A DEVELOPMENT OF SOFTWARE FOR DICOM COMMUNICATION

PAKAWAT NILAPONG

A THESIS SUMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF ENGINEERING
(BIOMEDICAL ENGINEERING)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNVERSITY
2001

ISBN 974-665-472-1

Thesis
entitled

A DEVELOPMENT OF SOFTWARE FOR DICOM COMMUNICATION

....
Mr. Pakawat Nilapong
Candidate

....
Mr.Trongtum Tongdee, M.D.
Major-advisor

....
Mr.Chatchai Neatpisarnvanit, Ph.D.
Co-advisor

....
Prof.Liangchai Limlomwongse, Ph.D.
Dean
Faculty of Graduate Studies

....
Asst.Prof.Pirojana Suvanasuthi, M.Eng.
Chairman
Master of Engineering Programme in Biomedical Engineering
Faculty of Engineering

Thesis
entitled

A DEVELOPMENT OF SOFTWARE FOR DICOM COMMUNICATION

Was submitted to the Faculty of Graduate Studies, Mahidol University for the degree of Master of Engineering (Biomedical Engineering)

on
March 20,2001

....
Mr.Pakawat Nilapong
Candidate

....
Mr.Trongtum Tongdee, M.D.
Chairman

....
Mr.Chatchai Neatpisarnvanit, Ph.D.
Member

....
Asst.Prof.Werasak Kurutach, Ph.D.
Member

....
Prof.Liangchai Limlomwongse, Ph.D.
Dean
Faculty of Graduate Studies
Mahidol University

....
Mr.Thanakorn Uan-on, D.Engr.
Dean
Faculty of Engineering
Mahidol University

ACKNOWLEDGEMENT

I'm very grateful to Trongtum Tongdee my principal supervisor for his guidance, precious advice and encouragement thoughout. He was always nice and kind when he suggested anything to me. I also would like to thank Chatchai Neatpisarnvanit my co-advisor for his encouragement and constructive comments in my English writing.

I wish to think my friends Biomed'1 and all staff of Biomedical Engineering program, faculty of Engineering, the Mahidol University.

Pakawat Nilapong

4137955 EGBE/M : MAJOR: BIOMEDICAL ENGINEERING; M.Eng. (BIOMEDICAL ENGINEERING)

KEY WORD : DICOM COMMUNICATION

PAKAWAT NILAPONG : A DEVELOPMENT OF SOFTWARE FOR DICOM COMMUNICATION. THESIS ADIVISORS : TRONGTUM TONGDEE M.D., CHATCHAI NEATPISARNVANIT Ph.D 86 p. ISBN974-665-472-1

This research concerns the development of software for DICOM communication. This software employed the object-oriented language C++ and LEAD TOOLS software. Also Microsoft Access 97 was applied for managing data supported by the developed software. The software was designed to work on a personal computer with the Microsoft Window 95/98 operating system. A physical connection to a TCP/IP network is necessary to interact with remote DICOM compliant devices.

The goal is to develop non-commercial DICOM software, which has the same function as the commercial DICOM software, allowing the developed software to access and retrieve DICOM files directly on a personal computer in the standard DICOM 3.0 file format.

The implementation result is that the developed software can display, store, query and retrieve data from the DICOM server. These capabilities are the same as the license DICOM software.

In the future version, the software should be implemented to support both the Service Class Provider (SCP) and the Service Class User (SCU). Finally, image management should be included in order to make more efficient software.

4137955 EGBE/M : สาขาวิชา : วิศวกรรมชีวการแพทย์ วศ.ม. (วิศวกรรมชีวการแพทย์)

ภควัต นิลพงษ์ : การพัฒนาโปรแกรมเพื่อการติดต่อสื่อสารระหว่างไฟล์ DICOM (A DEVELOPMENT OF SOFTWARE FOR DICOM COMMUNICATION) คณะกรรมการควบคุมวิทยานิพนธ์ : ตรงธรรม ทองดี M.D., ฉัตรชัย เนตรพิศาลวนิช 86 p. ISBN 974-665-472-1

งานวิจัยชิ้นนี้เกี่ยวข้องกับการพัฒนาโปรแกรมเพื่อการติดต่อสื่อสารระหว่างแบบ DICOM การพัฒนาจะใช้ Microsoft Visual C++ และตัวอย่างจากโปรแกรม LEADTOOLS เป็นส่วนหลักของโปรแกรม และใช้ Microsoft Access 97 ในการจัดการข้อมูลเพื่อสนับสนุนการพัฒนาโปรแกรม โปรแกรมนี้จะทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้ระบบปฏิบัติการ Microsoft Window 95 หรือ 98 และมีความสามารถในการเชื่อมต่อระบบเน็ตเวิร์คโดยใช้มาตราฐาน การส่งข้อมูลแบบ TCP/IP กับอุปกรณ์อื่นที่ทำงานในมาตราฐาน DICOM

วัตถุประสงค์ของงานวิจัยนี้คือ การพัฒนาโปรแกรม DICOM ที่ไม่มีลิขสิทธิ์เพื่อที่จะสามารถพัฒนาโปรแกรมนี้ต่อไปได้ และอีกทั้งยังต้องมีความสามารถเทียบเท่ากับโปรแกรมที่มีลิขสิทธิ์ นั้นก็คือ สามารถที่จะเข้าไปจัดการข้อมูลแบบต่างๆบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่สนับสนุนมาตราฐาน DICOM Version 3.0

ผลจากการทดสอบพบว่าโปรแกรมที่พัฒนาขึ้นสามารถเรียกดูข้อมูล แสดงผลข้อมูล จัดเก็บข้อมูล และค้นหาข้อมูล ซึ่งเหมือนกับโปรแกรม DICOM ที่มีลิขสิทธิ์

ในการพัฒนาโปรแกรมขั้นต่อไป โปรแกรมควรที่จะมีความสามารถในการกระทำตัวเองได้ทั้ง "SCU (Service Class User)" และ "SCP (Service Class Provider)" และสุดท้ายควรจะมีการรวมความสามารถในการจัดการเกี่ยวกับรูปภาพ เอาไว้เพื่อให้โปรแกรมมีประสิทธิภาพดียิ่งขึ้น

CONTENTS

CONTENTS (Cont.)

CONTENTS (Cont.)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Cont.)

NOMENCLATURE

DICOM	=	Digital Imaging and Communication in Medicine
PACS	=	Picture Archiving and Communication System
SCU	=	Service Class User
SCP	=	Service Class Provider
SOP	=	Service Object Pair
IOD	=	Information Object Definition
DIMSE	=	DICOM Message Service Element
DIMSE-C	=	DICOM Message Service Element - Composite
DIMSE-N	=	DICOM Message Service Element – Normalized
ACR	=	American College of Radiology
NEMA	=	National Electrical Manufacturer Association
RSNA	=	Radiological Society of North America
TCP/IP	=	Transmission Control Protocol/Internet Protocol
OSI	=	Open System Interconnection
NIU	=	Network Interface Unit
UID	=	Unique identifier
PDU	=	Protocol Data Unit
DUL	=	DICOM Upper Layer
AE	=	Application Entity

CHAPTER I

INTRODUCTION

This chapter describes what are the reasons and objects for doing this work in section 1.1 and 1.2 respectively. For scopes of work and expected results will be illustrated in section 1.3 and 1.4 respectively.

1.1 Background and problems

At this moment, healthcare market is dynamic and extremely competitive. The success of patient treatment by using image information is an interested idea. Patient information has to be readily accessible in a format that facilitates accurate treatment decision. Relevant image and information must be delivered to the diagnostician and clinician immediately, regardless of where they are in the network. In the past, hospital-radiology information, which keeps patient records, has not been uses efficiently. For example:

- 20-25 % of X-rays Film are often unavailable when requirement for re-examination (1).
- Digital Modalities are the another ones, which do not make integration or interoperation between modalities, and they are in the proprietary formats of the departmental system (1).
- Hospital Wide PACS (Picture Archiving and Communications System) is a single vendor for all equipment and proprietary system (1).

To improve and solve these problems a DICOM (Digital Imaging and Communications in Medicine) standard is proposed. The DICOM format is used in medical imaging devices, providing the following capabilities:

- Network Image Transfers: Two medical devices are able to communicate to each other by querying or retrieving objects.
- On-line Imaging Study Management: Allows medical devices on network capability to integrate data with various information systems.
- Network Print Management: DICOM can print images on a network camera for example, workstations printing images on a single share camera.
- Open Media Interchange: Provides the capability to manually exchange objects and related information (such as reports or information). DICOM standardizes a common file format, a medical directory, and a physical media. Examples include the exchange of images for a publication and mailing a patent imaging study for remote consultation.

Most of digital medical devices used in modern clinical work usually correspond to the DICOM Standard. However, the DICOM Standard still is not used widespread in Thai's hospital. Due to the fact that Thai's hospital cannot purchase commercial DICOM software due to high price by not only the cost development, but also technical difficulty.

1.2 Objectives

1. To develop a software with DICOM capability, which has the same function as commercial DICOM software.
2. To develop object-oriented language C++ for computerizing data class library for the DICOM standard that supports DICOM image management and can be transfer over a network.

1.3 Scopes of work

The developed software has a role as Service Class User (SCU), which can display, transmit, receive and storage DICOM file via DICOM network.

1.4 Expected results

- To replace license DICOM software
- User-friendly interface

CHAPTER II

LITERATURE REVIEW

This chapter contains information about DICOM standard such as History of DICOM either DICOM file structure or DICOM network. In addition the expresses related paper involving with DICOM in section 2.8.

2.1 History of DICOM

In the 1970's because of computed tomography (CT) followed by other digital diagnostic imaging modalities, and the increasing used of computers in clinical applications. The American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) (2) recognized the emerging need for a standard method for transferring image and associated information between devices manufactured by various vendor. These devices produce a variety of digital image format.

In the 1980's radiologists and the manufacturers of medical imaging equipment realized. That how much the digital image system rapidly grow such as the display workstation, archiving system, hospital-radiology information system. Therefore they developed the connectivity and interoperability between equipment. In order to simplify and improve equipment connectivity. The medical equipment manufacturers were forced to develop DICOM the Digital Imaging and Communications in Medicine

Standard. The result of the development is that medical imaging devices can be directly connected to another DICOM compatible device. From this succession DICOM is used with other related fields (e.g., pathology, endoscopy and dentistry).

Before DICOM would be developed, the ACR-NEMA committee found an effective standard format that acquired concepts of recording image on magnetic tap. The format had been developed by The American Association of Physicists in Medicine.

The standard officially known as ACR-NEMA 300-1985 became known in the industry as ACR-NEMA Version 1.0 (2). It was released to the public at the annual meeting of the Radiological Society of North America (RSNA) in 1985. But it was not success in commerce because there were ambiguities and incorrect in standard.

A description of image such as patient name contained in the header portion of the format and it could be accessed with identifying data elements. The pattern of data elements composes of a tag or keys (the name of the element) and the committee adopted this idea.

A parallel hardware interface was developed for point-to-point connectivity. At that time a networking environment unsupported with transferred data sets between device in the large size. Similarly developer and user had a clear understanding of the requirements for interconnection and interoperation.

Three years later ACR-NEMA 300-1988 commonly known as ACR-NEMA Version 2.0 (2) was published at RSNA in 1988. This version corrected ambiguous parts with protocol and included new data element.

The concept of "50-pin connector" was developed to replace “point-to-point connector” by GE, Siemens, Philips and Vortech Data Inc due to requirement a network standard. At the same time Vortech developed a version of the standard which implemented ACR-NEMA Version 2.0 over TCP/IP. Siemens and Philips developed the Standard Product Interconnect (SPI). This was also a network implementation of ACR-NEMA Version 2.0. These early network implementations formed the basis for the third version of the standard—DICOM (2).

The new standard was developed called ACR-NEMA Version 3.0 because it replaces ACR-NEMA Version 2.0. The name was changed to DICOM to appropriate with medical works such as cardiology, endoscopy and other types of medical images.

DICOM works on computer network connections and medical equipment Which can communicate with diagnostic modalities such as CT, MR, PET, Nuclear Medicine, Ultrasound, X-ray, CR, digital radiography, digitized film, video capture and HIS/RIS information. It also supports the connection of networked printers, such as laser images (cameras).

2.2 DICOM Standard

This Standard, now designated DICOM Version 3.0 embodies a number of major enhancements to previous versions of the Standard (3):

1. It is applicable to a networked environment. The previous versions were applicable in a point-to-point environment only; for operation in a networked environment a Network Interface Unit (NIU) was required. DICOM Version 3.0 supports operation in a networked environment using industry standard network protocols such as OSI and TCP/IP.

2. It specifies how devices claiming conformance to the Standard react to commands and data being exchanged. Previous versions were confined to the transfer

of data. But DICOM Version 3.0 specifies through the concept of Service Classess the semantic of commands and associated data.

3. It specifies levels of conformance. Previous versions specified a minimum level of conformance. DICOM Version 3.0 explicitly describes how an implementers must structure a Conformance Statement to select specific options.

4. It is structured as a multi-part document. This facilitates evolution of the Standard in a rapidly evolving environment by simplifying the addition of new features. ISO directives which define how to structure multi-part documents have been followed in the construction of the DICOM Standard.

5. It introduces explicit Information Objects not only for images and graphics but also for studies, reports and etc.

6. It specifies an established technique for uniquely identifying any Information Object. This facilitates unambiguous definitions of relationships between Information Objects as they are acted upon across the network.

The DICOM Standard is structured as a multi-part document using the guidelines established in the following document (3):

- PS 3.1: Introduction and Overview describes the overall structure of the Standard.
- PS 3.2: Conformance specifies the general requirements, which must be met by implementations claiming conformance and contents of a Conformance Statement.

- PS 3.3: Information Object Definitions the structure and attributes of objects, which are operated upon by Service Classes (PS 3.4). These objects include images, studies, and patients.
- PS 3.4: Service Class Specifications defines the operations that can be performed on instances of Information Objects PS 3.3 to provide a specific service. These services include image storage, retrieval, and printing.
- PS 3.5: Data Structure and Semantics specifies the encoding of the data content of messages, which are exchanged to accomplish the operations, used by the Services Classes (PS 3.4).
- PS 3.6: Data Dictionary defines the individual information Attributes that represent the data content (PS 3.5) of instances of Information Objects.
- PS 3.7: Message Exchange specifies the operations and protocol used to exchange messages. These operations are used to accomplish the services defined by the Service Classes (PS 3.4).
- PS 3.8: Network Communication Support for Message Exchange defines the services and protocols used to exchange messages (PS 3.7) directly on OSI and TCP/IP networks.
- PS 3.9: Point-to-Point Communication Support for Message Exchange defines the services and protocols used to exchange messages (PS 3.7) on the DICOM 50-pin interface.
- PS 3.10: Media Storage and File Format for Data Interchange defines the logical formats for storing DICOM information on various media.
- PS 3.11: Media Storage Application Profiles defines a means for users and vendors to specify a selection of Media among.

- PS 3.12: Storage Functions and Media Formats for Data Interchange reference3s industry specifications for the Physical Media and Media formatting file systems. It includes 5 types of Media: 5.25 inches MOD Mbytes, 5.25 inches MOD 1.3 Gbytes, 3.25 inches MOD 128 Mbytes, and the 3.5 inches Floppy Disk.
- PS 3.13: Print Management Point-to-Point Communication Support
- PS 3.14: Grayscale Standard Display Function

The relationship of each part of DICOM standard as illustrated in figure 1.

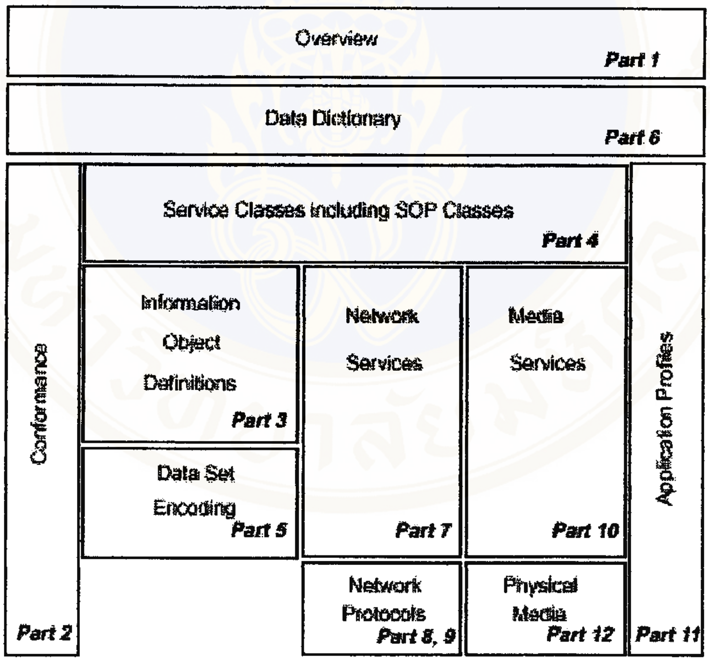


Figure 1: Relationship between each part of DICOM standard (3)

2.3 DICOM file structure

The ACR and NEMA in PS3.10 describe the DICOM file format specification "Media Storage and File Format for Media Interchange" of the DICOM Standard. See a basic DICOM file structure in figure 2.

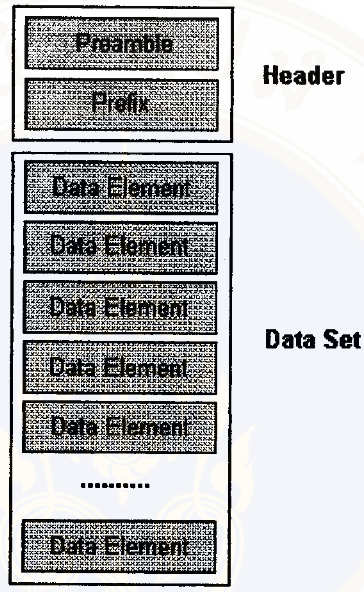


Figure 2: A basic DICOM file structure (4)

2.3.1 File header

It consists of a 128-byte file preamble, followed by a 4-byte DICOM prefix (DICM). The DICOM file may have or have not the header file. Preamble portion does not require any data in its. If the Preamble is not used all 128-byte will be set to 00H. Typically it is used to facilitate access to the images and other data in the DICOM file. A multimedia data for example contained in a DICOM data set. The same file can be accessed in two ways by a multimedia application using the preamble and by a DIOCM application without preamble. The structure of preamble is not a DICOM Date Element with a Tag and a Length.

2.3.2 Data Set

It represent an instance of a real world Information Object. It is constructed from Data elements containing the group of tag number and its encoded values. Each DICOM file should contain a single Data Set representing a single SOP Instance related to a single SOP Class. DICOM file service set the end of DICOM file by indicating at the end of the Data set.

2.3.3 Data Elements

They have unique value and occur at once in a Data set. Each data element is ordered by increasing of data element tag number. There are two types of data elements: Standard data elements and Private data elements. Standard Data Elements have an even Group Number that is not (0000,eeee), (0002,eeee), (0004,eeee), or (0006,eeee). Accordingly, private Data Elements have an odd Group Number that is not (0001,eeee), (0003,eeee), (0005,eeee), (0007,eeee), or (FFFF,eeee).

There are three structures of data elements, which should have only one type in data set. Two of them contain the Explicit VR (value representation). That they are different in expression their length. The other one is Implicit VR. All three structures contain the Data Element Tag, Value Length and Value for the Data Element.

The Data Element Tag is an ordered pair of 16-bit unsigned integers representing the Group Number followed by Element Number.

The Value Length is a 16 or 32-bit unsigned integer, depending on whether this is explicit or implicit VR. That contains the Explicit Length of the Value Field as the number of bytes. However, it does not contain the length of the data element tag, Value Representation and Value Length Fields.

The Value Field is an even number of bytes containing the Value(s) of the Data Element.

The Value Representation is a two-byte character string, which is encoded from the DICOM default character set. Each VR is defined in the Data Dictionary as specified in PS 3.6 of the DICOM Standard. The VR is either explicit or implicit depending on the negotiated transfer syntax. Data set should have only one type of VR neither explicit nor implicit. When using the explicit VR structures the data element should be composed of four consecutive fields: Data Element Tag, VR, Value Length, and Value. The different VR will construct the different structure of data element.

For VRs of OB, OW, SQ and UN, the 2 bytes are reserved to use in the later versions of the DICOM Standard. The reserved values are set to 0000H and without decoded. From these VRs, the value length field is a 32 bit unsigned integer.

The 16 bits following the two character VR Field are reserved for use by later These reserved bytes shall be set to 0000H and shall not be used or decoded. The Value Length Field is a 32-bit unsigned integer. If the Value Field has an Explicit Length, then the Value Length Field should contain a value equal to the length of the Value Field. Otherwise the Value Field has an Undefined Length as shown in figure 3.

For all other VRs, the reserved field disappear from the following two character VR field. They have the Value Length Field is the 16-bit unsigned integer. The value of the value length field still equals the length of the value field as shown in figure 4.

Group Number	Element Number	Value Representation	Reserved	Value Length	Value Field
2 bytes	2 bytes	2 bytes	2 bytes = 0x00, 0x00	4 bytes	"Value Length" bytes

Figure 3: An Explicit VR structure such as OB, OW, SQ, or UN (5)

Group Number	Element Number	Value Representation	Value Length	Value Field
2 bytes	2 bytes	2 bytes	2 bytes	"Value Length" bytes

Figure 4: An Explicit VR structure such as not OB, OW, SQ, or UN (5)

When using the Implicit VR structure the Data Element shall be constructed of three consecutive fields: Data Element Tag, Value Length, and Value. Similarly the other values have the same structures as the explicit VR as shown in figure 5.

Group Number	Element Number	Value Length	Value Field
2 bytes	2 bytes	4 bytes	"Value Length" bytes

Figure 5: An example of a Data Element with an Implicit VR (5)

2.4 DICOM file format

The Service Class describes the roles of user and provider depending on the individual context of the services. With DICOM both roles are named: Service Class User or SCU (client) and Service Class Provider or SCP (server) see in figure 6.

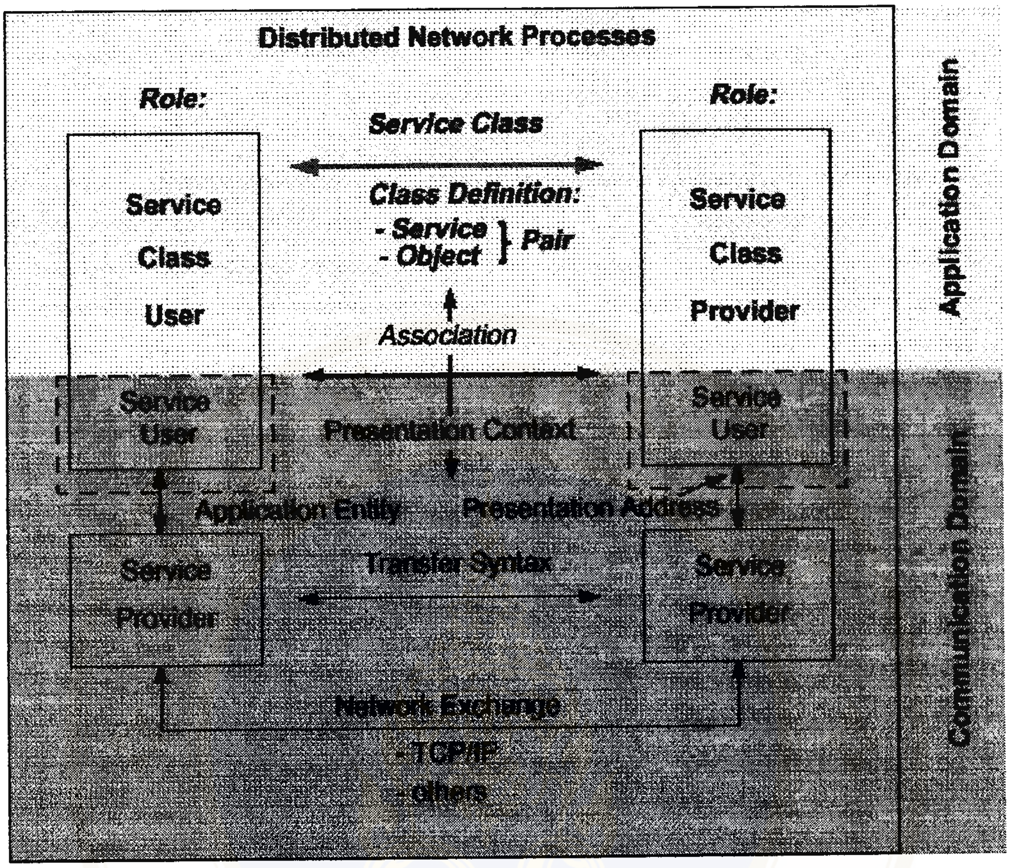


Figure 6: DICOM server classes (6)

2.4.1 Service Classes

The group of operation such as store, get, find or move defined from the DICOM standard are called DICOM message service elements (DIMSE). The service provided by DIMSE have both DIMSE-C (related to Composite IODs) and DIMSE-N (related to Normalized IODs). The encoded information and the DIMSE are combined in a class called Service Object Pair Class or SOP. The SOP Classes represent the element unit of functionality defined by DICOM.

DICOM has defined a type of Service Class. That any functionality will be standardized in DICOM standard in Service Class. Image information is encoded in composite IOS and included in the composite services. The Service Classes of this group are Storage Service Class, Query/Retrieve Service Class and Study Contents Notification.

First, Storage Service Class consists of SOP Classes for each modality. This Service Class involves the exchange of the data across a network. This Service does not need to mange image data.

Second, Query/Retrieve Class composed of FIND, MOVE and GET SOP Classes. The FIND can be used to query for a collection of images. Similarly, the MOVE and GET can be used to initiate for retrieving data.

Third, Study Contents Notification is used to notify an image management facility about the images created during a study. Moreover, it is used to check image data transferred. Before the transferring of image data, it is used to initiate any value.

2.4.2 Service Object Pair

Information and operations involve with the Service Class. The are combined with object oriented class definition, called a SOP class. Each SOP Class, a single Information object Definition or IOD is combined with one or more services.

Before information exchange, partners have to agree to use a SOP class and they have to act as their role. The SOP Class identification needs to deal first. For using the Service Class and other derived definitions, partners have to process the environment function together via the services provided by the exchange domain.

2.4.3 Information Object Definitions

The IODs, grouped in Information Entities, are information part on a SOP Class. Each entity contains information about a real world information such as a patient, an image depending upon the definition by the Service Class. An IOD consist of normalized IOD, one single information, or composite IOD, a combination of information. Service Class, involving management function, used to normalized IOD and for handling the flow of image uses composite IOD.

A Composite IOD represents parts of several entities included in the DICOM model of the real world. These related real-world objects provide a context for the exchanged information. The context is exchange between Application Entities when composite IOD is communicated as shown in figure 7.

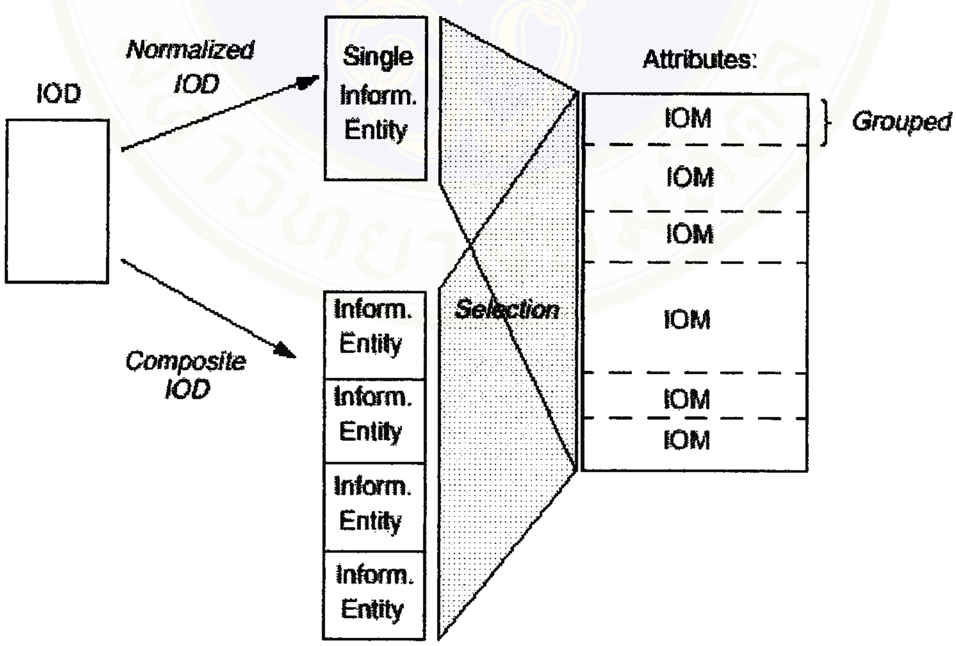


Figure 7: Relation between IODs and attributes (6)

The relationship between different information entities of composite IOD is expressed in the information model (figure 8). On the other hand, normalized IOD, there is no need any model.

According to figure 8, information entity consists of list of attributes, which describe the detail of its information. Attributes are grouped into information object modules or IOMs. IOMs can be used in more than one IOD. That it makes easily to describe of related attributes.

Attributes are basic information entities and have a consecutive structure: Attribute Name, Attribute Tag, Attribute Description, Value Representation, Value Multiplicity, and Type classification.

The definition of classification depends on SOP Class and SCU or SCP role. Each attributes is forced to have a value (type 1) or forced with or without a value (type 2) or optional (type 3).

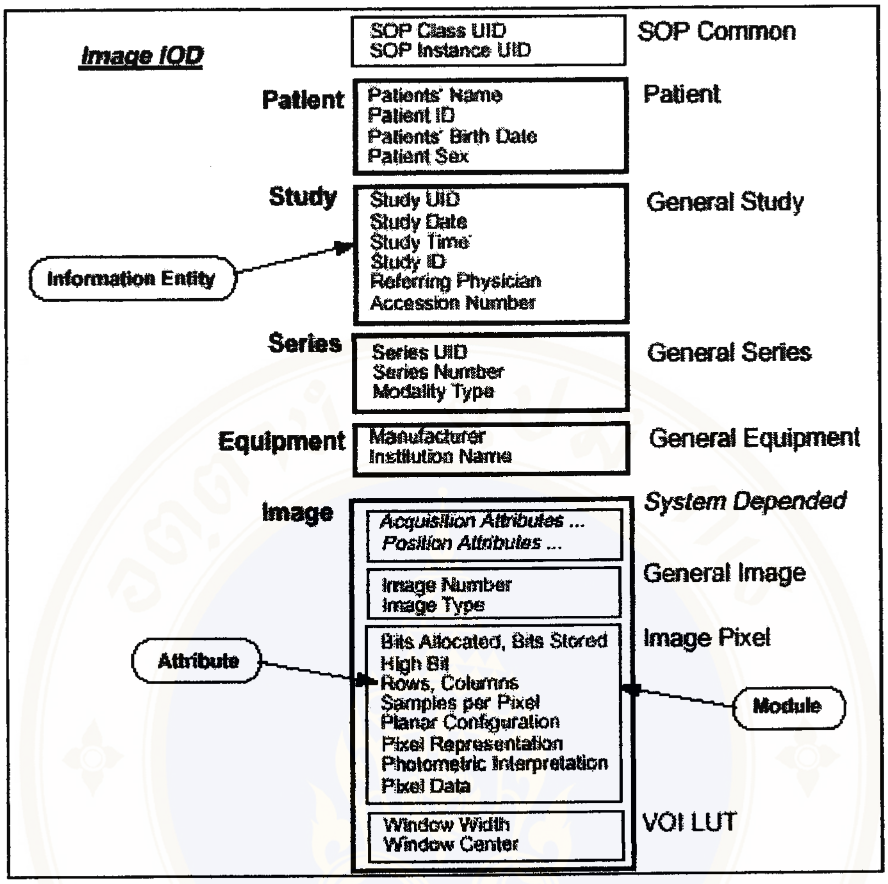


Figure 8: Example of composite image IOD (6)

2.5 DICOM Network Concepts

2.5.1 Application Entity

It is a major issue for a DICOM network connection. The initialization of Application Entities has to be done before SOP instances can be exchanged. For DICOM network partners recognize each other via Application Entities, which has an Application Title to setup communication. Application Entities contain functions to setup connection and to transfer their function. The Application Title is used to identify an application as source or destination in a directory.

2.5.2 Presentation Address

Although there is an Application Title in the communication, a network address is necessary. This address called Presentation address points to the Application Entity address. It is called Presentation Address because the service user is the Application layer, the service provider is the Presentation layer. The boundary of both layers is in the network access point where the data is transferred from the application layer to the networking layer. Each access point has a unique address, but the mapping of Application Title to the Presentation Address is not unique. The format of the Presentation Address depends on the network protocol used. DICOM networks use TCP/IP protocol stack.

2.5.3 Association Negotiation

The connection between two Application Entities is called an Association. For an Association, the Application context, defined in the DICOM standard, is a number of communications issues and both sides must agree on this context definition. An Application context is identified with a UID. This UID will be transferred to the partner while initiation of an Association. When partner receives UID, it can decide to accept the establishment or reject it in which depend on comparing UID. After this negotiation, both partners know the capability and limitation of each other because the application context contains the functionality of the information exchange. When an Association is no longer required, the Association is terminated.

2.5.4 Presentation Context

Agreed number between partners identifies a Presentation Context. It expresses the SOP Class for the information exchange. It involves about the transfer syntax used between the process.

2.5.5 Network Protocols

In the application layer, there are two groups of services have to be available for a DICOM implementation: the Association Control protocol (ACSE) and DICOM Message protocol (DIMSE). ACSE is a standard OSI protocol and DIMSE is used to implement the DICOM service. The information between these groups and application layer is DICOM interface. The ACSE, DIMSE and DICOM interfaces are part of the DICOM Application Context. The interface between application and DICOM Application Interface is the Application Program Interface (API). The API provides functionality to connect to other application as shown in figure 9.

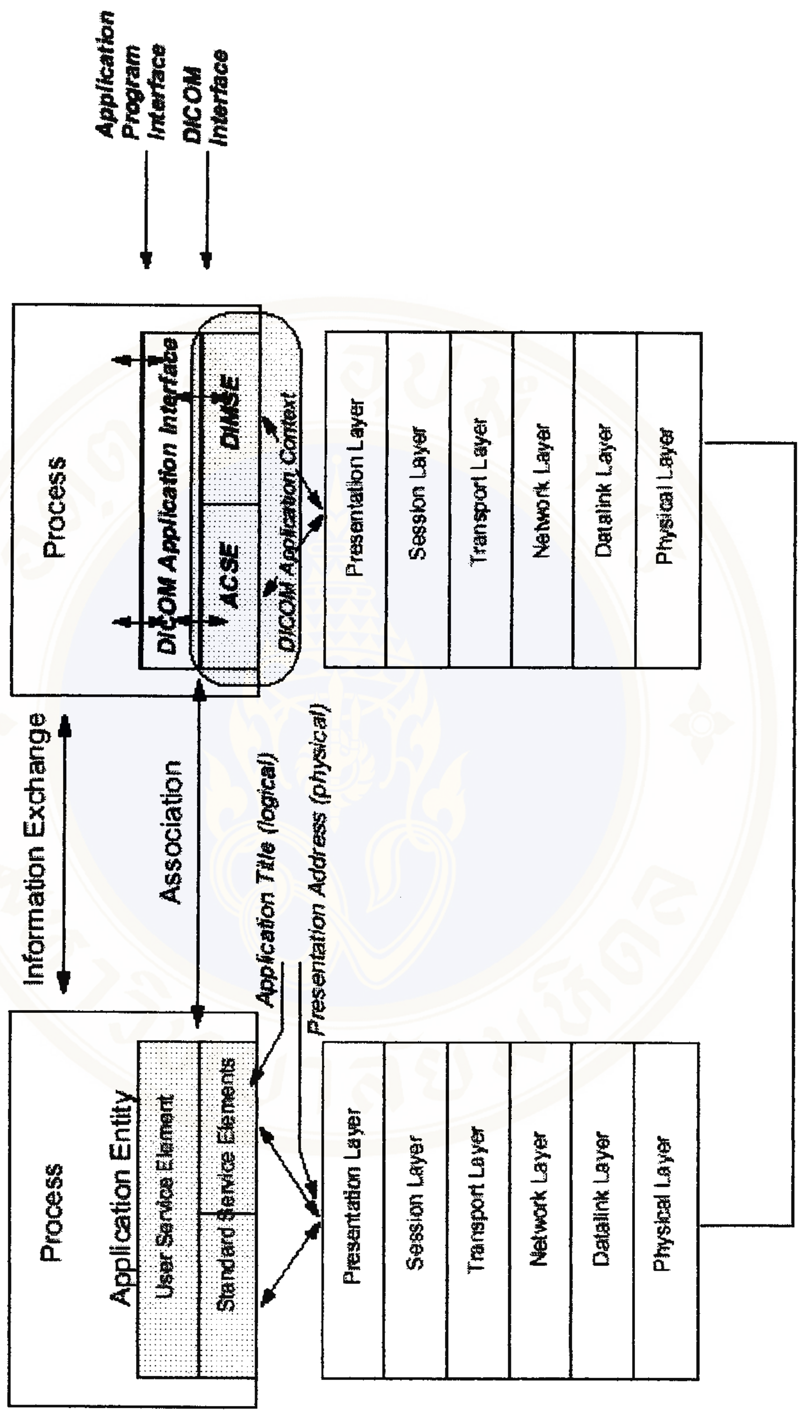


Figure 9: OSI Layers (2)

2.5.6 TCP/IP Protocol Stack

The combination of a TCP/IP stack and OSI layer is used to implement DICOM across the network. The application layer, presentation layer and session layer is included to one layer. This is DICOM Upper Layer or DUL. The DUL is a layer on the TCP/IP stack. The DUL uses DICOM interface between its and TCP layer. The DICOM Association between the Application entities is mapped to a TCP connection. The Presentation Address is mapped to a TCP port number, combined with the IP Number or Host name. This combination of TCP port number and IP number is called the Socket Address. Local host address and remote host address. Establish a TCP connection by using IP numbers, unique on network system. Each TCP connection is uniquely identified by the combination. The TCP port has to agree on both sides. It is usually defined as 104. TCP connection is shown in figure 10.

2.5.7 Transfer Syntax

Before exchanged information across network, the data have to be encoded to a byte stream. Three aspects have to be defined by the transfer syntax: VR, little endian or big endian, and compression. The handling of the transfer syntax is part of the service provider.

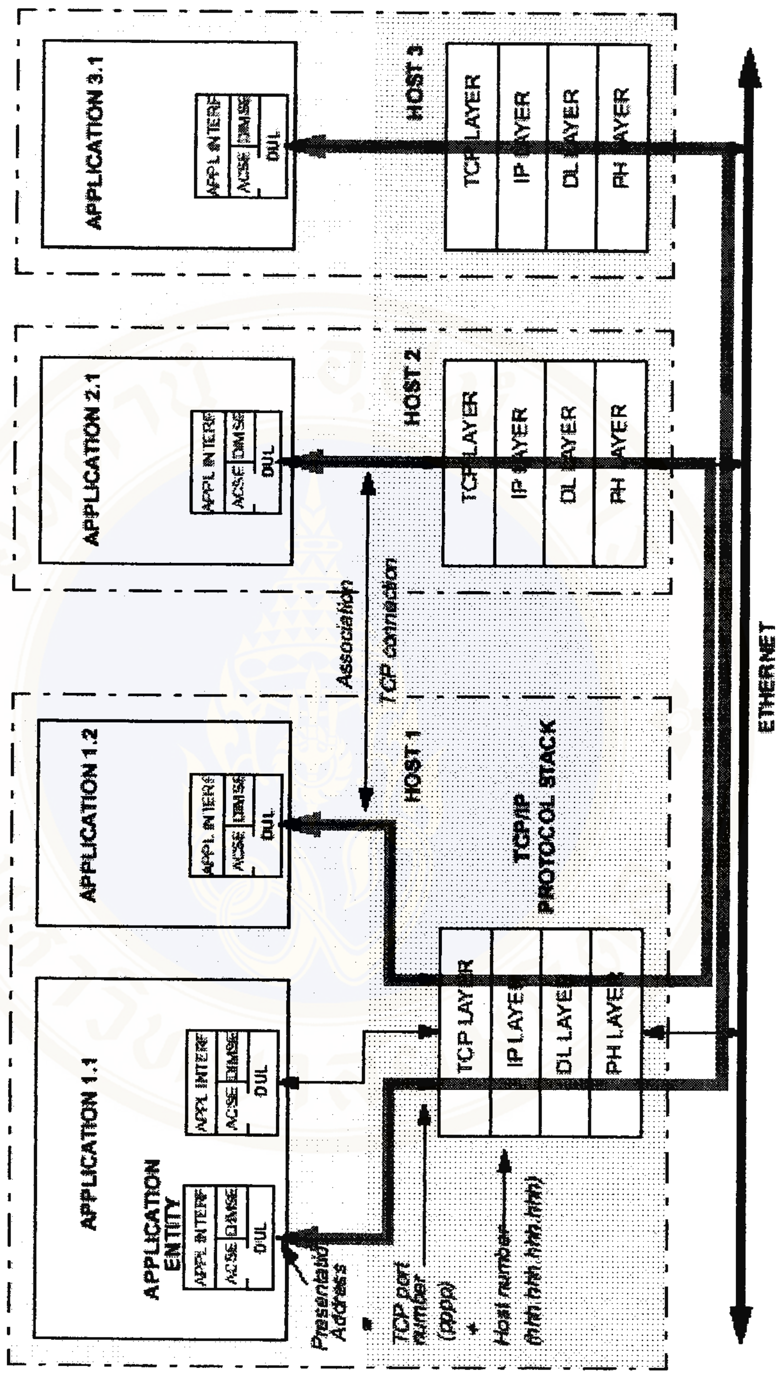


Figure 10: TCP connection (6)

2.6 DICOM Image SOP Instances

DICOM SOP Classes composed of an IOD and DIMSE. Image data are handling limited to Storage SOP Classes and media storage. Therefore, the image processing reference use a non-DICOM terms as Image SOP Class/Instance.

2.6.1 Image Information Model

The image information of DICOM has represented into the model. The structure of model needs to have a uniform and to describe the relation between information. An image information model is constructed from the image managing in the radiology department. Images are collected from one or more modalities into a patient folder. Images are ordered along the type of examination. Because each modality has its own terminology for ordering such as examinations, run, scans, or slice. The collection of image data needs to have the same information model as well as image data from the different source has to be collected in a single environment.

2.6.2 Mapping Real World Examinations

The DICOM Image Information Model is based on assumptions about the way in which information from different modalities is related, see figure 11. The four levels of this information model are Patient, Study, Series and Image.

2.6.3 Patient Level

The Patient level contains the identification of patient. The patient level is the highest level in which a study level is under it because more than one study of a patient can exist.

2.6.4 Study Level

The Study level is the most important level in this information model. A study level is used to collect information from the different system for a single examination requesting. The identification information can contain references to information, which is related to the same study in an administration system. All the image data is collected together with the same study as the root. A single patient can have many studies due to request for an examination procedure (see figure 11).

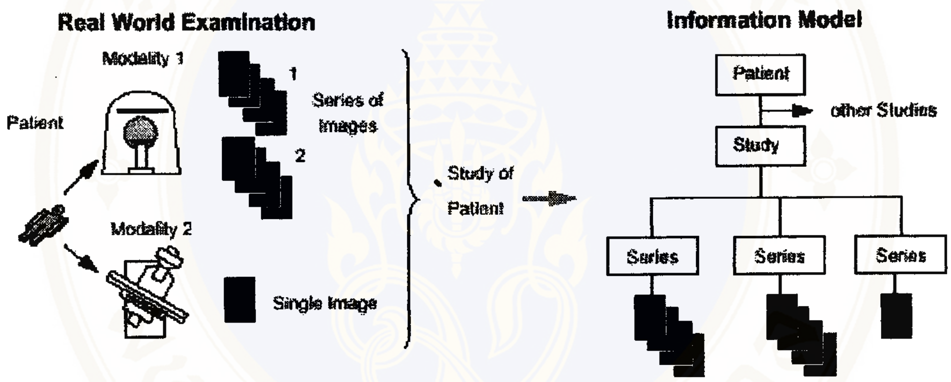


Figure 11: Levels of information model (6)

2.6.5 Series Level

Series are a collection of related image coming from a single modality. The same types of image are grouped into series. For grouping image does not concern about image acquisition and the image relationship is defined by acquisition. The series level identifies the image creation from modality such as date/time, series date, data bout examination and equipment. The acquisition, a spatial and temporal relation, can be grouped into a series. Some systems produce more than one image from an acquisition. CT systems for example image is collected in series having a spatial relationship. The next scan will be a new series because the scan is made from a different position. A new series must start when the relationship between images is no longer.

An another way for grouping image is to collect images of a single body part during an examination. For example, when a modality produces a number of images of a patient's stomach from different position and different moments in the examination. The images can be collected into a single series. For each type of modality has the different type of creating series, the rules used in a collection are in the DICOM conformance statement.

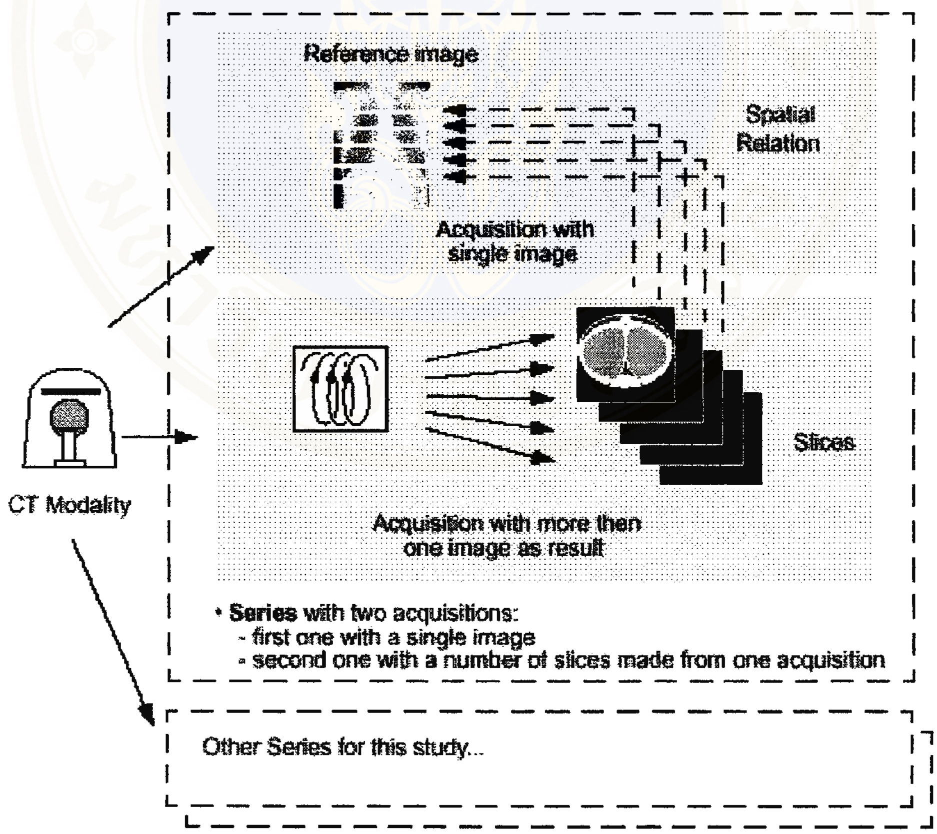


Figure 12: Image reconstruction (6)

2.6.7 Image Level

The lowest level of the information model is the Image level. Each image contains acquisition and positioning information as well as the image data itself. The image level can contain a single image, two images (biplane system) or multi-frame image. The multi-frame images are the duplication of information at the higher level. That the increments in the movements of the system and time are equal for all the single frames. The relationship between frames is used to determine whether system should be a series of single image or a multi-frame image as shown in figure 12.

2.7 Classification of Image Data

The SOP Instance information can be classified into it. And place it in the information mode due to relationship between model and creation process. In figure 13 show an overview of the classification and the relation of modality. Any classes are created while performing an examination. The results of attribute from each class can group into Image SOP Instance.

2.7.1 Patient Information

In a radiological department the patient information come from other sources, such as information systems or the request for the examination on paper, but this class contains information about the patient after an examination such as patient's name, patient ID, Patient' Birth Date. This information is acquired from modality in which they can be changed it easily.

The attributes in Patient Class are important. They are used to identify and connect to other information. So, the modality needs to include this attribute in the SOP Instance.

2.7.2 Study Information

Study information is a class including source of information. The modality will add study information about the patient undergoing the examination. Information from other system includes an identification of the study. A study Instance UID is used for identification. If there are no Study Instance UID from outside the modality, the modality has to generate their UID.

Images are copied from a local host to a remote host. They need to have the same Study Instance UID. Otherwise, there will have a problem. Those images have the different study identifications even coming from a single original examination. Similarly images does not collected together with the same modality. Other information such as names of physician is also included in the level.

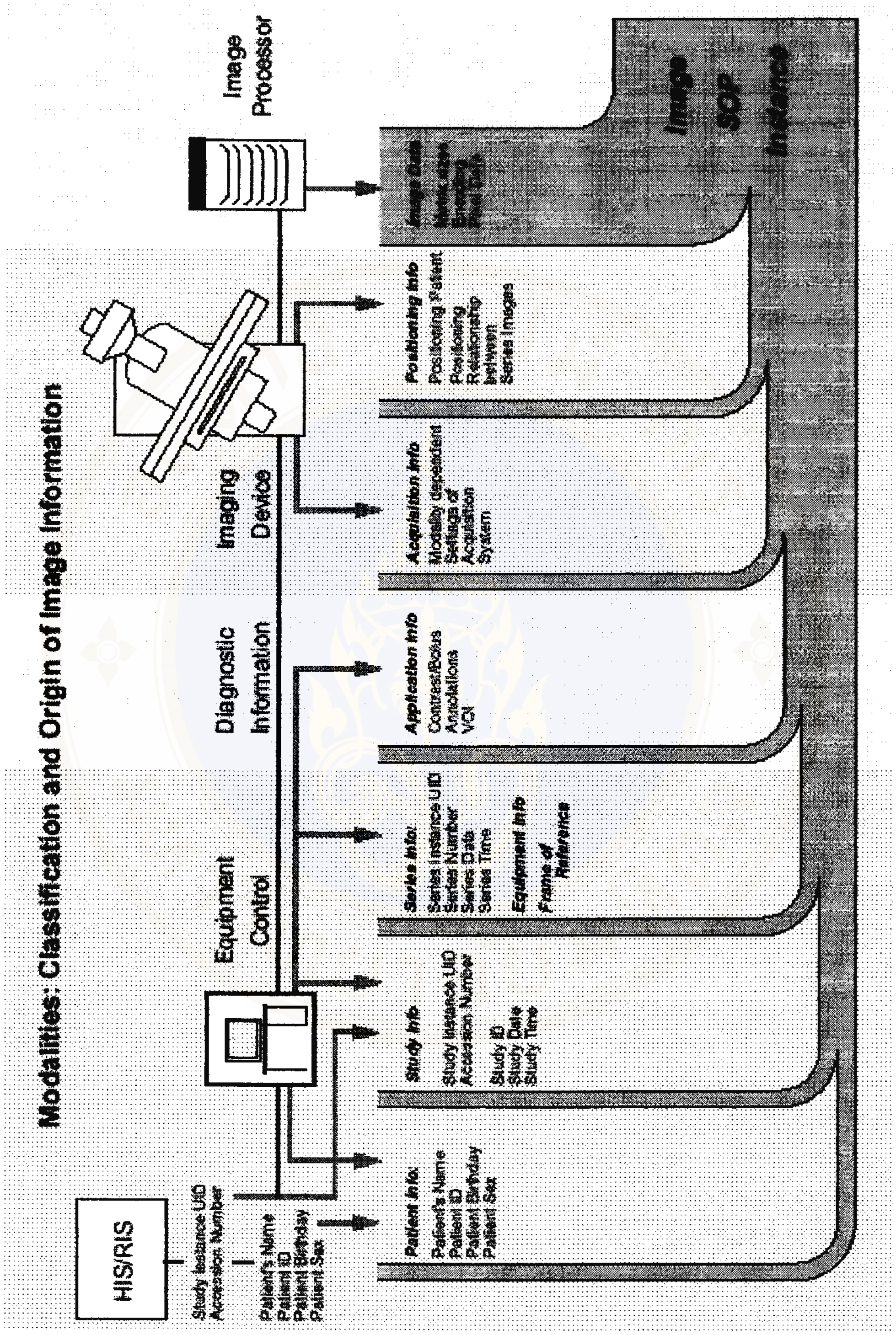


Figure 13: Classification of Image Data (6)

2.7.3 Series Information

In this class the type of system, the location and identification of the system are considered. Series Instance UID is used to be identification in series of image data environment. The local use Series UID to make a series sequence in a study. The Series ID has only a meaning for the modality. The other information, people involved, series performance, positioning relevant for the whole series are supplied in the series classes.

The equipment information contains the used information in series. It includes information about the location, type and series identification, calibration, and etc. This information may be shored with series belonging to the same study. A frame reference is used to group images, which have a spatial or temporal relationship.

2.7.4 Application Information

The attributes in this class give information about the image contained in the SOP Instance required for the diagnosis and other applications for example comment, details about contrast agent, therapy and devices.

The Value of Interest (VOI) described the window width and window center of image, are very important member of this class. The VOI is the selection of the full range of pixel values. It is used to display or print image. The information inside the specific range is displayed in gray scale, but all others will be displayed in back and white value. Information about draws lines or add text can be in the form of overlay matrices, which appear into the pixel matrix or viewing station attribute.

2.7.5 Acquisition Information

In this class contain details of the acquisition system setting such as X-Ray kV Values, Collimator shape, Image Intensifier diameter. The information depends on the type of modality.

Images of same acquisition can be identified with on acquisition number. The acquisition does not relation with the DICOM Information Model and no equivalent UID identification.

2.7.6 Positioning Information

In this class contain image information about the positioning of the image. The image matrix is positioned by using a simple term such as anterior, posterior, right, hear, and etc.

In a series class, image information from CT or MR provides three-dimensional space of the patient's body. These information supports physician to plan the radiotherapy treatment. The positioning information of vascular system is to describe the movement after injecting contrast/bolus through the vein. This information is used for post-processing a group of images to create a single picture, with elements from the collection of images, showing the progress of the Bolus.

2.7.7 Image Data Information

The image data acquired by the acquisition system are in the class. This class describes the specific detail about the interpretation of pixel information such as the size of the pixel matrix, the presentation of a pixel value. When modality produce color picture, information about image color plane has to be provided in class. The image is identified by the Image UID, used as SOP Instance UID. This UID is used to identify the instance when transferred or retrieved from an image store or, to identify the image entity itself when using it in a hierarchical tree of information.

2.8 Related papers

There is a paper, which implementation DICOM software namely "Object-Oriented Implementation of a DICOM Network Client in Smalltalk" (7). It uses Cyclops, a knowledge-based image analysis system developed at the University of Kaiserslautern, to analyze such images. The Cyclops is divided into class for example CyclopsDBObject can be divided into CyclopsDicom Image, CyclopsCT or CyclopsMR as shown in figure 14.

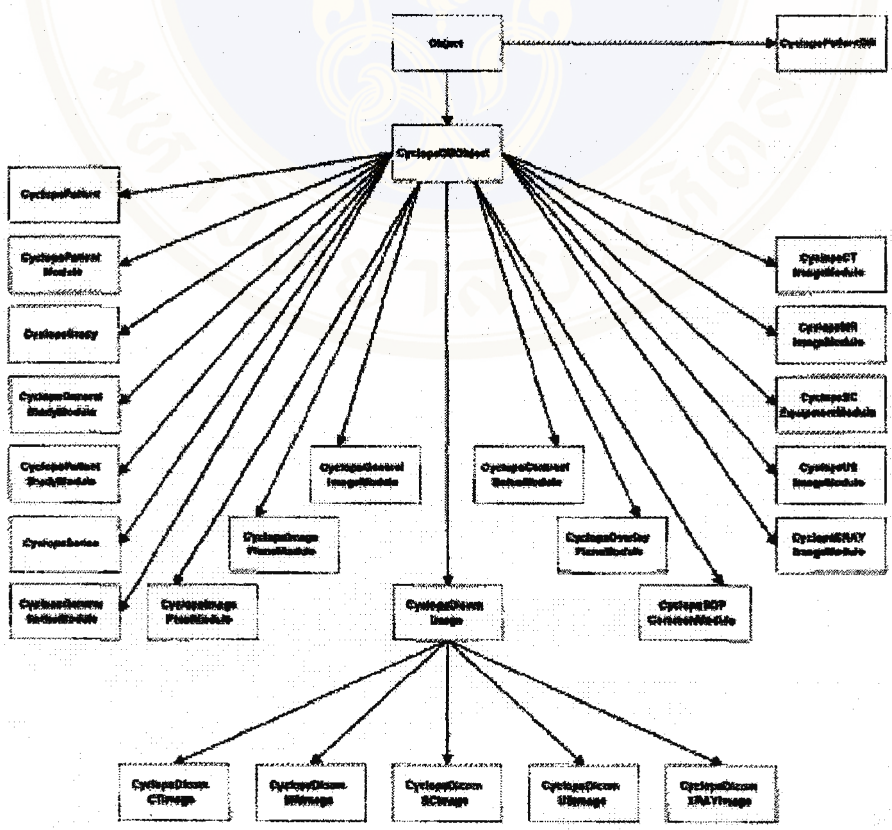


Figure 14: Dividing class of Cyclop (7)

The author implemented the SCU role. For the communication with the image server needed the standard Query/Retrieve Service Classes, mainly C-STORE and C-FIND, C-GET for every Information Entity and Module.

- The C-STORE services for requesting the storage of Composite SOP Instance information by the DICOM Image Server.
- The C-FIND services for matching a series of attribute strings against the attributes of the set of SOP Instances managed by the DICOM Image Server. The C-FIND service returns a list of requested attributes and their values for each match.
- The C-GET services for fetching the information for one or more Composite SOP Instances from the DICOM Image Server.

The implemented software gives access to all patients stored in the database and displays all patient, study and series attributes. It is possible to change them and store these changes in to the database. The series images are distributed as thumbnails, so that the user can check and it can distributes all images of a series on screen. In addition, it gives the possibility to change the images sizes and mark special regions with a red pen without destroying the original image. This is realized with the overlay-concept described in the DICOM Standard. In case of a contrast series the images are distributed in a specific way so that they can easily be interpreted by radiologist. The image itself can be manipulated by using several image-processing operators.

According to a paper namely “Class library design for DICOM in C++” (8), designing a computerized data class library for the DICOM standard which supports DICOM image management and transfer over a network has been developed by using the object-oriented language C++. The classes have been used to build a computed radiography archive system, and will be used in developing DICOM compatible medical imaging applications based on a variety of network protocols. The paper describes the data structures, implementation of various basic classes including Data Element classes, Composite Image IOD classes, DIMSE Service classes, and Message Classes.

There are classes built in order to manage DICOM such as Class DataElement define the object of a DICOM data element- tag, length of value and the value. As well as Class StringDE is defined as a concrete data type with the appropriate constructors, destructor, and overloaded operators defined, which can support resizeble, variable-length strings of arbitrary binary data and concatenation, substring, and filling operations.

For composite image IOD class c_image is defined to contain pointers to the common models of the image IOD as its data members. And for DIMSE services, class C_STORE_RQ, C_STORE_RSP to be used such that services based on the Service Class Specification. A message is defined as a context dependent ordered set of information elements. There are three message classes: L_msg, F_msg and C_msg defined for different purposes in the communication. L_msg is defined for a linked message, which is the query message form. F_msg is defined for a formatted message, which is the transfer/storage message form and C_msg is defined for a composed message, which is the base message form.

2.9 Summary

In summary from concepts of DICOM standard, the development of DICOM software need to concern with structure of DICOM file and DICOM networking. The steps of processing have to use benefit from header and Data Set of DICOM file.

There is an identification DICOM file by using the header file, and uses patient information from data element. In addition, DICOM network concept will be used in development such as SOP Class, TCP/IP protocol stack, or DICOM Image Information Model.

CHAPTER III

METHODOLOGY

This chapter is divided into two sections, materials (section 3.1) and methods (section 3.2). Materials have both hardware resources and software resources. Methods can be subdivided into 6 steps: data investigation, system design, software design, implementation, software test and system test.

3.1 Materials

3.1.1 Hardware resources

3.1.1.1 PC Computer 1 unit

CPU	:	Intel Pentium 166 MHz
RAM	:	64 MB
Hard Disk	:	10 GB
Monitor	:	Super VGA Monitor
Peripheral	:	Keyboard, Mouse, and LAN card

3.1.1.2 Notebook Computer 1 unit

CPU	:	Intel Pentium 100 MHz
RAM	:	40 MB
Hard Disk	:	3 GB
Monitor	:	Super VGA Monitor

3.1.2 Software resources

Operating System	:	Microsoft Window 98
Compiler	:	Microsoft Visual C++ Version 6.0
DBMS	:	Microsoft Access 97
DICOM tools	:	LEADTOOLS Medical imaging V. 12.0
DICOM software	:	eFilm Version 1.320, PiView Version 3.0

3.2 Methods

Research methodology is divided into 6 steps as the following: data investigation, system design, software design, implementation and software test as shown in figure 15.

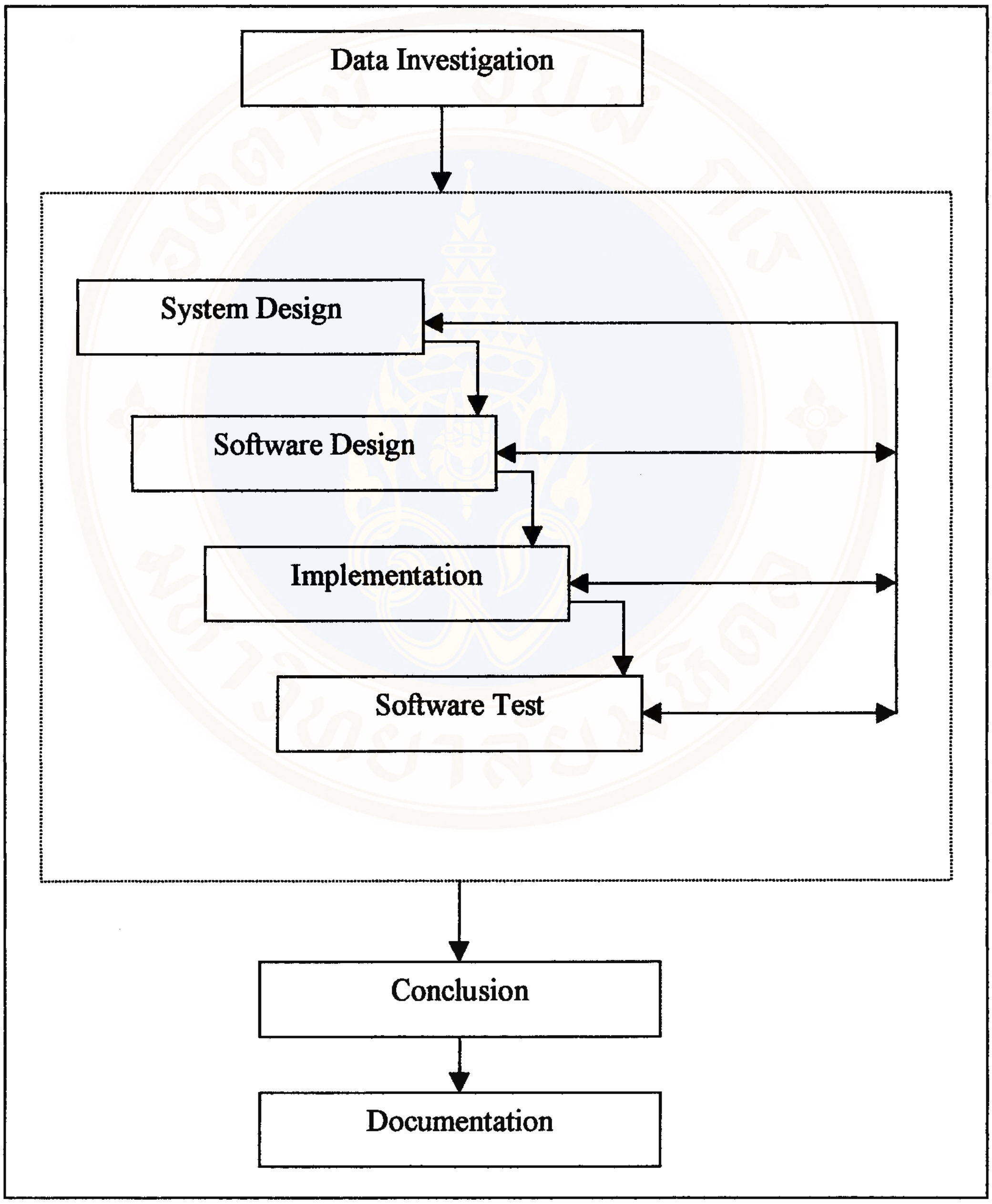


Figure 15: Processes of doing research

3.2.1 Data Investigation

In this step, related information was collected from many sources such as books, journals, electronic data, and so on, examples of data are:

- Information of DICOM format
- Examples of DICOM software including source codes and documents

3.2.2 System Design

Before developing, program specification will be determined. The system design can be divided into 2 steps, application design and database design.

3.2.2.1 Application Design

This step is to design

- Network connection
- Data storage the DICOM file
- Query/Retrieve the DICOM file
- Load, Save and Display the DICOM file

3.2.2.2 Database Design

The database is used to store DICOM information such as patient's name, patient's sex, patient identification, and so on.

3.3 Software Design

Microsoft Visual C++ is chosen for this work. Because the C++ is one of several new languages that use a programming style called object-oriented programming. To write a large software, which is correct, readable, modifiable, affordable, and efficient, requires the same creative effort and persistence characteristic as other endeavors in science and engineering. In addition, the C++ programming language provides many mechanisms for expressing abstractions and relationships among them. Some of these, like subroutines and data structures, are provided in languages widely used in scientific and engineering programming.

The implementation this program will also applies source codes from "LEADTOOLS Medical Imaging Suite", which has C++ source codes. LEADTOOLS DICOM Class libraries (DLL) provides two custom controls developed by LEAD Technologies, Inc. The LEAD DICOM Control Module (32-bit) provides a custom control for incorporating DICOM image file format support into applications and supporting the DICOM Network Communication for Message Exchange.

Microsoft Access 97 is selected for storing information because of its popularity. In addition, it is flexible such that the C++ application can access data from the Microsoft Access.

3.4 Implementation

The LEADTOOLS Class Library is designed as a set of standard C++ objects. These objects use the LEADTOOLS API, which is standard Window Dynamic Link Libraries. There are 4 main procedures in the implemented software.

3.4.1 Loading and Saving Images

LEADTOOLS provides many options when loading and saving image files. Images are loaded at their color resolution, manipulated, and displayed. Image can also be converted to a specific color resolution (bits per pixel) as loaded or saved it. The code can be as simple as the following:

```
LbitmapBase TmpBitmap; /* LbitmapBase object for the temporary bitmap */
/* Load a bitmap at its own bits per pixel */
TmpBitmap.Load("c:\\images\\example1.dcm", 0, ORDER_BGR);
/* Save the image as a 24-bit Window BMP_file */
TmpBitmap.Save("d:\\temp\\test.dcm", FILE_BMP, 24, 0);
```

3.4.2 Displaying a DICOM image

The LEADTOOLS's functions for painting an image use property of four points on rectangle. Two of them, the source rectangle and the destination rectangle, determine how big an image is scaled (zoomed) when it is displayed. Another the source clipping rectangle and the destination clipping rectangle, determine which part of an image is painted.

When painting an image with a display mode of 256 colors or less, User must specify the palette to use. LEADTOOLS gives the following two choices for painting to a device of 256 colors or less:

- A fixed palette, The fixed palette is the same for any image. It is good when more than one image is displayed at the same time and when user wants all of the images to look good. A fixed palette is always used when painting a high-color or true-color image. The default is the LEAD fixed palette.
- The bitmap own palette is the best available representation of the image's colors. It may be an optimized palette or a fixed palette, depending on how the image was created.

DICOM Client program uses a fixed palette because it does not consider the image resolution.

User can zoom on an image either fit to window or normal size using these functions.

- ZOOM_FIT: To fit the bitmap in the window while maintaining the aspect ratio.
  - ZOOM_STRETCH: To stretch the bitmap to the window.
  - ZOOM_NORMAL: To display the bitmap in its normal size.
  - ZOOM_FITWIDTH: To fit the bitmap to the width of the window.
  - ZOOM_FITHEIGHT: To fit the bitmap to the height of the window.

3.4.3 Create DICOM network

When establishing a connection between two AEs, for the purpose of transferring DICOM data and messages, the first step is to create a DICOM Network connection. User need to define his or her PC's AE Title and host name, and know the AE Titles, IP address (host names), port numbers of the remote hosts that wish to network with. The definitions of each configuration are:

- IP Address - The Internet Protocol address is a numeric address that identifies the station to other TCP/IP devices on the network. The IP addresses consist of four integer numbers separated by periods (e.g., 10.31.4.156).
- Host name - The host name is a common alphanumeric alias for the IP

address. Although user can use the IP address on DICOM network instead of the host name, the host name is better than IP address on the complex network system.

- Port - The TCP/IP protocol uses a number of ports for communication. The standard port number that is used for DICOM networking is 104. This means that this program will receive and send files only through the Port 104.
- AE Title - Application Entity Title is an unique identity value of all connected DICOM modalities and workstations. The AEs Title defines the device as a valid DICOM source and / or destination.

The AE acting as both a SCP and a SCU must have a DICOM Network object. In addition, every time the SCP accepts a new connection request from the SCU, the SCP creates information about the SCU and that particular connection. This program uses codes from LEADTOOLS that provides functionality for connecting two AEs over a network. After two AEs are connected over a network, they can establish a DICOM Association.

The establishment of an Association is the first step in setting up communication between the two AEs. Only after the connection is associated can DICOM command messages be exchanged between the AEs. AEs use the Association to negotiate which SOP classes can be exchanged and the encoding format for the data. Two AEs establish an association as follows:

- AE#1 sends an Associate Request message to AE#2. AE#1 is known as the requester.
- AE#2 responds by sending an Associate Accept response message to AE#1. AE#2 is known as the acceptor.

Alternatively, AE can reject an Associate Request by sending an Associate Reject message to the requesting AE. the association is gracefully terminated as follows:

- AE#1 sends an Associate Release request message to AE#2.
- AE#2 responds by sending an Associate Release response message to AE#1.

An association can be abnormally terminated by either AE sending an Associate Abort request message. This message is a non-confirmed service, and does not have an associated response. After an association has been established, AEs can exchange DICOM messages, which are formatted as Associate Data messages.

The diagram shows a simple example of an SCU-SCP connection and a series of function calls used to transfer messages and data.

SCU	Description	SCP
Startup	Initialize the DICOM DLL. This must be done before calling any DICOM operation function, but only needs to be called once per application.	Startup

	The SCP listens for incoming connection requests.	Listen
Connect	The SCU calls this function to connect to SCP. This generates a call to this function on the SCP.	OnAccept
OnConnect	The SCP calls CreateNet for each connection it accepts. This provides the SCP memory to store information about this specific connection. Accept function is called to accept the connection. This generates a call to function on the SCU.	Accept
SendAssociate-Request	DICOM Network Connection established. The SCU calls this function to request a DICOM Associate connection. This generates calls to these functions on the SCP	OnReceive OnReceiveAssociateRequest
OnReceive OnReceive-AssociateAccept Or OnReceive OnReceive-	The SCP calls SendAssociateAccept to accept the DICOM Associae connection. This generates calls OnReceive and OnreceiveAssociateAccept on the SCU. SendAssociateReject is sent to reject the	SendAssociateAccept Or SendAssociateReject

AssociateReject	DICOM Associate connection. This generates calls to OnReceive and OnReceiveAssociateReject on the SCU.	

Table 1: The example of an SCU-SCP connection

3.4.4 Message Exchange

DICOM Messages are used to communicate information across the DICOM network. A DICOM Message is a DICOM Command Set, and can be followed by a conditional DICOM Data Set. The Command Set indicates the operations and/or notifications that the requesting AE wishes to have the accepting AE perform on or with the conditional Data Set.

Command Sets are made up of Command Elements, which contain encoded values for each field of the Command Set. The semantics for the Command Set are specified in the DIMSE protocol. Each element in a Command Set is made up of an explicit Tag, a Value Length, and a Value Field. Within the Command Set, Elements are ordered by increasing Tag number. The Tag uniquely identifies an Element, and occurs only once in a Command Set. The requirements for Command Sets and Command Elements are defined in the DIMSE protocol.

DICOM AEs use services that are provided by the DICOM Message Service Element (DIMSE). DIMSE specifies two sets of services: DIMSE-C and DIMSE-N; however, DICOM Client use only DIMSE-C.

DIMSE-C operates with composite SOP Classes and provide effective compatibility with previous versions of the DICOM Standard. In LEADTOOLS, the following are DIMSE-C commands: COMMAND_C_STORE, COMMAND_C_FIND, COMMAND_C_GET, COMMAND_C_MOVE, COMMAND_C_ECHO. Each of them have details are:

- C-STORE This service allows an invoking service-user to request the storage of Composite SOP Instance information by a peer service-user.
- C-FIND This service allows an invoking service-user to match a series of Attribute strings against the Attributes of the set of SOP Instances being handled by a peer service-user. This service will return a list of requested Attributes and their values for each match.
- C-GET This service allows an invoking service-user to fetch information for one or more Composite SOP Instances from a peer service-user. The information returned by the performing service-user is based on the Attributes supplied by the invoking DIMSE-service-user.
- C-MOVE This service allows an invoking service-user to move information for one or more Composite SOP Instances from a peer service-user, to a third party service-user. The information moved is based on the Attributes supplied by the invoking service-user
- C-ECHO This service allows an invoking service-user to verify end-to-end communications with a peer service-user.

3.4.5 Creating Database

User must create an ODBC DataSource. The name of this datasource should be "DSN=DICOMSERVERDB;UID=sa" and point that datasource to selected MDB file.

Then user has to copy file, LEADDICM.MDB, to working directory. These tables below show the designing of each field in database file. The tables 2-5 show how to create patient, studies, series and image fields, in database file.

Patients:

PatientName	Text	200
PatientID	Text	200
PatientBirthDate	Date/Time	8
PatientBirthTime	Date/Time	8
PatientSex	Text	10
PatientEthnicGroup	Text	50
PatientComments	Memo	Undifined
ReferenceDir	Text	200
SaveImage	Integer	2

Table 2: Patient field

Studies:

StudyInstanceUID	Text	200
AccessionNumber	Text	50
ReferencedFile	Memo	Undefined
PatientID	Text	200
StudyID	Text	200

StudyDescription	Memo	Undifined
StudyDate	Date/Time	8

Table 3: Studies field

Series:

StudyInstanceUID	Text	200
Modality	Text	50
SeriesInstanceUID	Text	200
ReferencedFile	Memo	Undefined

Table 4: Series field

Images:

SOPInstanceUID	Text	200
StudyInstanceUID	Text	200
SeriesInstanceUID	Text	200
ImageNumber	Integer	2
ReferencedFile	Memo	Undefined

Table 5: Image field

3.5 Software Test

A simulation of physical using LAN connection by eFilm and PiView is used to test the DICOM communication. The procedure of this test is that one computer run DICOM server software (eFilm or PiView) and another run the developed software

(DICOM Client software). In software test, the developed software is tested with both eFilm server and PiView server.

Firstly, to open DICOM file, user uses “open file” command to open DICOM file from local hard disk. The result of the command the developed software should display DICOM information and image.

Secondly, before testing DICOM connection user has to set the DICOM server destination by initializing value in a new connection dialog (e.g. server’s AE title = PIVIEW, client’s AE title = Dicomc, server’s IP address = 10.17.11.1 and port number = 104). After setting their information. User can connect with DICOM server by using “connect” command. The results of this command the developed software sends “associate request” to server. After that it looked forward to response from DICOM server.

Thirdly, testing storage service class, user has to use “send file” command to send DICOM file from local hard disk into DICOM server. If the sent DICOM files appear at DICOM server, this function works.

Fourthly, for query service class, user has to test both “query patient root” and “query study root” commands. After using this command, the developed software should receive the DICOM information resident on the server.

Finally, for retrieving service class, it can be tested in two ways. First, the DICOM server sends DICOM files into the developed software to save these files in database of the developed software, and the another way user uses the “retrieve” command from the developed software to save DICOM files, which are queried by “query patient root” command.

3.7 Summary

The DICOM software is created on desktop computer on the Windows 95/98 operating system. The Microsoft Visual C++ and Microsoft Access 97 are chosen for developing the software. Functions (e.g. displaying, querying or retrieving) are derived from LEADTOOLS' source codes. A TCP/IP network connection is essential for testing the developed software. Lastly the developed software was tested at the Siriraj Hospital.

CHAPTER IV

RESULTS

This chapter describes the testing results the developed software or "DICOM Client" described in section 3.5. All information in chapter 4.1- 4.5 comes from testing DICOM Client with PiView server. In section 4.6 describes the testing result of testing DICOM Client with eFilm server.

4.1 Testing open DICOM file

For testing the open DICOM file, when opening file with the developed software, it displays data elements such as patient's name, patient's sex or patient birthday in the left-hand side (Patient's IOD Window) and its information on the right one (Patient's Information Window). In addition, selecting image pixel can show image information. These are shown in figure 16 and 17.

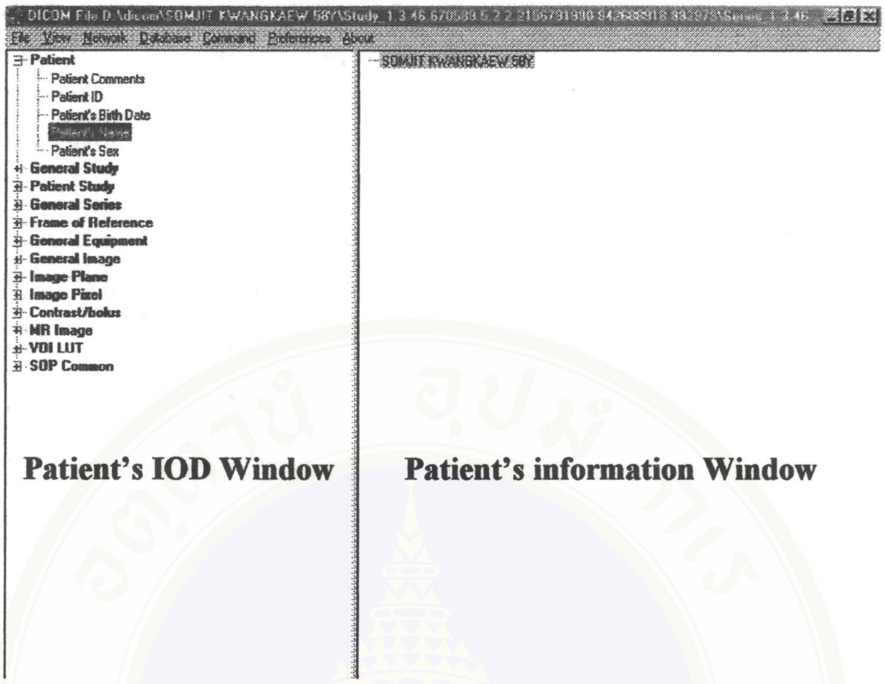


Figure 16: The result of displaying patient's name

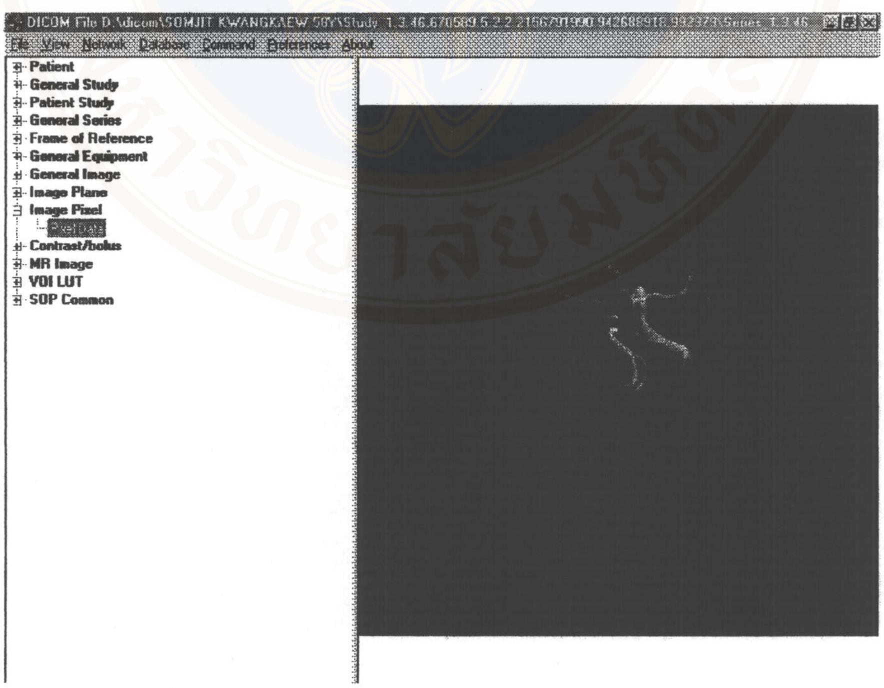


Figure 17: The result of displaying image pixel

4.2 Connection Testing

According to connection testing, user needs to set essential values in connection dialog before making a connection. Figure 18 shows the set values in order to connect with PiView server. After that, DICOM Client sends the "associate request" to PiView and waits for responding from server. The result is this DICOM server sends the "associate accept" back to the developed software in order to confirm that server software and client software are linked as shown in figure 19 at Messages Window.

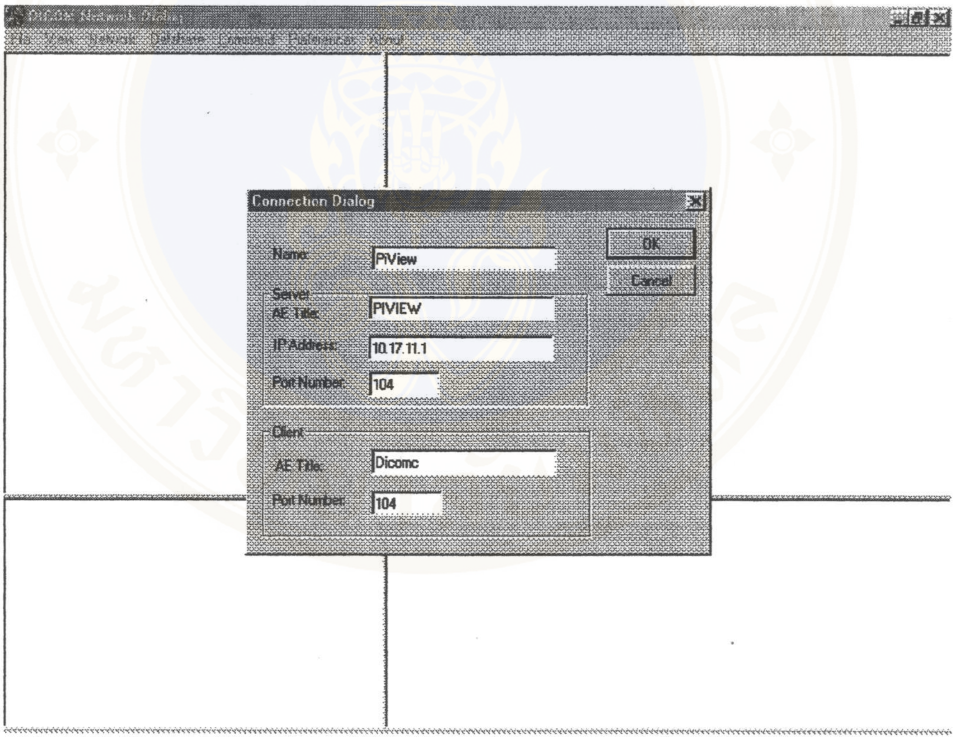


Figure 18: show setting server and client connection

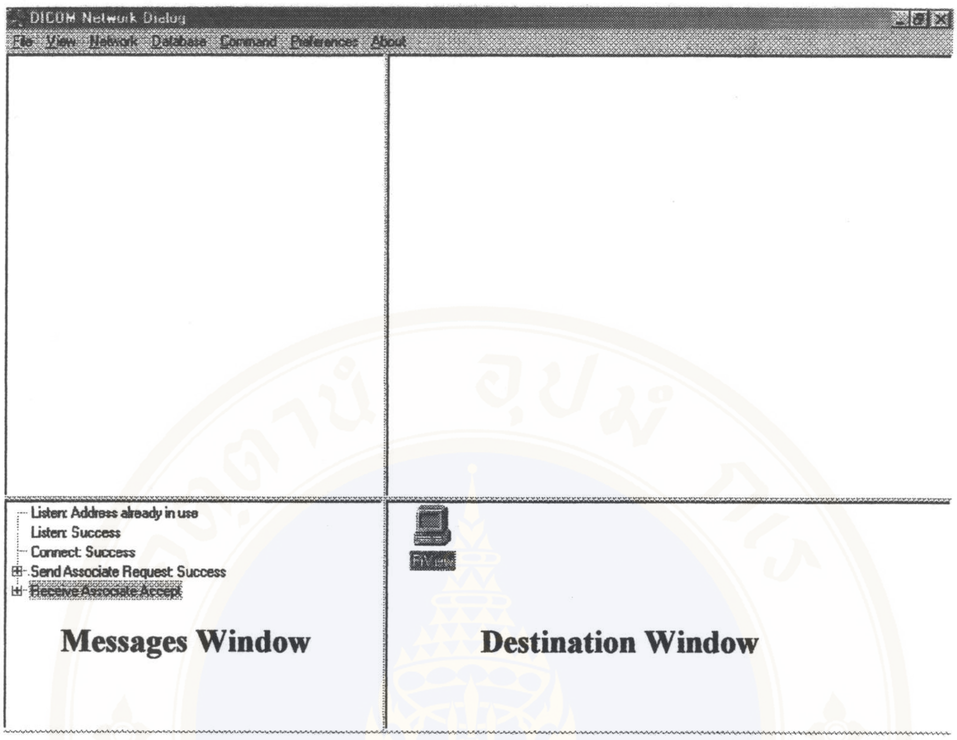


Figure 19: Messages exchanged between client and server

4.3 Storage service class testing

The result of storage service class testing is that DICOM server can store DICOM file sent from the developed software. The DICOM Client gets C-store message from server as shown in figure 20 when the storage service class is successful.

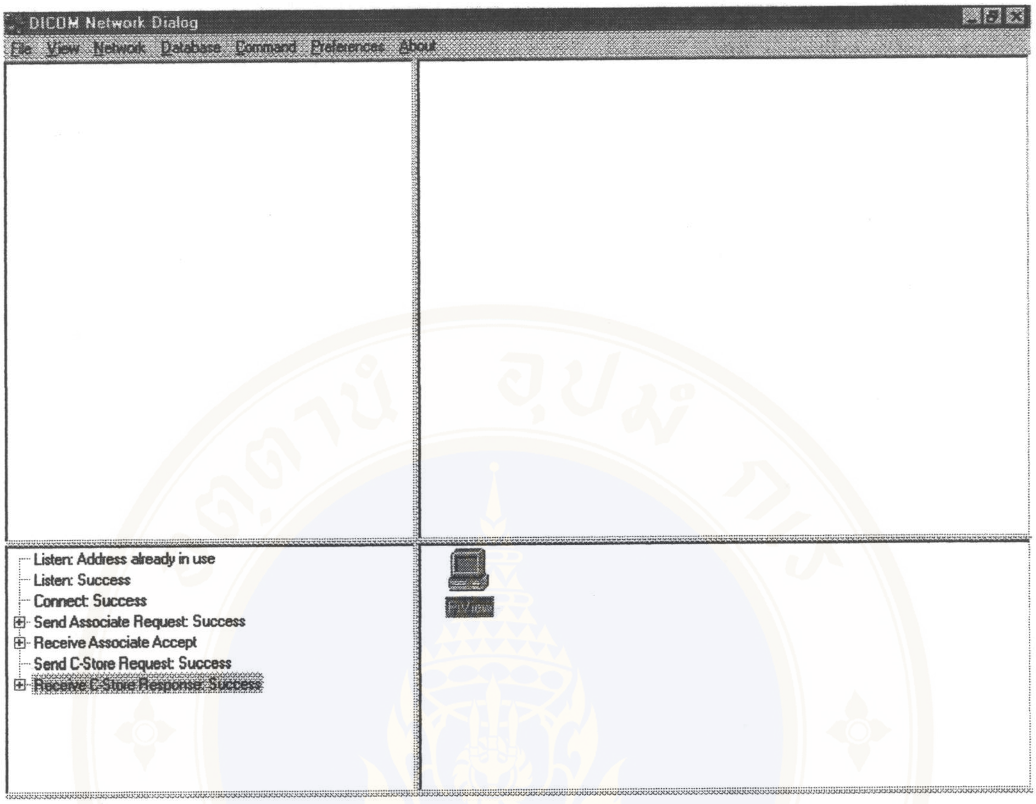


Figure 20: Show messages exchanged after using "send file" command

4.4 Query service class testing

The DICOM Client supports query service classes both patient root query and study root query. When using "query patient root" command the DICOM Client displays patient, study, series and image folder, whereas the result of "query study root" command displays study, series and image folder. These are shown in figure 21 and 22.

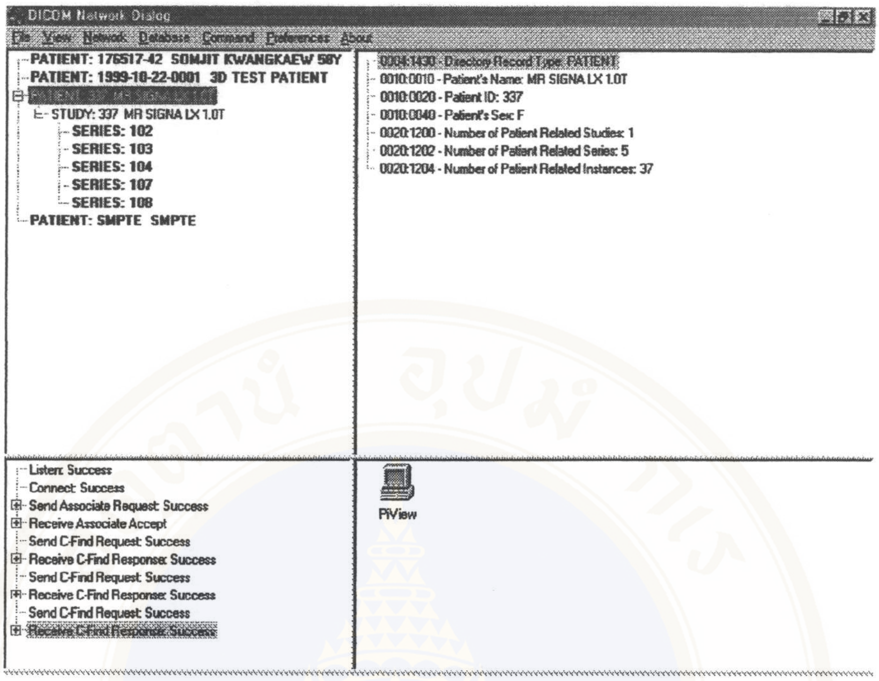


Figure 21: The result of querying patient root

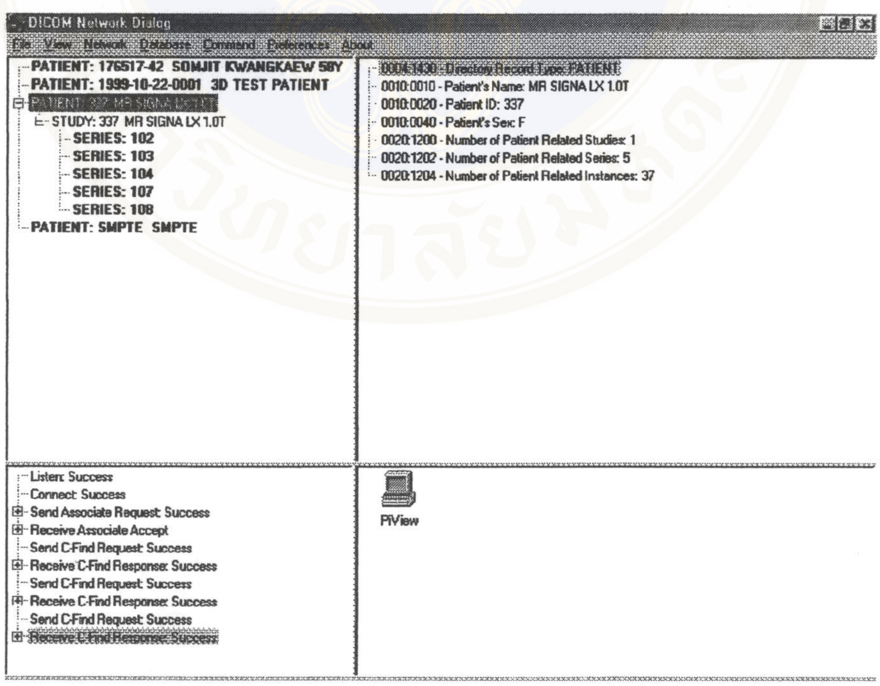


Figure 22: The result of querying study root

4.5 Retrieve testing

For retrieving service class, the result of testing is the DICOM Client can support this service class by retrieving C-Get message response from PiView, and patient information sent from server shown in the Patient's IOD Window. In figure 23, the information, of a patient namely "SOMJIT KWANGKAEW" in which sent from server, is displayed on the DICOM Client.

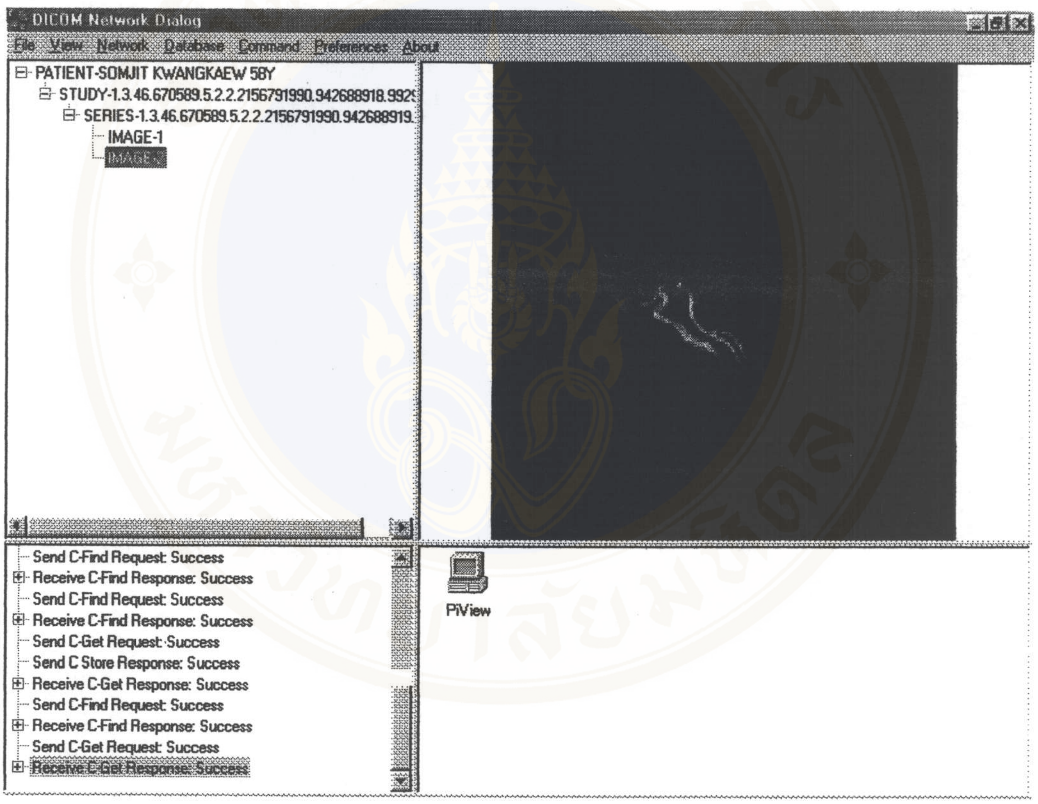


Figure 23: The result of receiving DICOM data

4.6 Testing DICOM Client with eFilm server

The results are the same as testing DICOM Client with PiView except the "query patient root" and "retrieve" commands are not effective. The DICOM Client does not receive information back after using these two commands.

4.7 Summary

In summary the results of testing the DICOM Client with PiView server, this software supports displaying DICOM file, querying, retrieving and storing service class. Whereas the testing result the DICOM Client with eFilm server, it cannot use retrieve function and patient root query function.

CHAPTER V

DISCUSSION

This chapter is a discussion the results of testing the developed software. The discussion topics focus on three aspects: user interface, software implementation, and software testing.

5.1 User interface

Visual C++, which is used to create user interface such as menu bar, command buttons, and text boxes, cannot make user-friendly interface because the command structure of Visual C++ involving graphic is complex and infamous for user interface implementation.

5.2 Software implementation

Even thought source code from LEADTOOLS software makes the development easier, these codes affect the software development. User cannot improve program efficiency because the source code modification is not allowed from LEADTOOLS.

To use this software, user must include the special files (see appendix 2) into window system folder. These files are associated with the LEADTOOLS functions.

5.3 Software testing

The software testing with PiView server has results are that they support storage, query and retrieve service class as well as display DICOM file. Whereas, for testing with eFilm server, it has the different results, the developed software unsupported not only the specific querying, such as patient's name, patient identification, or patient's sex but also the "query patient root" and "retrieve" command. From problem analysis, the mistakes come from both eFilm server and LEADTOOLS codes. When examining the data transferring between eFilm and PiView, eFilm cannot give any information with PiView by using specific querying. This problem occurs from eFilm software. Similarly, querying patient root, eFilm server misunderstands with the requested data pattern of the "query patient root" command. Lastly, LEADTOOLS's codes unsupported the "retrieve" command after querying study root because these codes about retrieving function are incorrect.

5.4 Summary

In conclusion, although Visual C++ has efficiency for coding program, it is not appropriate for making user interface. The development this software is quite difficult because of the limitation of LEADTOOLS software, which does not allow modifying the main part of its function. Moreover, some error parts of these codes are the problems with DICOM server and the developed software.

CHAPTER VI

CONCLUSION

6.1 Conclusion

Nowadays, all digital medical devices use DICOM standard for communication between each other. DICOM standard provides capabilities such as network image transfers, online imaging study management, network print management, and open media interchange. However, the commercial DICOM software is expensive. It is impossible that every hospital buy DICOM software. This research is to design and develop DICOM software, which is not licensed software. In addition, Microsoft Visual C++ is applied to this software in order to computerize data class library for the DICOM standard that supports DICOM image management and transfer over a network. Methodology of this thesis is divided into 6 steps as the following: data investigation, system design, software design, implementation, software test, and system test.

The system design is a step for determining specification of DICOM software that is application design and database design. The software design step is a development detail of each function, which is defined from system design step. Microsoft Visual C++ is selected for doing this software, and source code is provided from LEADTOOLS. The LEADTOOLS Class Library is used to implement a set of standard C++ objects. These objects make use of the LEADTOOLS API, which are standard Window Dynamic Link Libraries.

The developed software is tested any functions via LAN connection with DICOM server software, eFilm and PiView before it is tested with DICOM network system by using PiView as DICOM server at the Siriraj Hospital.

The result of developing this software is application program prototype, which has the following functions:

- Create and initialize a DICOM Network Connection
- Create a DICOM Associate Connection to allow the transfer of data and messages
- Retrieve information about the DICOM Network Connection
- Retrieve information about the DICOM Associate Connection
- Support the DICOM Storage Service Class
- Support the DICOM Retrieve Service Class
- Support the DICOM Query Service Class – only SCU
- Load, Save and Display the DICOM file

DICOM Client software applies source code from LEADTOOLS, so the modification its algorithm is not allowed. Any functions have to process along step of LEADTOOLS algorithm, and some parts of source code are wrong, therefore it causes that the retrieve function do not work when users call "Find study root".

In addition, Efilm DICOM software does not support a specific query patient root of DICOM Client software.

6.2 Recommendation

1. Although this application is used for DICOM server at the Siriraj hospital. It can be applied to other hospitals, which have DICOM servers.
2. In future version, it should be embedded into Service Class Provider (SCP).
3. Database file structure should be improved by using capability of Microsoft Access.
4. DICOM Client does not cover all DICOM service. Nevertheless, it should include printing service class in future version.
5. Image processing function should be integrated in this software because, in the practical case, clinician uses image process at work.
6. Hospital staff needs to be trained to use this software before using in practical events.

REFERENCES

1. UNCTAD Trade Point Development Center Overview. [Online]. 1999. Available from: http://www.gse.rmit.edu.au/untpdc/conference/interact /presentation/SGI/ [Accessed 2000 August 2].
2. Eastman Kodak Company. KODAK. Health Imaging Learning Center-DICOM History. [Online]. 1994-2000. Available from: http: //www.kodak.com/US/en/health/learning/dicom/history.shtml [Accessed 2000 April 25].
3. ACR/NEMA Standard, Digital Imaging and Communications in Medicine (DICOM). 1998.
4. LEAD Technologies, Inc. Basic DICOM File Structure. [Online]. LEADTOOLS; 1998. Available from: http://www.leadtools.com/home2/DICOM /ltdc1.htm [Accessed 2000 August 20].
5. LEAD Technologies, Inc. DICOM Data Element Structure. [Online]. LEADTOOLS; 1998. Available from: http://www.leadtools.com /home2/DICOM/ltdc5.htm [Accessed 2000 August 20].
6. DICOM Cook Book for Implementations in Modalities chapter 1&2 version 1.1. 1997. Copyright Philips Medical System Nederland B.V. 1996,1997.
7. Dirk k., Kerstin F. Object-Oriented Implementation of a DICOM Network Client in Smalltalk. [Online] 1999; Available from: http://www.ieee.org [Accessed 2000 August 20].

8. Hua-lei Z, Steven L., Steve R. Class library design for DICOM in C++ [Online] 1997; Available from: http://www.ieee.org [Accessed 2000 August 20].

9. DESACC Inc. Integrating Your Computer into Your Hospital's DICOM Network. [Online]. Desacc: 1999. Available from: http://www.desacc.com /dicom/dicomworkflow/example1/example1.html [Accessed 2000 April 24].

10. John J. Barton, Lee R. Nackman, Scientific and Engineering C++ an introduction with advanced techniques and examples 1994.

11. LEAD Technologies, Inc. DICOM Data Element Fields. [Online]. LEADTOOLS; 1998. Available from: http://www.leadtools.com/home2/DICOM /ltdc3.htm [Accessed 2000 August 20].

12. LEAD Technologies, Inc. DICOM Value Representations. [Online]. LEADTOOLS; 1998. Available from: http://www.leadtools.com /home2/DICOM /ltdc4.htm [Accessed 2000 August 20].

APPENDIX I

User guide

1. Starting the DICOM Client

When user clicks at the DICOM Client icon, it shows main window and its details as shown in figure 24.

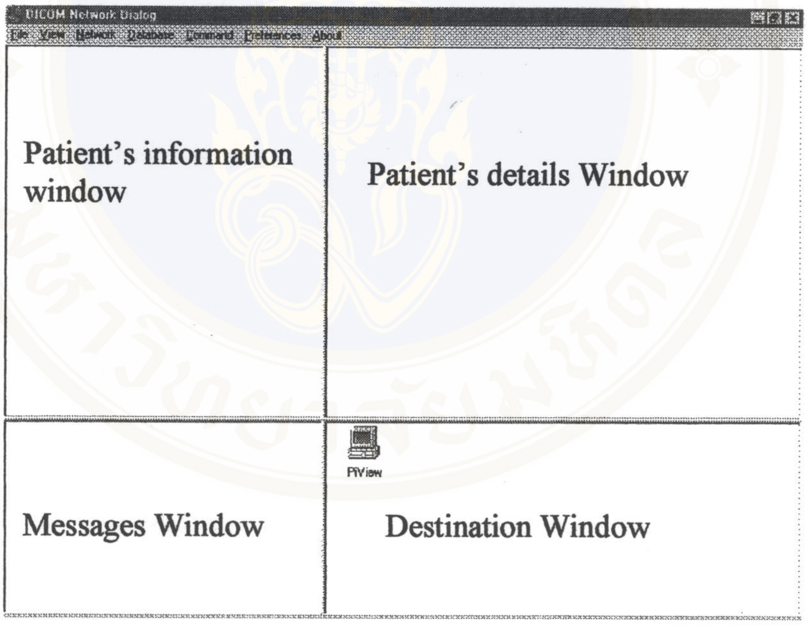


Figure 24: Main window.

2. Exiting the DICOM Client

User can exit from the DICOM Client by clicking at close button or at exit menu as shown in figure 25.

Figure 25: Exiting from DICOM Client

3. Open DICOM file

DICOM file can be opened along these steps which shown in figure 26.

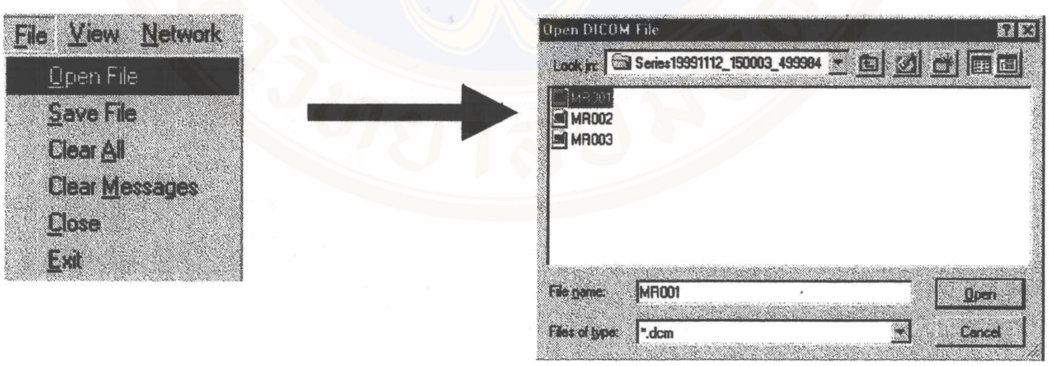


Figure 26: Steps of opening DICOM file.

4. Save image as

User can change name of DICOM file after open it. The processes are shown in figure 27.

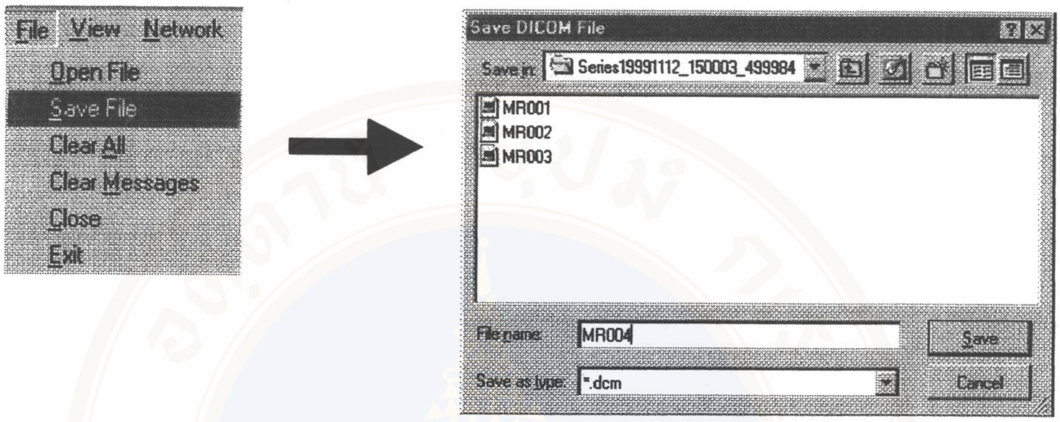


Figure 27: Steps of saving image

5. Close DICOM file

DICOM file can be closed which shows in figure 28.

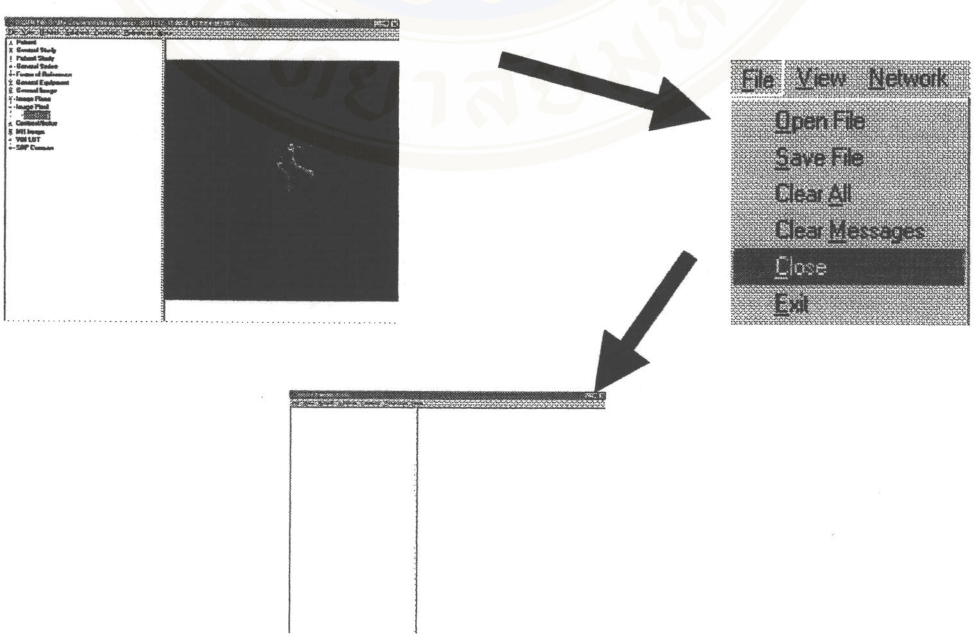


Figure 28: Steps of closing

6. Clear All Windows

Clear all windows has the same function as close, but it only clear data from all windows except destination window.

7. Clear Messages

User can clear only a message window from this command.

8. Show image in normal size or fit to window

Image can be expressed either normal size or fit to window that depending on selecting of user. The default of show image is fit to window. The results of commands are shown in figure 29 and 30.

Figure 29: Showing image in fit to window mode

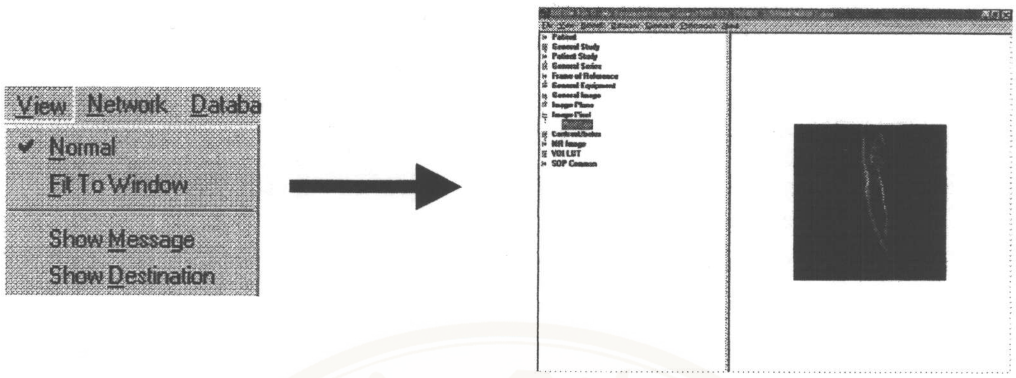


Figure 30: Showing image in normal mode

9. Show Message and Destination Window

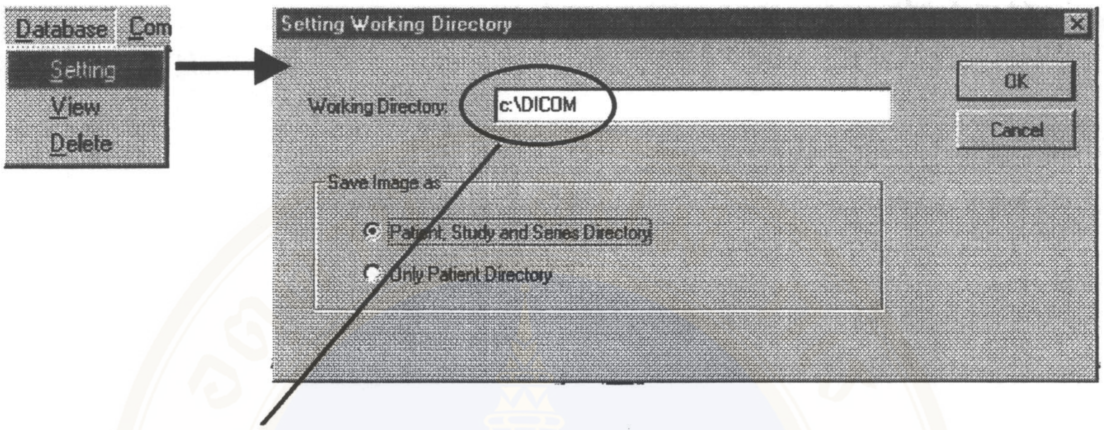
From main window, user can close or show either message window or destination window.

10. Database management

DICOM files, which receive from retrieving, will be kept in database directories, and user can mange type of saving image as well.

10.1 Setting working database directory and saving image type

In figure 31, it shows how to setting working database directory.



Typing working directory

Figure 31: The steps of setting working database directory

DICOM Client allows user to specify where incoming files will be stored. It is able to extract information from each incoming DICOM file in order to store it hierarchically to a local volume. Parameters, which can be extracted, are patient name, study UID, series UID and image number. In another way, incoming DICOM file will be store only in patient directory. In figure 32 show how to set an address of incoming DICOM file and figure 33 and 34 show the difference of saving image in two methods.

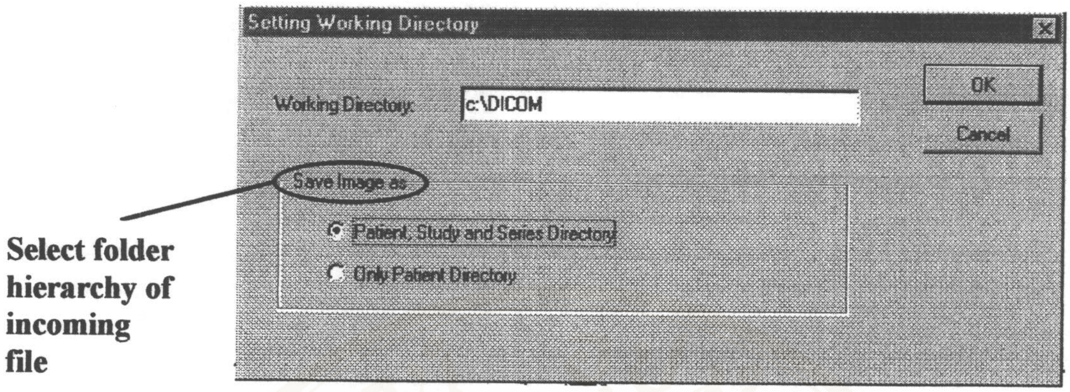


Figure 32: The steps of selecting folder hierarchy

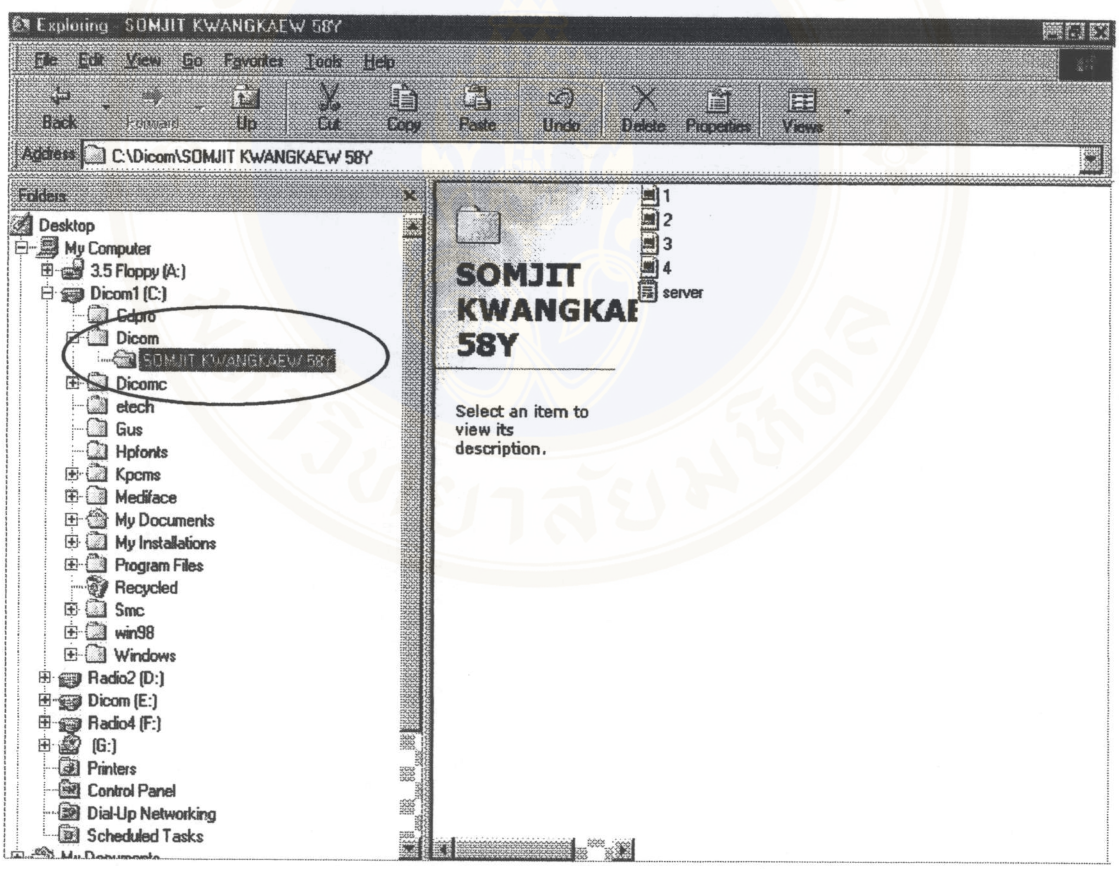


Figure 33: Saving image in only patient folder

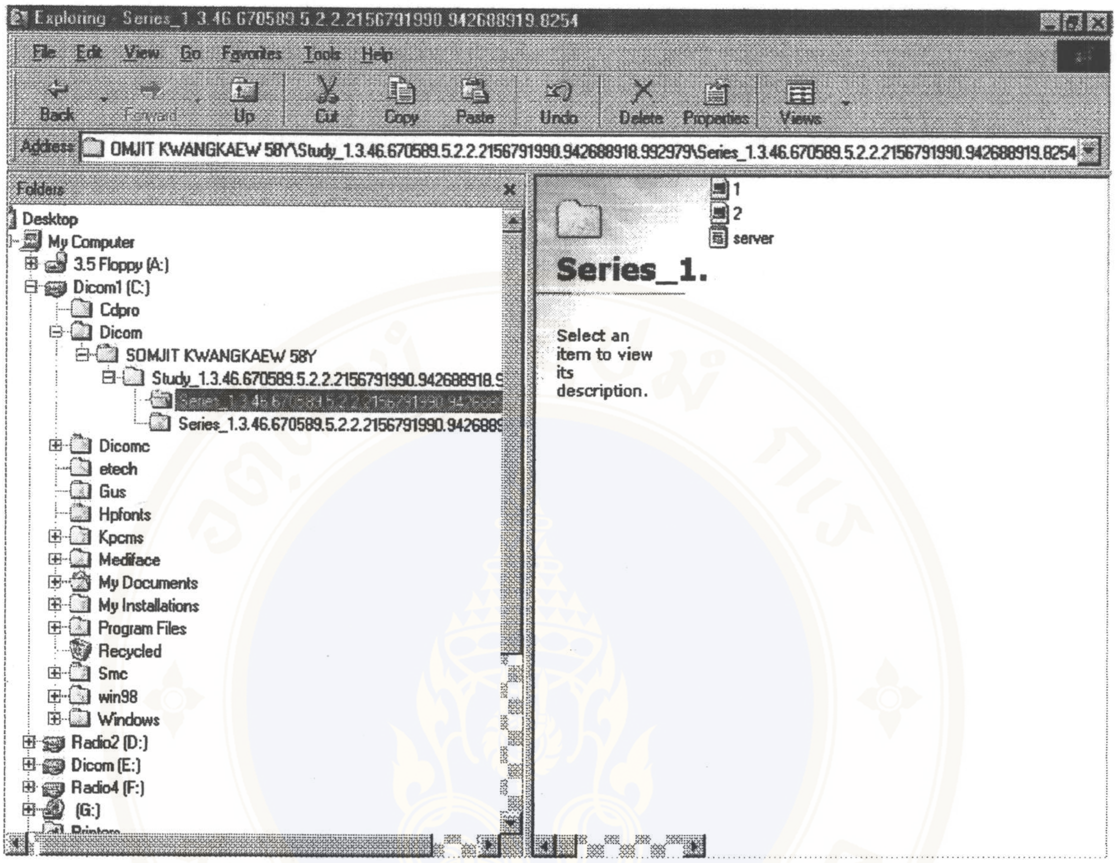


Figure 34: Saving image in patient, study and series folder

10.2 View information in database folder

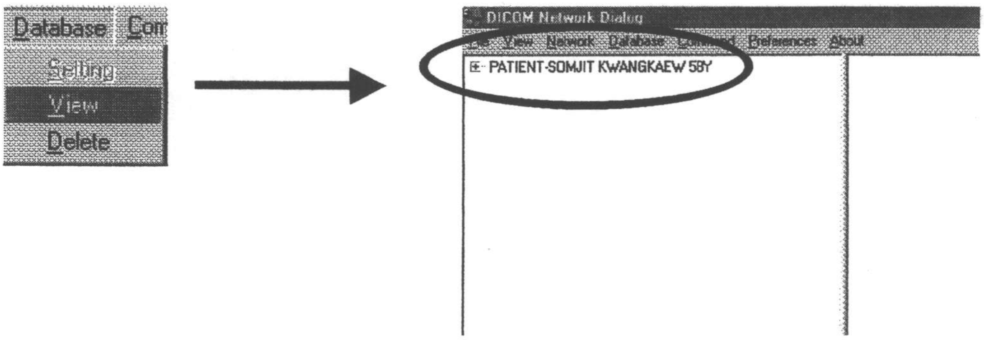


Figure 35: Displaying information in database folder

When user has to display information in database folder, making along the figure 35.

10.3 Delete information in database folder

User can delete unimportant folder by doing along the steps in figure 36.

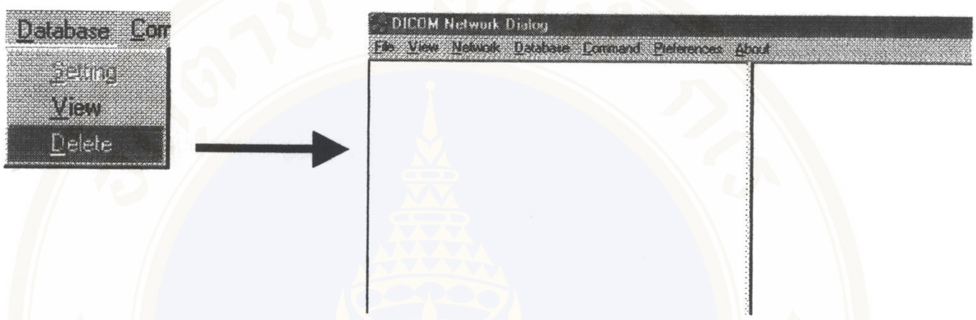


Figure 36: Show the result of using deletes function

11. DICOM Network

11.1 Insert a new connection

Before starting DICOM network, user has to insert a connection before as shown in figure 37.

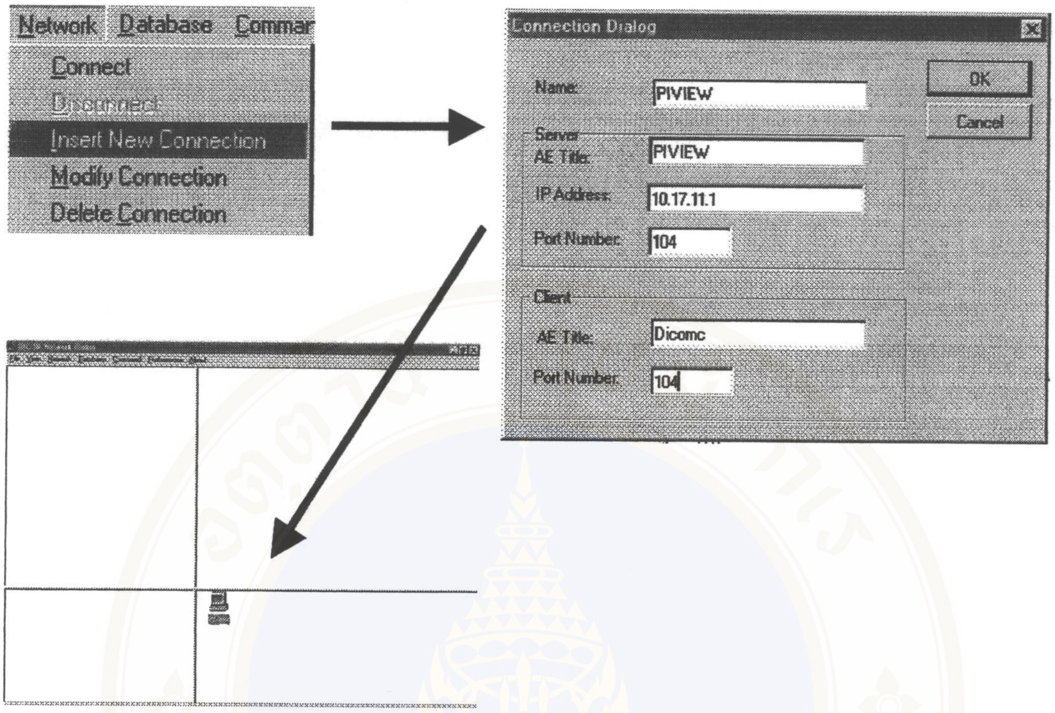


Figure 37: The steps of setting a connection

11.2 Modify connection

User is able to modify parameters within connection dialog by doing along the step shown in figure 38.

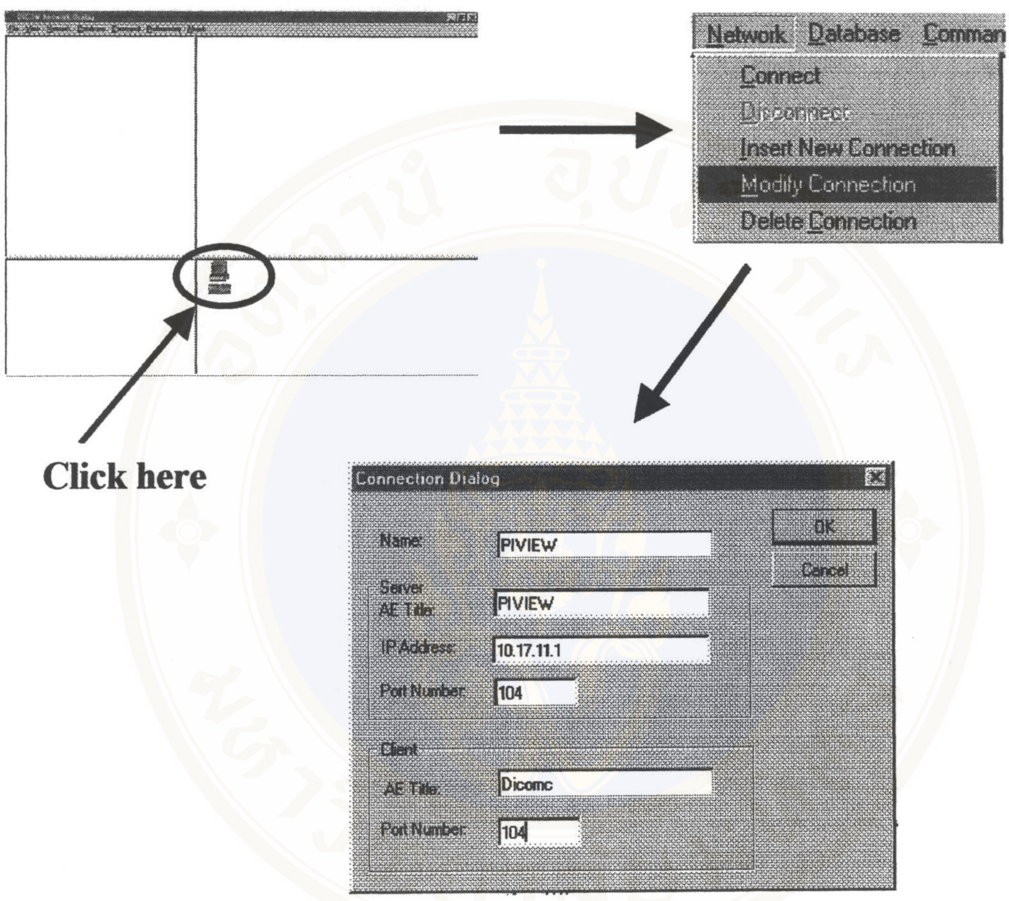


Figure 38: The methods of modify connection

11.3 Delete connection

Destination can be deleted from delete connection function which shown in figure 39.

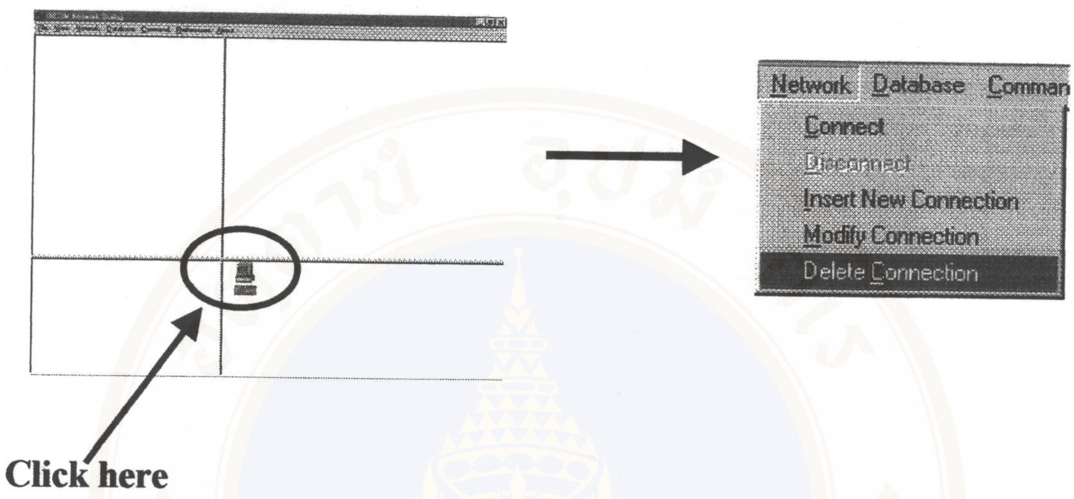


Figure 39: The steps of delete connection

11.4 Connect to DICOM network

When requiring to connecting with DICOM network, there are two types of connecting as shown in figure 40.

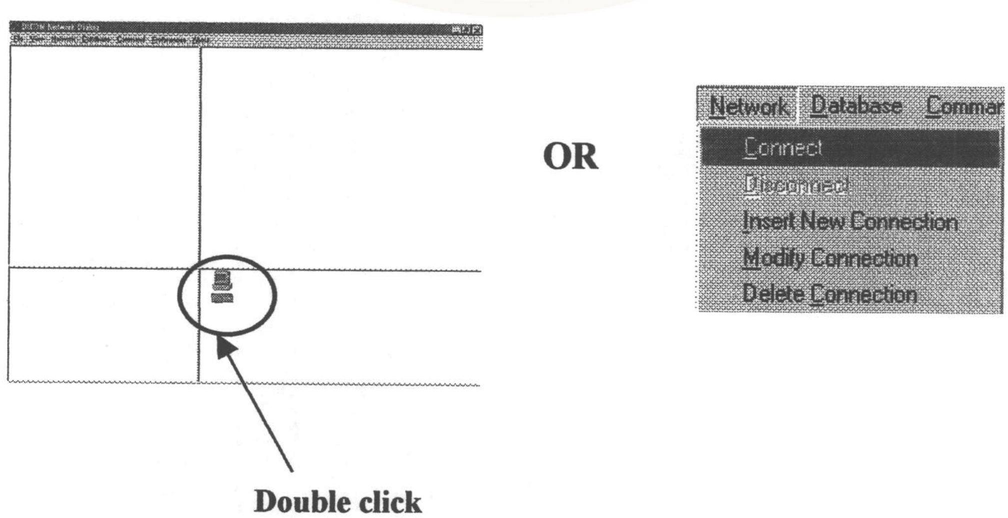


Figure 40: Show two methods of connection with DICOM network

11.5 Disconnection from DICOM network

Figure 41 shows how to disconnection from DICOM network.

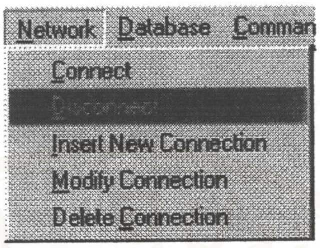


Figure 41: express the step of disconnects

12. Echo command

This command is used to checking status of network that still online or not. If online, program will show that receives echo response. As shown in figure 42

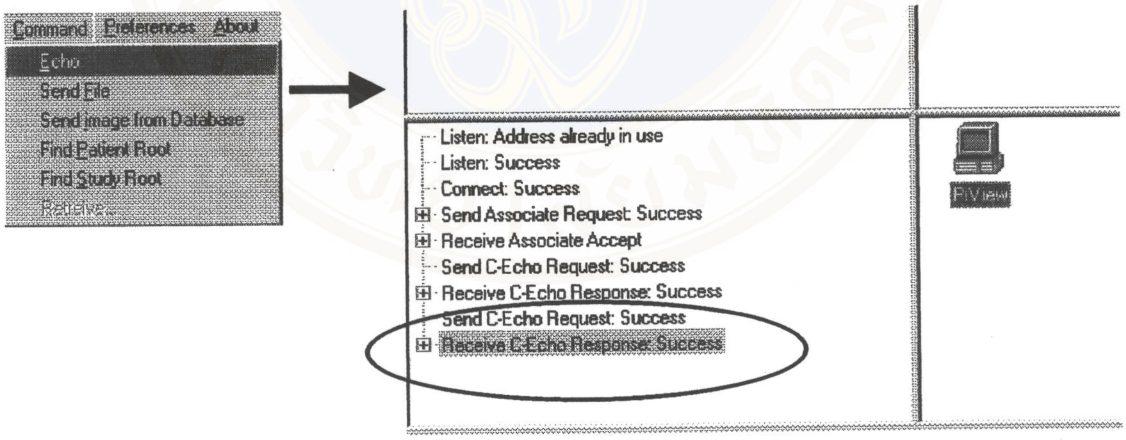


Figure 42: Show the result of using echo function

13. Send DICOM file to server

User can select a wanted file and send it to server by using this function. The steps of sending file are shown in figure 43.

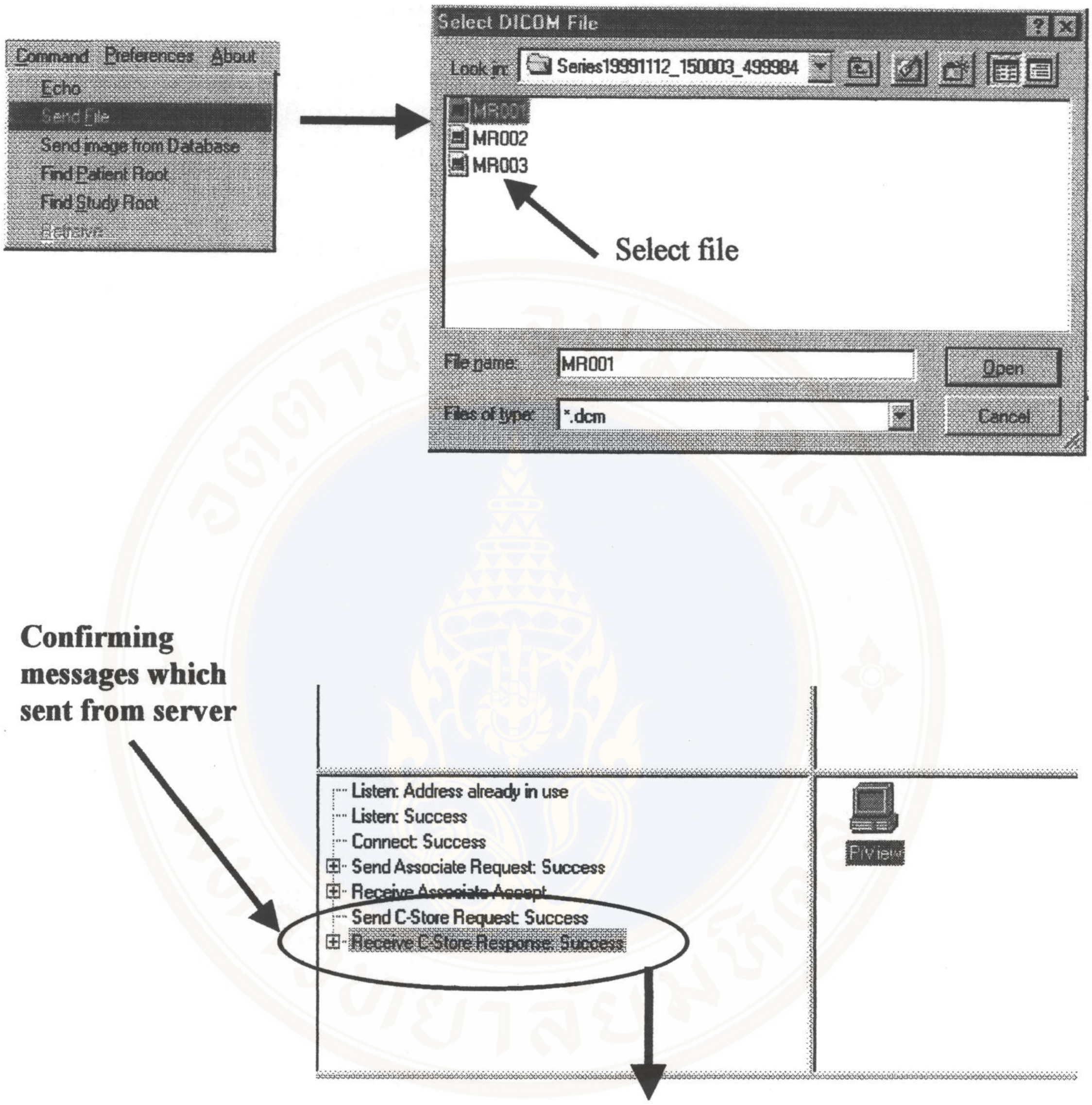


Figure 43: Steps of sending file to server

14. Send image from Database

User is able to select image file from database and sends it to server by this function as show the processes in figure 44.

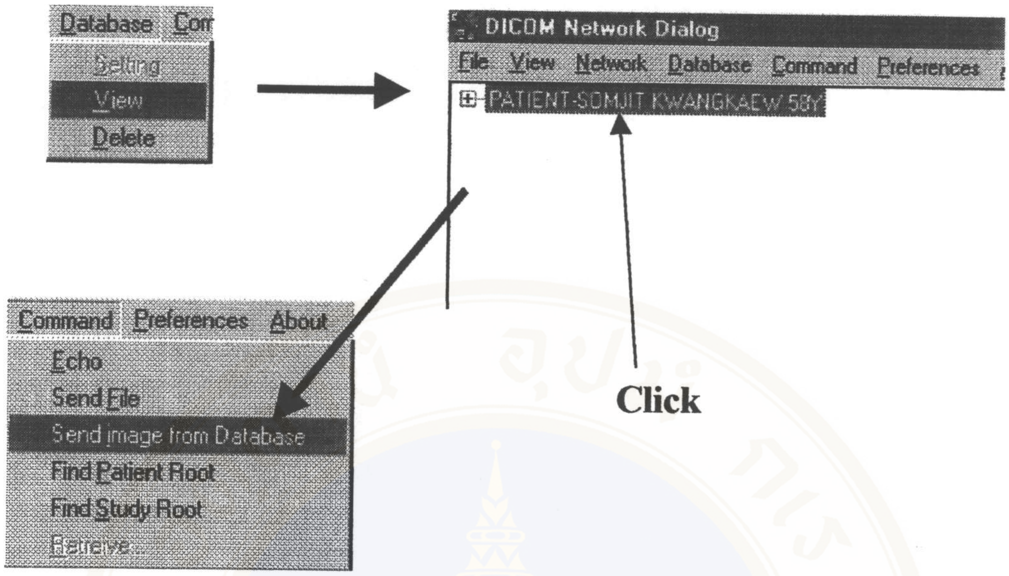


Figure 44: Steps of sending image from Database to server

15. Query Patient Root

DICOM Client program supports query service class, which in figure 45 shows the query patient root.

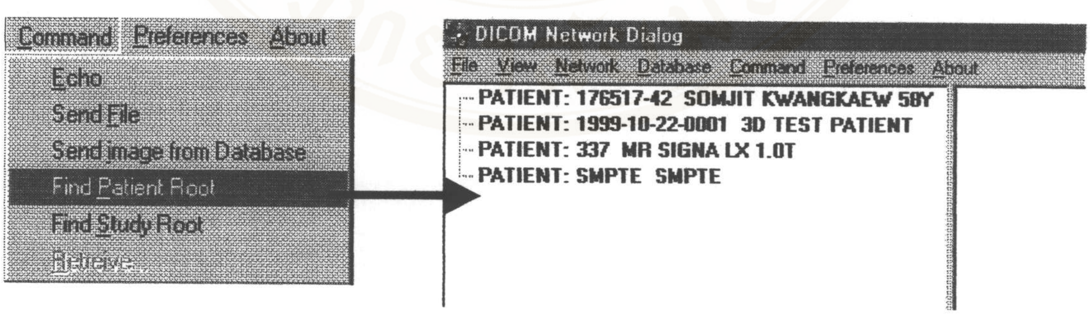


Figure 45: the result of query patient root

16. Query Study Root

In figure 46, the expression results of query study root.

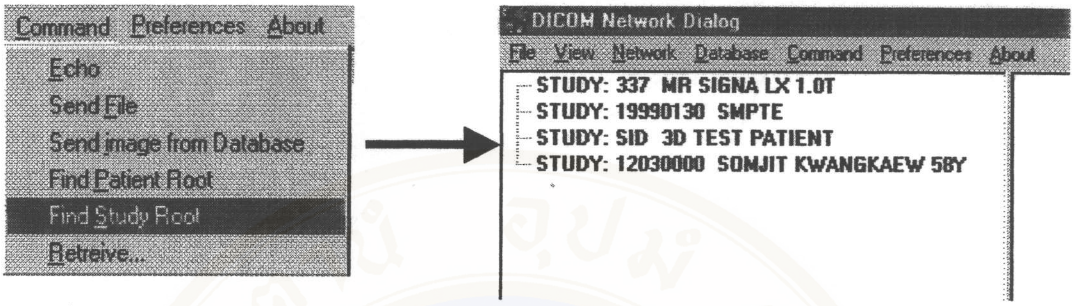


Figure 46: Show the result of query study root

17. Retrieve service class

The last command of this program is retrieve function. The steps of retrieving image from server as shown in figure 47.

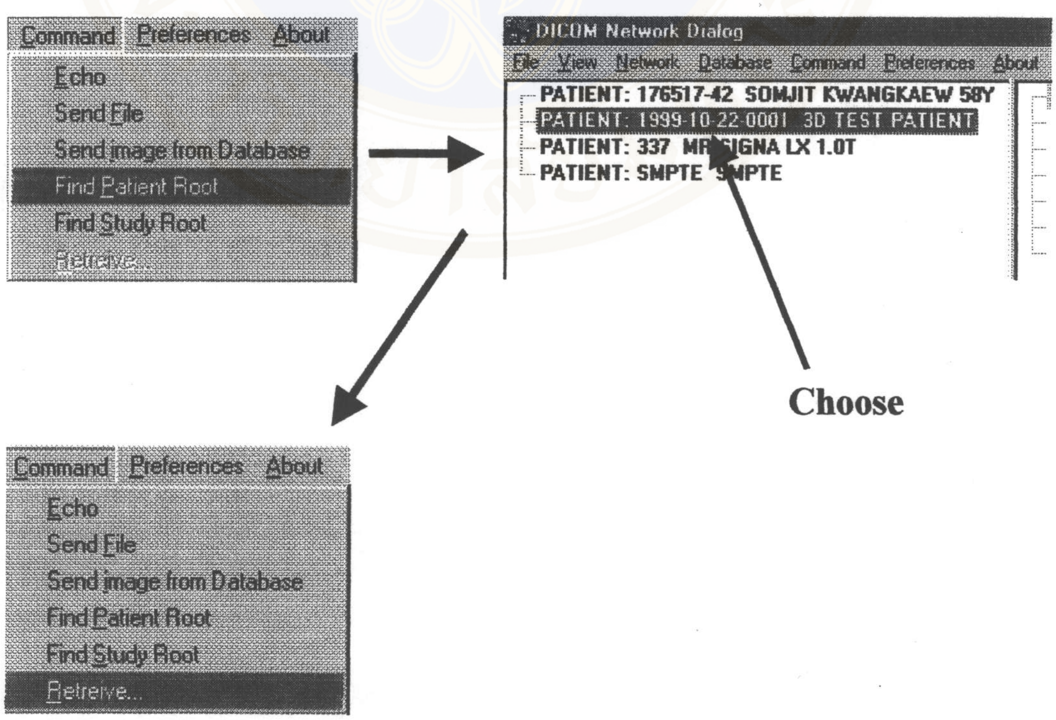


Figure 47: Steps of retrieving image

18. Unique Identifiers (UID)

UIDs have been defined by several of the DICOM working groups. This software allows modify, insert and delete a DICOM UID. UID dialog is shown in figure 48.

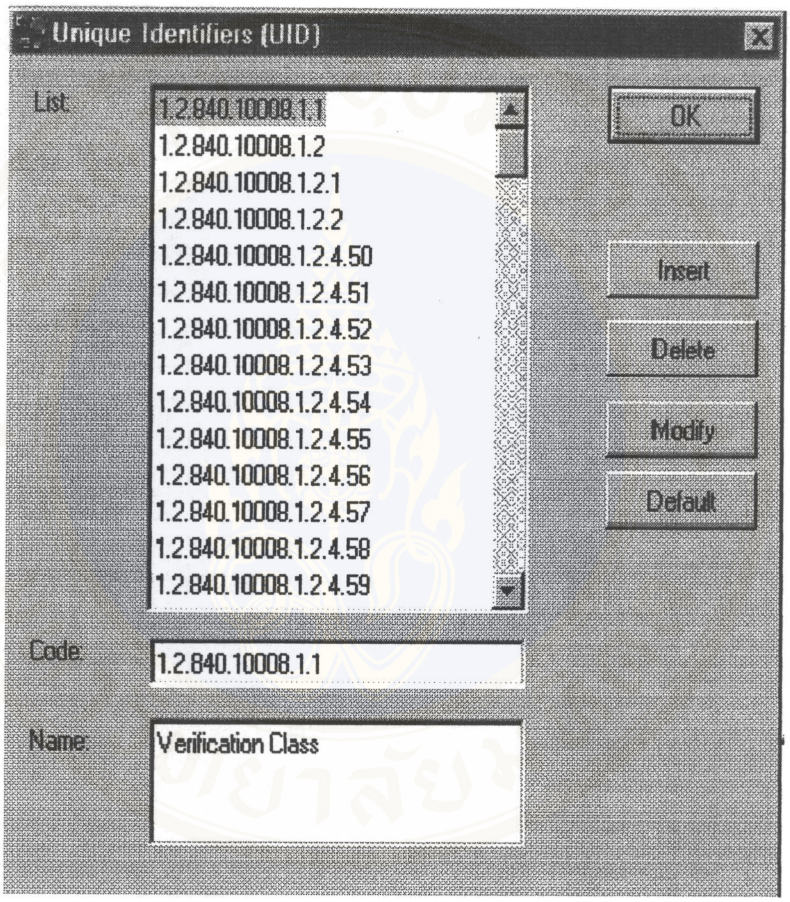


Figure 48: UID dialog

19. Association

User can modify transfer syntax from association dialog shown in figure 49.

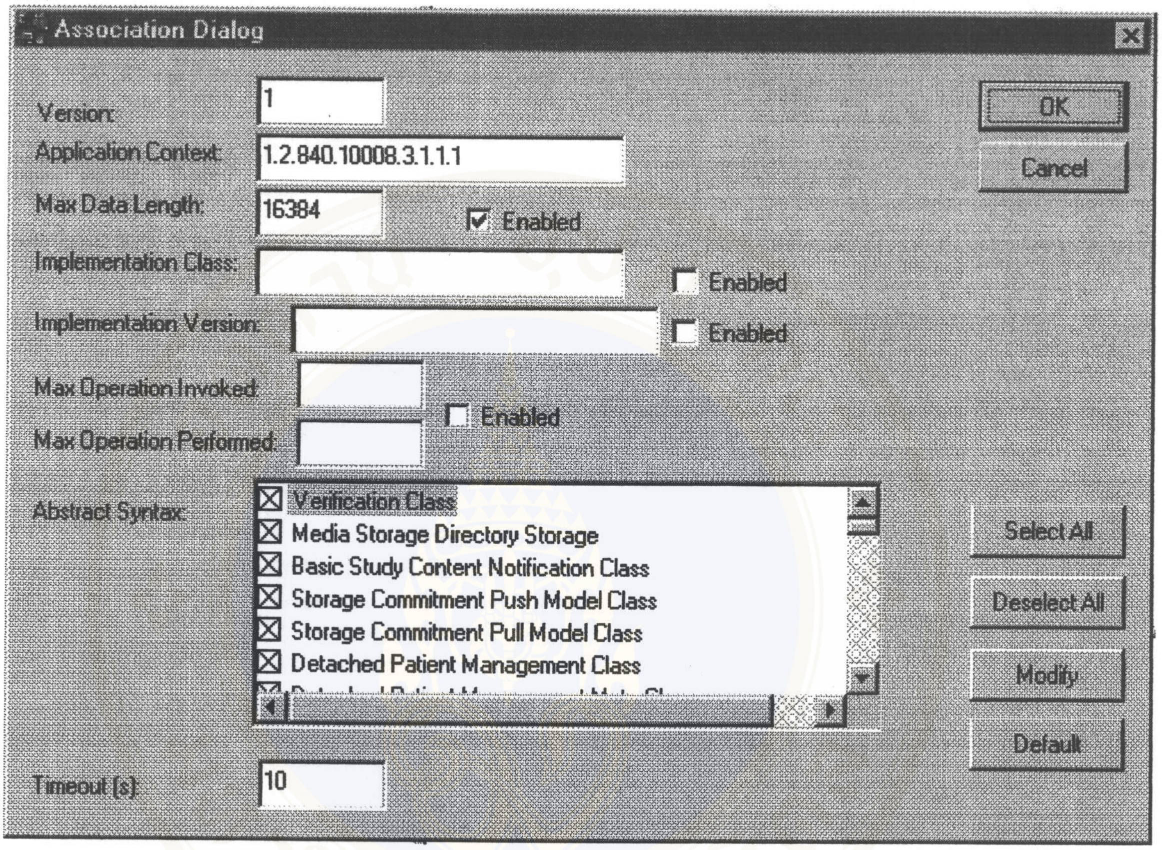


Figure 49: Association dialogs

APPENDIX II

File to be included with application

File	Explanation
LTKRN12N.DLL	LEADTOOLS kernel, required for all applications. The header file is LTKRN.H. The library file is LTKRN_N.LIB or LTKRN_W.LIB. This DLL is required for all other DLLs.
LTDIS12N.DLL	Display support, required for most applications. The header file is LTDIS.H. The library file is LTDIS_N.LIB or LTDIS_W.LIB.
LTWVC12N.DLL	LTWVC12N.DLL is the LEADTOOLS Class Library DLL used with Microsoft Compilers. The library file is LTWVC_N.LIB.
LTFIL12N.DLL	File read and write support. The header file is LTFIL.H. The library file is LTFIL_N.LIB or LTFIL_W.LIB.
LTDLG12N.DLL	Imaging Common Dialog support. The header file is LTDLG.H.
LFDIC12N.DLL	DICOM file support. Note: LFDIC12N.DLL is no longer shipped. All DICOM support is provided in the LTDIC12N.DLL file.

BIOGRAPHY

NAME	Mr. Pakawat Nilapong
DATE OF BIRTH	November 11, 1977
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Mahidol University, 1996-1999:
	Bachelor of Engineering (Electrical)
	Mahidol University, 1999-2001:
	Master of Engineering (Biomedical Engineering)