor may outperform an adaptor that is rated faster but tends to lose more packets.

- Device Drivers - The efficiency of the device drivers used is another significant factor. A stripped-down, bare-bones driver will be faster than the same driver with added support for statistics generation, flow control, signalling and network management. Of course, how well the particular implementation is optimized is important as well.
- Workstation Performance - The baseline performance of the workstation is sometimes an overlooked factor. However, the relative performance of any given system is key to determining the overall performance of the adapter.
- Network Topology - Many benchmark studies test products on single-connection systems; this does not reflect how ATM is likely to be used on our testbed. The performance of adaptor cards and switches in handling multiple simultaneous connections can have a strong effect on overall efficiency.

Experiments to measure the effects of most of the above factors were performed using three public domain programs: *ttcp*, *netperf* and *netspec*. We optimized the controllable factors as much as possible in order to obtain a practical maximum throughput that we can compare with DICOM analysis results. Using a TCP connection over a 100 Mbps link between the two fastest workstations on the testbed (SGI Indigo II Extreme and SGI Indigo II Impact, both in the Imager Lab), we obtained maximum memory-to-memory transfer speeds of approximately 70 Mbps. From the Sun Sparc 10 in Radiology to the Indigo II Extreme in the Imager Lab, we measured a maximum throughput of approximately 60 Mbps.

4.2 DICOM Performance

The use of DICOM greatly increases the functionality for image data transfer, but significantly reduces the effective throughput. To measure the overhead required by the MIR implementation of DICOM and to identify which CTN facilities consume the most bandwidth, we placed software probes within parts of the libraries. Two simple CTN test programs, *send_image* and *simple_storage* were used to transfer images between workstations. These programs call the CTN facilities directly, and do not make use of the ALI libraries. For our experiments, the 100 Mbps ATM link between the two SGI Indigo II's mentioned above was used to transfer test images of various sizes. Table 4.1 shows the measurement results for a file size of 525350 bytes and a number of protocol data unit (PDU) sizes. For each PDU size, the measured throughput and the percentage of total time (exclusive of file reading and writing) spent in each group of facilities are given.

PDU (Bytes)	Throughput (Mbps)	Percentage of Time			
		SRV	MSG & DCM	DUL	Sockets/TCP/IP
16384	18.7	2.7%	4.4%	46.3%	46.5%
32768	24.2	3.0%	4.7%	26.0%	66.3%
65536	35.7	4.0%	6.7%	20.0%	69.4%
131072	37.7	3.8%	6.8%	12.0%	77.5%
262144	43.1	4.2%	7.6%	9.1%	79.1%
524288	50.6	4.8%	8.6%	8.0%	78.4%
1048576	55.2	6.0 %	9.7 %	6.3 %	78.0%
2097152	54.8	6.0 %	9.7 %	6.4 %	77.9%

Table 4.1: CTN Facilities Performance Statistics

The table shows that as the PDU size nears that of the file size, the throughput increases. The throughput stops increasing and levels off when the PDU size reaches the file size. The percentage of time spent in the transport layers (sockets/TCP/IP) also increases with PDU size, implying greater efficiency within the

CTN facilities. The percentage of time spent in the DUL facility decreases dramatically with increasing PDU size, while the other facilities show relatively slight increases. This can be explained by the fact that the DUL facility is responsible for breaking up the byte stream into appropriately sized units and translating them into the correct transfer syntax. For small PDU's, this requires a large number of function calls and memory copies. For example, using a PDU size of 16384 bytes to send a file of size 525350 bytes requires 32 more function calls than using a PDU size of 1048576 bytes. Thus, devices for which the PDU size is changeable should be configured to use the largest PDU size available, unless restricted by the lack of memory resources. The maximum throughput obtained was approximately 55 Mbps for all image sizes tested, which is 15 Mbps lower than the maximum measured for file transfers without using DICOM. This clearly shows that MIR DICOM implementation requires a large amount of overhead.

4.3 Application Performance

The most significant performance measurements are at the application level, because they represent a practical appraisal of the level of responsiveness that the user can expect. In these experiments, additional factors such as the overhead required by the ALI Support Suite, application efficiency and disk speed all can potentially reduce the speed at which users can send and receive images.

In our experiments, we used the SCP functionality in dtest in order to simulate transfers to and from the PACS. The reason why we did not use the UltraPACS itself for testing is that the Pentium PC on which UltraPACS is installed currently only has a 10 Mbps Ethernet connection to the network. Upgrading the link to 100 Mbps has been delayed because of the lack of ATM and Fast Ethernet device

drivers for NeXTSTEP 3.3. ALI is scheduled to release a Windows-NT port of UltraPACS in late 1997; our system will be upgraded at that time. Given that the Storage Class modules in ditest are the same as the ones in UltraPACS, our simulation results should have a reasonable degree of accuracy, at least for evaluation of the software implementation. The assumption, which will be verified when the system is upgrade, is that changing the operating system and upgrading the network hardware will not degrade performance to a level much below what we have observed.

To simulate image transfers from Radiology to the PACS, we used Radiology's GE Advantage Workstation to send data sets to ditest in the Imager Lab. We inserted software probes into ditest to distinguish between the amount of time spent by the applications in setting up the connection and the amount of time for actual image transfer. We measured the entire time taken for the images to be moved across and saved onto a locally (*i.e.*, non-nfs) mounted disk. The two devices take an average of 0.5 to 1.0 seconds to negotiate an Association and initiate the transfer process. The actual data transfer follows, and occurs at approximately 17 Mbps. Finally, Association release takes another 0.5 to 1.0 seconds. The relatively slow data rate can be explained by the fact that the clinical software in Radiology is internally configured to use a maximum PDU size of 16384 bytes. Changing the PDU size will require further investigation by GE engineers.

To simulate transfers between the Imager Lab and the PACS, we performed a number of ditest-to-ditest experiments between ATM-connected Indigo II workstations. Again, Association negotiation and release each takes 0.5 to 1 seconds. Data throughput results are very similar to those done for analysis of performance of the CTN facilities. Table 4.2 shows the test results for an image size of 525350.

PDU	Throughput
16384	18.0
32768	22.3
65536	34.8
131072	37.0
262144	41.3
524288	48.3
1048576	52.2

Table 4.2: Imager-to-PACS Simulation Results for Image Size 525350

In comparison with Table 4.1, we can see that the throughput numbers are just slightly below that for the CTN facilities. This means that the ALI Support Suite functions do not add a great deal of processing overhead during data transfers. These results also show that reading and writing to a local disk (at least on the Indigo II systems) do not negatively impact transfer speed. These results are also representative for image transfers from Oral Biology to the PACS, because Oral Biology also uses an Indigo II system connected to the ATM network.

To test the querying performance of the PACS, we used UltraPACS itself, because querying without retrieval requires very little network bandwidth, and is not affected by the slow Ethernet link. On average, the entire process from submission of the request using `query_test` (with a patient name or Study UID as the search key) to display of the matching list takes one to three seconds. However, this time will likely increase as the size of the database grows. Currently, there are only about 45 data sets stored in the PACS. The system is designed and expected to handle thousands of patient cases.

4.4 Evaluation of Model

To evaluate the model that we used, we examine the experience of developing a volume rendering application on our testbed. A volume renderer takes a data set composed of a number cross-sectional slices of one or more objects, such as those from a medical scanner, and reconstructs the volume to form a three-dimensional (3D) representation of the object. Figure 4.1 shows the scanning and reconstruction process. We started with a volume renderer from SGI called *Volren*, and used the testbed to add and alter features to customize it for our purposes.

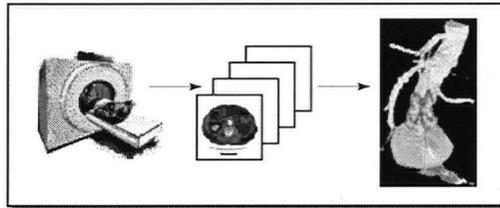


Figure 4.1: Scanning and 3D Reconstruction

In order for the volume renderer to depict the target objects clearly and without occlusion from other objects, the two-dimensional (2D) images must go through a process called *segmentation*. Segmentation involves processing the raw images to enable or enhance the visualization of structures that are useful for analysis, while suppressing or eliminating the structures that would otherwise obstruct our view of the areas of interest. Figure 4.2 shows an example of the segmentation of a 2D CT image. In this case, the objects we want to see are the aneurysm, the aorta and the tyres. The figure shows how segmentation greatly enhances the clarity of the useful tissues.

The segmentation process is ideal for integration into the query/retrieval

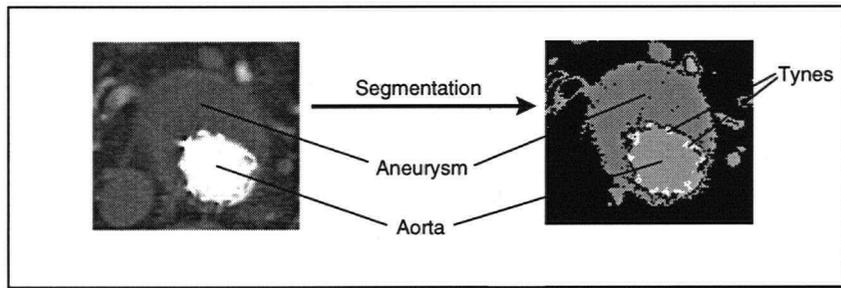


Figure 4.2: Segmentation of a 2D CT Slice

process, because every data set must be segmented before being fed into Volren. In addition, different segmentation filters are often used for one data set to get different rendered effects, and integration allows different filters to be easily applied to any data set.

After segmentation, additional pre-processing of the 2D images is required before the volume can be rendered. Volren only takes images that have dimensions that are powers of two, so a cropping and/or rescaling filter is sometimes required. Also, Volren requires images to be in either the TIFF or RGB formats. This is handled by format conversion filters.

After a patient has been scanned in Radiology, clinical users send the images to the PACS. The BIT researcher in the Imager Lab creates a query using `query_test`, with the patient's name or scan date as the search key. The query is submitted to the PACS, which returns a list of matching cases to `query_test`. The BIT researcher then submits a retrieval request for the correct data set. Once the images arrive into a temporary location on the local disk, `query_test` reads the image Attributes, and decides whether to apply a cropping/scaling filter. Application of the filter is done automatically. Meanwhile, the user can use the Attribute values to decide which

segmentation filter is most appropriate for this data set. Query_test then applies the selected segmentation filter, followed by a format conversion filter that results in a set of images suitable for use by Volren. Because Volren requires many command-line parameters, an automatic script generation program was also integrated into query_test to create the appropriate executable scripts during the filtering process. A function has been built into the interface of query_test to execute the created scripts, so that the user does not have to leave query_test to test Volren.

One deficiency found in our model is that it assumes that images in the final filtered form can be sent back to the PACS for storage. This assumption is derived from the idealistic notion that all of our medical imaging applications should take DICOM images as input. In this case, because the filtered images are in TIFF or RGB format, and not in DICOM format, they cannot be sent to the PACS. This problem is solved by configuring query_test not to erase some of the temporary files created during each stage of the filtering process. For each filtered data set, there is one set of temporary images that have been cropped/scaled and segmented, but not converted to TIFF or RGB. These images are in their raw format (*i.e.*, same as DICOM, but without the header), and can be easily re-combined with their headers to form DICOM images. These can be sent to the PACS, but they have to be run through the image conversion filter each time they are retrieved. A more elegant solution would probably be to incorporate some image conversion routines into the PACS, so that it can export the image data in various formats while outputting DICOM header information as well.

Before being sent to the PACS, each filtered image has its Study UID re-assigned to one that is unique to Volren test data. Because there is more than one data set (currently seven) used for Volren, each set is given a Series UID that is

unique among Volren data sets. There is a question of how to generate Study UID's such that they do not conflict with those generated by other devices on the testbed. This is important, because the unfiltered data sets retain their original Study UID's. The DICOM standard specifies how commercial devices should create UID's using properties such as the device type (*e.g.*, MR, CT, etc.), serial number of the device and data type. We can use a device type number that is not among the ones in the standard. By using an "imaginary" device, we can be sure not to conflict with other devices. A Series UID can be derived from the Study UID by adding a suffix (*e.g.*, ".1" for the first data set for this BIT, ".2" for the second one, etc.).

A problem with the model that is related to data re-organization is that there is no efficient means by which the original, unfiltered data set can be re-organized with new UID's. This is a result of the fact that the DICOM standard does not contain services for changing Attributes of Information Objects once they have been stored by a Storage Class SCP. The user currently has the option of retrieving a data set without applying any filters to it, assigning new Study UID's to the images, then sending them back to the PACS. While this method allows the re-organized data to be retrieved using the new Study UID, the PACS would have to store two sets of practically identical data, because it considers the new Study UID to indicate a new data set. ALI is currently investigating ways of handling this problem in UltraPACS.

Besides the problems described above, the system provides efficient and organized storage for the BIT developer. Radiology staff can send images to the Imager Lab with very little interruption to clinical operation. Patient data is organized, and safely stored and protected from unauthorized access. The integrated filtering and retrieval process automates what would otherwise be time-consuming tasks. Sites

such as Oral Biology can easily send DICOM images directly to the PACS.

Chapter 5

Related Work

This chapter describes work aimed at solving problems similar to those we outlined in Chapter 1. A number of related projects are compared and contrasted with our testbed. Most of the papers discussed are in the area of medical informatics, more specifically health care information infrastructure and standards.

Many testbeds are being developed with the same general goal as ours; that is, the sharing of resources by a number of geographically distributed sites. Many such testbeds demonstrate case-specific clinical uses of PACS's on networks, such as [LWH+96]. Others are for sharing information with non-clinical sites. [GD95] is building a software engineering environment that is distributed among hospitals and software development sites, using three of the primary objectives that we have adopted: horizontal integration, vertical integration and adaptability. This environment is used for developing health care information systems. Developers in [TBK+96] are using an ATM network and DICOM to share computing power and an MR image database between a technical university and two university hospitals. As in our testbed, security issues associated with having a remotely located patient

image database are an important concern.

Work such as [MYGT96] and [MYWY96] use process modelling to select and estimate the benefits of information systems such as a PACS. They create process models that simulate the activities and communications within the radiology department, which is similar to how we model the processes in and between sites on our testbed. However, [MYGT96] and [MYWY96] go to greater depths in deriving metrics for cost, efficiency and quality improvements, and also have a mostly clinical focus. Other work, such as [BSS95], take a similar approach to ours in that they relate key requirements to DICOM processes to implement a DICOM network. [PCZK94] links processes with specific SOP requirements and SCU/SCP roles, provides an implementation model and discusses some implementation issues. [PCZK94] is a "proof of concept" paper for DICOM, and provides a good basic model for building any DICOM network. For example, it demonstrates how the developer should start by creating an application profile for each site, then determine the Service Classes required. We used the same basic procedures to build our testbed.

A number of papers focus on the design and/or evaluation of the underlying network. Papers such as [BK94] and [MLS95] concentrate on the design of the network, and discuss issues such as network architecture and traffic modelling. [WHL⁺96] performs a number of throughput experiments in transferring radiologic images over an ATM network, but, unlike our work, does not try to analyze factors affecting transmission rate. [DHV⁺96] compares transfer performance over an ATM network with and without using DICOM, which is similar to what was done in our experiments. But they also assess the effect of network capabilities on productivity, in this case for clinical practice.

A performance analysis of DICOM over TCP is presented in [MPB⁺94], as part of a comparison with DICOM over OSI protocols. The basic experimental procedures in this paper are similar to what we used to obtain our DICOM performance results. For example, the placement of software probes within the CTN facilities and the use of different PDU sizes are done in both cases. However, our analysis goes into greater depth, and discusses reasons why DICOM affects performance so strongly.

One main difference between our use of the PACS and the typical clinical usage is that most hospitals store only one image modality on a PACS. This is because the image processing and/or display software connected to a clinical PACS typically only handles one modality. The DICOM processing software in the Imager Lab currently handles CT, MR and ultrasound images, so the PACS currently stores these modalities. In the future, the PACS, facilitated by the addition of sub-modules in the diobj module of the ALI Support Suite, will be able to store any and all modalities added to the testbed. [FHS95] discusses some of the issues involved in implementation of a multi-modality DICOM server.

A number of papers deal with the integration of PACS [WHH96] or DICOM [WCC95] with image processing tools. [WHH96] investigates the integration of the storage and communication components of PACS with 3D visualization tools. This is similar to our integration of the query/retrieval process with the image filtering and volume rendering processes. [WCC95] presents a DICOM image display and processing tool for teleradiology, and is an example of Attribute-based image processing. This tool is an end-user application, unlike our filters, which are primarily intermediate processing tools.

Chapter 6

Conclusions and Future Work

6.1 Summary and Conclusions

In this thesis, we presented a testbed for the efficient transfer and storage of medical image data to support the distributed development of biomedical imaging tools. We used a client-server model of distributed software development and the specific requirements of each site to derive an implementation model. DICOM 3.0 is the image transfer protocol and image format that we use to facilitate horizontal and vertical integration at each site. It is a complex standard, but the major concepts in DICOM fit very well into a client-server model, and we had little trouble linking testbed requirements to specific DICOM facilities, most important of which are the Service-Object Pair Classes. DICOM is found to satisfy most of the site-specific requirements, and provides a high-level of interoperability between heterogeneous computing environments. A number of deficiencies were discovered in our model, specifically in the use of the DICOM information model hierarchy to re-organize data; but they are not expected to cause serious problems in the use of the testbed.

A series of performance experiments were done to check the efficiency of data transfers. It was discovered that the MIR implementation of DICOM requires a great deal of overhead, and reduces data transmission rates substantially. The ALI DICOM Support Suite greatly simplifies implementation of DICOM applications, while adding little transport overhead. The modularity of the Support Suite allows for the relatively easy addition of processing capabilities for new modalities on the testbed. Overall, the initial implementation of the testbed was successful in that practically all of the goals outlined at the beginning of this thesis were achieved.

6.2 Future Work

There are still a number of improvements that can be made to our testbed. The greatest urgency is to upgrade the PACS so that it is connected to the network via a high-speed link. This will allow more proper performance experiments to be done. All of our simulations using ditest as a replacement should be re-done using the real PACS. Also, the querying speed of the PACS should be checked as the size of the database grows.

UltraPACS can be improved in several ways. It should contain a facility that allows the user to change Attributes of stored image Information Objects. It should have the capability of exporting image files in different formats, while retaining DICOM header information. Real-time data compression should be supported.

Data throughput can be increased in a number of ways. The performance of the MIR implementation of DICOM can potentially be greatly improved. The PDU size used by the clinical workstations should be re-configured to be larger. So far, the default implementations of TCP/IP has been used on all systems. The use of improved versions of TCP, such as TCP.Vegas, can potentially increase the available

bandwidth.

More imaging processing tools should be built and integrated into the query/retrieve application. A number of these could be end-user applications such as a DICOM image display tool. Such applications will allow the user to visually analyze the results of image processing.

Finally, the distribution of data types other than images will be further investigated. A number of projects have already begun that aim to facilitate remote consultation to distribute domain-specific knowledge. Development of an environment to share applications or software components is also a future goal.

Bibliography

- [Ame94] American College of Radiology and National Electrical Manufacturers Association. *Digital Imaging and Communications in Medicine (DICOM): Version 3.0*. ACR-NEMA Committee, Working Group VI, Washington, DC, 1994.
- [BK94] S. Banerjee and M. Kabuka. Atm-based fiber optic network for scalable and modular pacs design. In *Proceedings of the SPIE Conference on Medical Imaging 1994: PACS Design and Evaluation*, pages 218–227, Newport Beach, CA, 1994.
- [BSS95] S. M. Beaver and T. M. Sippel-Schmidt. Approach to implementing a dicom network: incorporating both economics and workflow adaptation. In *Proceedings of the SPIE Conference on Medical Imaging 1995: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 124–131, San Diego, CA, 1995.
- [CG96] P. Cahoon and E. Grant. Telemedicine and shared multidimensional workspaces. *Computer Graphics*, 30, February 1996.
- [DHV+96] A. Duerinckx, A. Hayrapetian, D. Valentino, E. Grant, D. Rahbar, M. Kiszonas, G. Franco, R. Shimabuku, G. Hagan, M. Melany, S. Narin, and N. Ragavendra. Assessment of asynchronous transfer mode (atm) networks for regional teleradiology. In *Proceedings of the SPIE Conference on Medical Imaging 1996: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 61–70, Newport Beach, CA, 1996.
- [FHS95] M. Frost, J. Honeyman, and E. Staab. Dicom server for ct and mr acquisition. In *Proceedings of the SPIE Conference on Medical Imaging 1995: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 92–98, San Diego, CA, 1995.

- [GD95] R. Greenes and S. Deibel. Collaborative health care information system development through sharable infrastructure, services, and paradigms. In *Proceedings of the Eighth World Congress on Medical Informatics*, pages 190–194, Vancouver, BC, 1995.
- [KC95] M. Ko and P. Cahoon. A shared 4-d workspace. *University of British Columbia Department of Computer Technical Report*, 95(19), August 1995.
- [LWH⁺96] J. Lee, A. Wong, H. K. B. Huang, T. Bazzill, J. Zhang, and K. Andriole. Atm-distributed pacs server for icu application. In *Proceedings of the SPIE Conference on Medical Imaging 1996: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 14–21, Newport Beach, CA, 1996.
- [MLS95] M. Meissner, B. Levine, and Mun Seong. Building an integrated clinical and research network. In *Proceedings of the SPIE Conference 1995: Health Care Information Infrastructure*, pages 92–99, Philadelphia, PA, 1995.
- [MPB⁺94] S. Moore, F. Prior, G. J. Blaine, D. Channin, and J. Kilday. Measured performance of dicom-tcp and dicom-iso implementations. In *Proceedings of the SPIE Conference on Medical Imaging 1994: PACS Design and Evaluation*, pages 359–367, Newport Beach, CA, 1994.
- [MYGT96] S. Mills, R. Yeh, B. Girorir, and M. Tanik. Process-driven selection of information systems for healthcare. In *Proceedings of the SPIE Conference on Medical Imaging 1995: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 347–354, San Diego, CA, 1996.
- [MYWY96] S. Mills, S. Yeh, M. Wasniewski, and R. Yeh. Estimating the benefits of clinical pacs through process modelling. In *Proceedings of the SPIE Conference on Medical Imaging 1996: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 253–264, Newport Beach, CA, 1996.
- [PCZK94] F. Prior, D. Channin, M. Zaidel, and J. Kilday. Dicom user conformance profile: Proof of concept. In *Proceedings of the SPIE Conference on Medical Imaging 1994: PACS Design and Evaluation*, pages 348–358, Newport Beach, CA, 1994.

- [TBK⁺96] A. Thiel, J. Bernarding, M. Krauss, S. Schulz, and T. Tolxdorff. Distributed medical services within the atm-based berlin regional testbed. In *Proceedings of the SPIE Conference on Medical Imaging 1996: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 32-43, Newport Beach, CA, 1996.
- [TV96] Roger C. Tam and Son Vuong. Practical considerations on atm performance. In *18th Biennial Symposium on Communications*, pages 283-286, Kingston, Ontario, 1996.
- [WCC95] C. Wang, W. Chimiak, and H. Craig. Dicom 3.0 image display and processing tool for teleradiology. In *Proceedings of the SPIE Conference on Medical Imaging 1995: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 498-501, San Diego, CA, 1995.
- [WHH96] S. Wong, H. S. Hoo, and H. K. B. Huang. Empower pacs with content-based queries and 3d image visualization. In *Proceedings of the SPIE Conference on Medical Imaging 1996: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 569-579, Newport Beach, CA, 1996.
- [WHL⁺96] A. Wong, H. K. B. Huang, J. Lee, T. Bazzill, and X. Zhu. High-performance image communication network with asynchronous transfer mode technology. In *Proceedings of the SPIE Conference on Medical Imaging 1996: PACS Design and Evaluation: Engineering and Clinical Issues*, pages 44-52, Newport Beach, CA, 1996.