ซอฟต์แวร์ DICOM เกตเวย์

DICOM Gateway Software

โดย

นาย สุรณรงค์ คามตะศิลา

วัน เดือน ปี 17 ก.ค. 2541
เลขทะเบียน 039036
เลขเรียกหนังสือ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

039036

ซอฟต์แวร์ DICOM เกตเวย์

DICOM Gateway Software

โดย

นาย สุรณรงค์ คามตะศิลา 38013084

อาจารย์ที่ปรึกษา

ผศ.ดร.สุวิพล สิทธิชีวภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2540

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ซอฟต์แวร์ DICOM เกตเวย์**

DICOM Gateway Software

ผู้จัดทำ

นายสุรณรงค์ คามตะศิลา 38013084

อาจารย์ที่ปรึกษา

(ผศ.ดร.สุวิพล สิทธิชีวภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์ DICOM เกตเวย์

DICOM Gateway Software

โดย นายสุรณรงค์ คามตะศิลา

อาจารย์ที่ปรึกษา ผศ.ดร.สุวิพล สิทธิชีวภาค

บทคัดย่อ

ปัญหาจากความหลากหลายในการจัดเก็บภาพทางการแพทย์อันเนื่องมาจากเครื่องมือแพทย์ที่ต่างผู้ผลิตกัน ก่อให้เกิดปัญหาอย่างมากในระบบไทรเวช (Telemedicine) ซึ่งจำเป็นต้องมีการแลกเปลี่ยนข้อมูลกันระหว่างโรงพยาบาล

โครงงานนี้นำเสนอวิธีการนำข้อมูลภาพทางการแพทย์ที่ได้จากเครื่องมือแพทย์เข้าสู่ระบบการจัดเก็บและจัดการสื่อสารภาพ PACS (Archiving and Communication Systems) โดยการแปลงและจัดเก็บข้อมูลภาพทางการแพทย์และการทำงานของข้อกำหนดการสื่อสารชั้นบนที่สนับสนุนการแลกเปลี่ยนข่าวสารในสภาวะแวดล้อม TCP/IP เป็นไปตามมาตรฐาน DICOM (Digital Imaging and Communications in Medicine) 3.0[1] ซึ่งร่วมกันกำหนดขึ้นโดย ACR (American College of Radiology) และ NEMA (National Electrical Manufacturers Association) โดยจะอธิบายโครงสร้างของแฟ้มข้อมูลแบบ DICOM และโปรโตคอลส่วนบนแบบ DICOM จากนั้นกล่าวถึงการแปลงข้อมูล การสร้างโปรโตคอลส่วนบน สุดท้ายเป็นการแสดงการทำงานของโปรแกรมและผลที่ได้จากการทำงาน

Abstract

The variety in medical image file formats, due to the different of device manufacturers, lead to compatibility problem in telemedicine applications which concern about the exchanging of data from doctors between hospitals.

This project presents the method of storing medical images to PACS (Picture Archiving and Communication Systems). The converting and storing medical images structure, the upper layer protocols that support the Application Message Exchange in the TCP/IP network environment are followed the DICOM (Digital Imaging and Communications in Medicine) 3.0 standard, developed by ACR (the American College of Radiology) and NEMA (the National Electrical Manufacturers Association). First, the structure of DICOM file format and DICOM Upper Layer (DUL) Protocol are described. Next, the converting method and the DUL construction are explained. Finally, the implemented software and results from the real application are shown.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

สารบัญ II

สารบัญรูป III

บทที่ 1 บทนำ 1

1.1 หลักการและเหตุผล 1

1.2 วัตถุประสงค์และขอบเขตของโครงงาน 2

1.3 ผลที่คาดว่าจะได้รับ 3

1.4 เนื้อหาของโครงงาน 3

บทที่ 2 ทฤษฎี 4

2.1 บทนำ 4

2.2 โทรเวชและการสื่อสารภาพรังสีการแพทย์ 4

2.3 ภาพรวมของระบบการสื่อสารภาพรังสีการแพทย์ 4

2.4 มาตรฐาน DICOM 7

2.5 หลักการของมาตรฐาน DICOM 7

2.6 โครงสร้างข้อมูล DICOM 8

2.7 โปรโตคอลการสื่อสาร DICOM 9

2.8 โปรโตคอล TCP/IP 11

บทที่ 3 การออกแบบและโครงสร้างของโปรแกรม 20

3.1 บทนำ 20

3.2 โครงสร้างส่วนของการแปลงข้อมูลให้เข้าสู่โครงสร้างข้อมูลแบบ DICOM 20

3.3การพัฒนาการสื่อสาร 23

บทที่ 4 การทดลอง 31

4.1 บทนำ 31

4.2 การทดลองแปลงข้อมูล 31

4.3 การทดสอบโปรโตคอลส่วนบนแบบ DICOM 36

บทที่ 5 สรุปและแนวทางการพัฒนา 44

5.1 สรุปผลการทดลอง 44

5.2 แนวทางการพัฒนาขั้นต่อไป 44

บรรณานุกรม .....

กิตติกรรมประกาศ .....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 หลักการของโครงงาน 2

รูปที่ 2.1 ภาพรวมของระบบโทรเวชและขอบเขตของระบบเครือข่าย
ที่ใช้มาตรฐาน DICOM 4

รูปที่ 2.2 ตัวอย่างการเชื่อมต่อระบบการสื่อสารภาพรังสีการแพทย์เข้า
กับระบบโทรเวชของโรงพยาบาล 5

รูปที่ 2.3 แบบจำลองข้อมูลแบบ DICOM 8

รูปที่ 2.4 โครงสร้างข้อมูลที่ประกอบกันเป็น IOD 9

รูปที่ 2.5 สถาปัตยกรรมการสื่อสารข้อมูล DICOM รุ่นที่ 3.0 10

รูปที่ 2.6 โครงสร้างของ IP โปรโตคอล 12

รูปที่ 2.7 แพคเกตขนาด 1500 ไบต์ถูกแบ่งออกเป็น 3 แพคเก็ต 13
เมื่อผ่านเข้าสู่เครือข่ายที่ 2 และค่า fragment offset ที่บ่งถึงลำดับแพคเก็ต

รูปที่ 2.8 โครงสร้างของโปรโตคอล TCP 15

รูปที่ 2.9 ลำดับชั้นของแอพพลิเคชั่นและชุดของโปรโตคอล TCP/IP 18

รูปที่ 2.10 โปรโตคอลในชั้นบนสุดจะถูกห่อหุ้มด้วยโปรโตคอลในชั้นที่ต่ำกว่า 18

รูปที่ 2.11 สถานี A และ B ถูกกำหนดให้อยู่ในเครือข่ายเดียวกัน 19

รูปที่ 3.1 ขั้นตอนการแปลงข้อมูลให้อยู่ในโครงสร้างข้อมูลแบบ DICOM 20

รูปที่ 3.2 ผังงานการเข้ารหัสข้อมูล 22

รูปที่ 3.3 โครงสร้างแฟ้มข้อมูลหลังผ่านการแปลงเมื่อแสดงในรูปแบบเลขฐาน 16 23

รูปที่ 3.4 การวางตัวของชั้นโปรโตคอลส่วนบนแบบ DICOM 23

รูปที่ 3.5 การขอการเชื่อมต่อ 27

รูปที่ 3.6 การส่งข้อมูลผ่าน PDV 28

รูปที่ 3.7 การยกเลิกการเชื่อมต่อ 29

รูปที่ 4.1 กรอบข้อความเพื่อกำหนดเลข IP 31

รูปที่ 4.2 กรอบข้อความเพื่อเข้าสู่ระบบ 31

รูปที่ 4.3 หน้าจอหลักของโปรแกรม 32

รูปที่ 4.4 เลือกแฟ้มต้นทาง 32

รูปที่ 4.6 การใส่ข้อมูลคนไข้ 33

รูปที่ 4.7 แฟ้มเป้าหมายจะปรากฏ ณ รากเป้าหมาย 33

รูปที่ 4.8 แสดงด้วยโปรแกรมแสดงภาพ DICOM 34

รูปที่ 4.9 ผลจากเครื่องเอกซเรย์คอมพิวเตอร์ 35

รูปที่ 4.10 ภาพจากเครื่องถ่ายภาพกำทอนแม่เหล็ก 36

รูปที่ 4.11 ใส่ชื่อผู้ใช้และรหัสผ่าน 36

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

สืบเนื่องมาจากปริมาณแพทย์ต่อจำนวนประชากรที่มีอยู่ มีไม่เพียงพอต่อความต้องการ ไม่ว่าจะเป็นแพทย์ทั่วไป แพทย์ผู้ชำนาญในแต่ละสาขา รวมไปการขาดแคลนถึงเครื่องมือทางการแพทย์ที่ทันสมัย หรือถ้ามีแล้วก็มีผู้ชำนาญการวิเคราะห์ผลน้อย การที่จะใช้ทรัพยากรต่าง ๆ ที่มีอยู่ให้เกิดประโยชน์สูงสุดหรือการเพิ่มประสิทธิภาพของเครื่องมือที่มี หรือแม้แต่การที่ทำให้แพทย์สามารถที่จะตรวจรักษาผู้ป่วยได้สะดวกรวดเร็วขึ้นย่อมเป็นสิ่งที่บรรเทาปัญหาเหล่านั้นลงมาได้ การนำเทคโนโลยีสารสนเทศเข้ามาใช้ในวงการแพทย์ก็เป็นทางหนึ่งที่จะเพิ่มประสิทธิภาพในการดำเนินงาน[2]

งานโทรเวช (Telemedicine) เป็นการประยุกต์ใช้เทคโนโลยีโทรคมนาคม เทคโนโลยีสารสนเทศเข้ากับเทคโนโลยีทางการแพทย์ ระบบโทรเวชจะมีระบบหนึ่งที่เรียกว่าการสื่อสารภาพรังสีการแพทย์ (Teleradiology) เป็นระบบย่อยอยู่ภายในส่วนนี้จะเป็นการจัดการสื่อสารข้อมูลภาพรังสีวิทยาเป็นหลัก ประโยชน์ของระบบการสื่อสารภาพรังสีการแพทย์คือ

1.สามารถช่วยให้โรงพยาบาลที่ขาดแคลนรังสีแพทย์ผู้เชี่ยวชาญสามารถที่จะส่งผลการตรวจไปยังที่ที่มีผู้เชี่ยวชาญช่วยวินิจฉัยได้

2.ประหยัดตัวกลางในการจัดเก็บภาพที่ได้จากการตรวจเนื่องมาจากภาพที่จัดเก็บในระบบอยู่ในระบบดิจิตอล

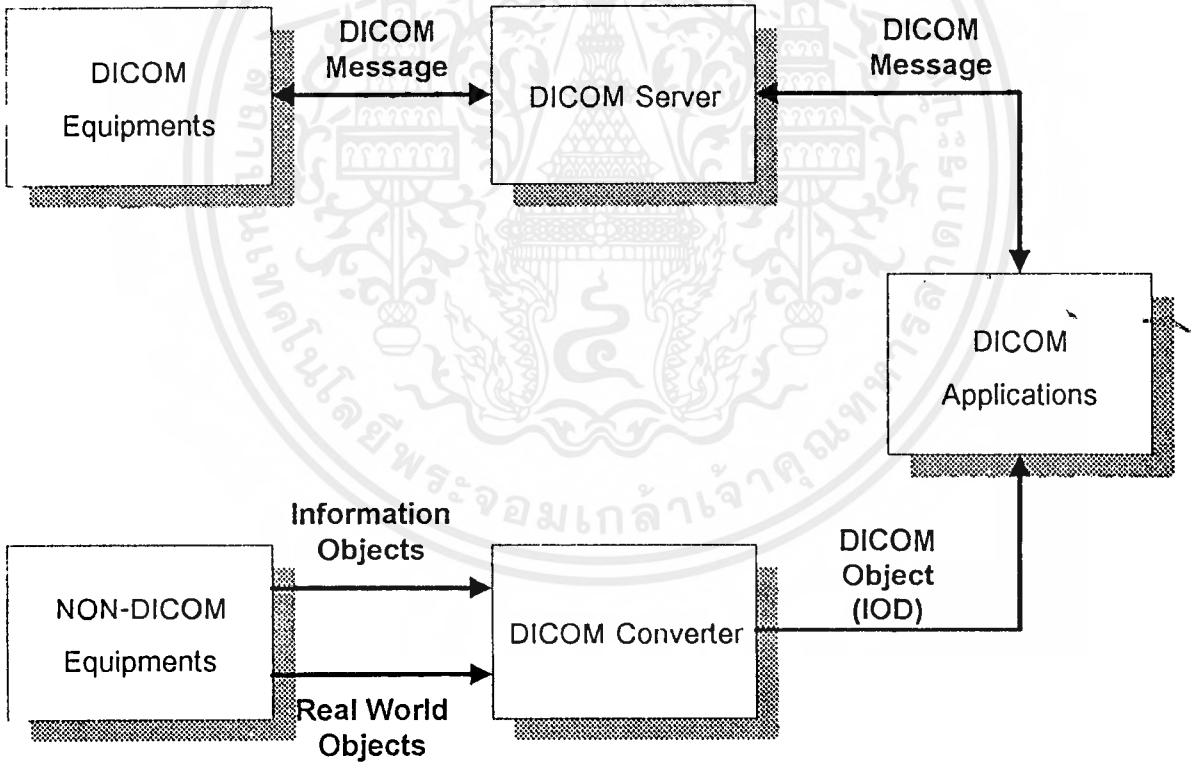
3.ทำให้สามารถเรียกดูประวัติการรักษา โดยเฉพาะภาพรังสีของผู้ป่วยที่มีประวัติเดิมอยู่ต่างโรงพยาบาลได้ อีกทั้งโอกาสการสูญหายของประวัติการรักษาโดยเฉพาะภาพรังสีมีน้อยมากหรือแทบจะไม่มีเลย

ในอดีตนั้นภาพในทางการแพทย์ที่ได้จากเครื่องมือทางรังสีวิทยาต่าง ๆ เช่น เครื่องถ่ายภาพกำทอนแม่เหล็ก (MRI) เครื่องเอกซเรย์คอมพิวเตอร์ (CT) และเครื่องอัลตราซาวด์ (US) ต่างก็มีวิธีการแสดงผลและการจัดเก็บภาพในรูปแบบที่แตกต่างกันไปขึ้นอยู่กับผู้ผลิตเครื่องมือนั้น การแลกเปลี่ยนข้อมูลระหว่างเครื่องมือต่างผู้ผลิตกันไม่สามารถทำได้โดยสะดวก ปัจจุบันจึงได้มีการกำหนดมาตรฐาน DICOM ขึ้นสำหรับเครื่องมือแพทย์เพื่อใช้เป็นมาตรฐานกลางสำหรับการแลกเปลี่ยนภาพโดยไม่ขึ้นกับผู้ผลิต แต่ถึงแม้ว่าเครื่องมือแพทย์รุ่นใหม่จะสามารถให้ข้อมูลได้ตามมาตรฐานนี้แล้วก็ตาม แต่ในประเทศไทยยังมีเครื่องมือแพทย์รุ่นเก่าที่ไม่ได้รองรับมาตรฐาน DICOM อยู่อีกเป็นจำนวนมาก หากสามารถทำให้ข้อมูลที่อยู่ในรูปแบบเก่าให้เข้าสู่รูปแบบข้อมูลมาตรฐาน DICOM ร่วมกับการพัฒนาระบบสื่อสารตามโปรโตคอลแบบ DICOM ก็สามารถนำข้อมูลจากเครื่องมือรุ่นเก่าเหล่านั้นเข้าสู่ระบบการสื่อสารภาพรังสีการแพทย์ อันจะเชื่อมโยงไปยังการใช้งานร่วมกับระบบโทรเวชได้ ส่งผลให้เป็นการเพิ่มประสิทธิภาพในการให้บริการของเครื่องมือแพทย์ที่มีอยู่ และสามารถประหยัดงบประมาณในการจัดซื้อเครื่องมือรุ่นใหม่ลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์และขอบเขตของโครงงาน

จุดประสงค์หลักของโครงงานนี้คือ การนำข้อมูลจากเครื่องมือแพทย์ที่ไม่ได้สนับสนุนมาตรฐาน DICOM ให้นำเข้าสู่ระบบที่ใช้งานมาตรฐาน DICOM ได้ดังแสดงในรูป 1.1 โดยทำการพัฒนาการจัดการข้อมูล ได้แก่ การแปลงข้อมูลภาพที่ไม่ได้อยู่ในมาตรฐาน DICOM ให้เข้าสู่มาตรฐาน DICOM[3] การอ่านและการแสดงภาพข้อมูล DICOM อีกส่วนหนึ่งคือการจัดการสื่อสารข้อมูลด้วยโปรโตคอล DICOM จากนั้นนำทั้งสองส่วนมาพัฒนาเพื่อให้สามารถนำข้อมูลจากเครื่องมือแพทย์ที่ไม่ได้ใช้มาตรฐาน DICOM เข้าสู่ DICOM Server ได้ การพัฒนาใช้ตัวแปลภาษาวิชวลซีรุ่นที่ 5 และทำงานบนระบบปฏิบัติการวินโดว์ 95 ของบริษัทไมโครซอฟต์ทดสอบความถูกต้องของข้อมูลโดยการแสดงผลข้อมูลที่ผ่านการแปลงด้วยโปรแกรม Osiris[4] ที่เป็นโปรแกรมมาตรฐานที่ใช้แสดงภาพ DICOM และใช้ โปรแกรม DICOM Server ของ RSNA[5] ซึ่งเป็นมาตรฐานที่ใช้ทดสอบความเข้ากันได้ของอุปกรณ์ที่ใช้มาตรฐาน DICOM ทดสอบความถูกต้องการจัดการสื่อสารข้อมูล ทำการทดสอบกับเครื่องมือแพทย์ 3 ชนิดคือ เครื่องถ่ายภาพกำทอนแม่เหล็ก เครื่องเอกซเรย์คอมพิวเตอร์และเครื่องอัลตราซาวด์ของภาควิชารังสีวิทยา ศูนย์การแพทย์สิริกิตติ์ โรงพยาบาลรามาธิบดี



รูปที่ 1.1 หลักการของโครงงาน

1.3 ผลที่คาดว่าจะได้รับ

ผลหลักของโครงงานนี้คือสามารถนำข้อมูลจากเครื่องมือแพทย์ที่ไม่ได้สนับสนุนมาตรฐาน DICOM เข้าสู่ระบบเครือข่าย DICOM ได้ โดยเมื่อนำข้อมูลเข้าระบบได้แล้วผลที่ได้ตามมาโดยอาศัยข้อดีของการสื่อสารภาพรังสีการแพทย์คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.แพทย์สามารถที่จะเรียกดูประวัติและภาพรังสีของผู้ป่วยได้จากทุกจุดที่ต่อกับระบบเครือข่ายของโรงพยาบาล

2.ช่วยให้โรงพยาบาลที่ขาดแคลนผู้เชี่ยวชาญการวิเคราะห์ภาพรังสีสามารถส่งภาพรังสีมาให้ผู้เชี่ยวชาญอ่านได้โดยตรง ทำให้สะดวกทั้งแพทย์และคนไข้ คนไข้สามารถไปที่หน่วยงานที่มีเครื่องมือถ่ายภาพรังสีจากนั้นผลก็จะไปถึงมือของแพทย์ผู้รักษาได้ทันที

3.ลดการใช้งานของตัวกลางในการจัดเก็บภาพรังสีลงเมื่อจัดเก็บภาพเหล่านั้นลงในคอมพิวเตอร์ ทำให้เป็นการประหยัดงบประมาณในการจัดซื้อและดูแลรักษาตัวกลางในการจัดเก็บ

4.ประวัติการรักษาและภาพรังสีของคนไข้สามารถสืบค้นและยากที่จะสูญหาย ทำให้การตรวจรักษาของแพทย์ทำได้รวดเร็วและปลอดภัยยิ่งขึ้น

5.ทำให้ระบบเครือข่ายของโรงพยาบาลที่ใช้เป็นมาตรฐานสากลทำให้สามารถแลกเปลี่ยนข้อมูลกับโรงพยาบาลในต่างประเทศได้ อีกทั้งเป็นการรองรับเทคโนโลยีที่จะเกิดขึ้นใหม่ในอนาคต

.4 เนื้อหาของโครงงาน

รายละเอียดของโครงงานนี้ประกอบไปด้วย

บทที่ 1 บทนำ

อธิบายถึงที่มาของปัญหาและวิธีที่จะนำมาแก้ปัญหา

บทที่ 2 ทฤษฎี

อธิบายระบบการสื่อสารภาพรังสีการแพทย์ โครงสร้างข้อมูลและการจัดการสื่อสารข้อมูลตามมาตรฐาน DICOM

บทที่ 3 การออกแบบและการพัฒนาโปรแกรม

อธิบายขั้นตอนการทำงานของโปรแกรมและการเขียนโปรแกรม

บทที่ 4 การทดลอง

แสดงการนำโปรแกรมไปใช้งานกับข้อมูลจริงและผลที่ได้

บทที่ 5 สรุป วิจารณ์ และแนวทางการพัฒนาต่อ

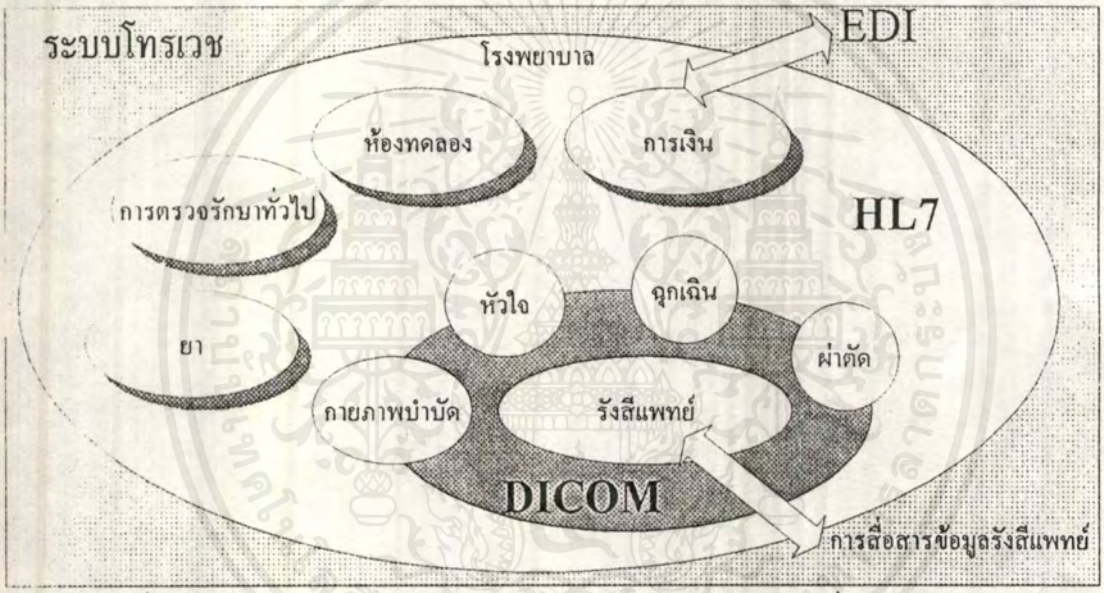
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 บทนำ

ในบทนี้จะกล่าวถึงภาพรวมของโทรเวช การสื่อสารภาพรังสีการแพทย์ มาตรฐาน DICOM ทั้งในด้านข้อมูลและการสื่อสารข้อมูล ในด้านข้อมูลกล่าวถึงรายละเอียดข้อมูลแบบ DICOM โครงสร้างข้อมูล และการจัดเก็บข้อมูลลงหน่วยความจำสำรอง ในด้านการสื่อสารกล่าวถึงการจัดการโครงสร้างส่วนบนแบบ DICOM ซึ่งทำงานบนโปรโตคอลชั้นขนส่งแบบ TCP/IP

2.2 โทรเวชและการสื่อสารภาพรังสีการแพทย์



รูปที่ 2.1 ภาพรวมของระบบโทรเวชและขอบเขตของระบบเครือข่ายที่ใช้มาตรฐาน DICOM

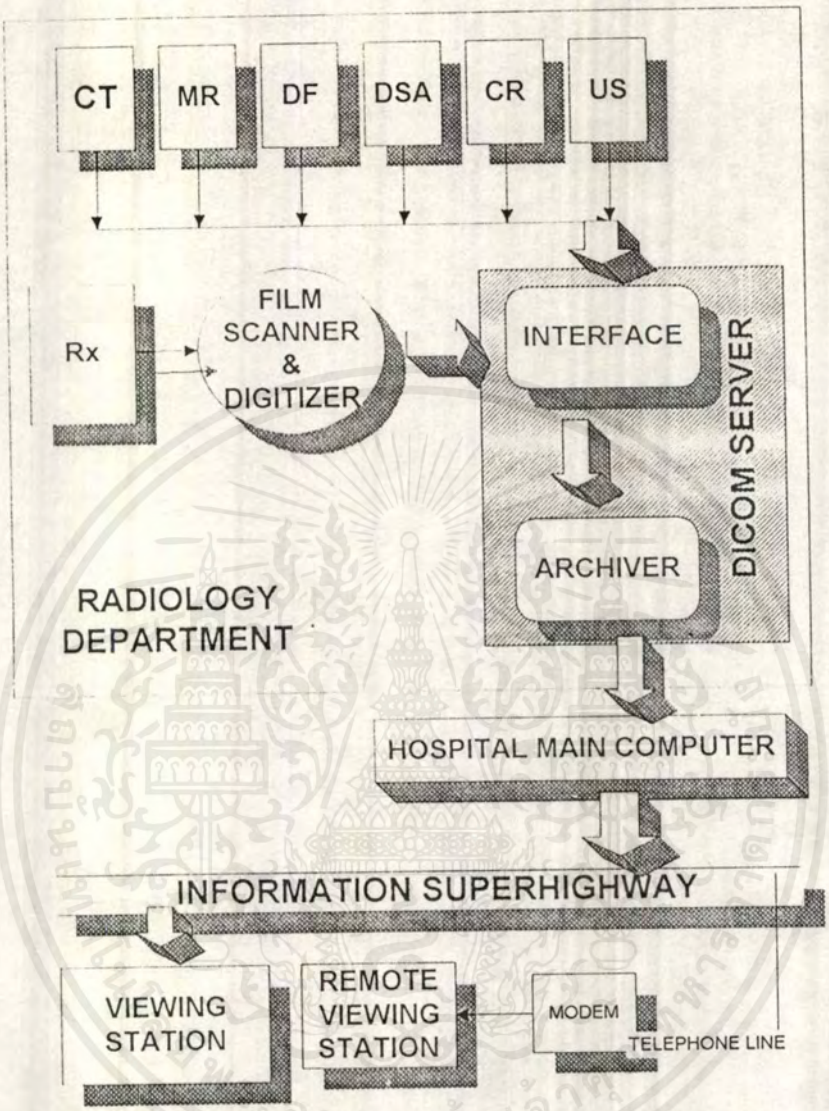
ในระบบโทรเวชได้มีการกำหนดมาตรฐานในการทำงานร่วมกันของหน่วยงานแต่ละหน่วยงานให้มีมาตรฐานที่เหมาะสมของหน่วยนั้นๆ กล่าวคือ ระบบโรงพยาบาลอันเกี่ยวกับข้อมูลการรักษาทั่วไป ข้อมูลการทดลอง ข้อมูลยา จะมีมาตรฐานอ้างอิงเรียกว่า Health Level 7 - HL7 ส่วนมาตรฐานทางการเงินใช้มาตรฐานตามข้อกำหนด EDI ส่วนงานที่เกี่ยวกับด้านรังสีแพทย์หรือภาพทางการแพทย์นั้นจะใช้มาตรฐาน DICOM เป็นมาตรฐานกลาง

2.3 ภาพรวมของระบบการสื่อสารภาพรังสีการแพทย์

จากรูปที่ 2.2 ระบบจะประกอบไปด้วยอย่างน้อยคืออุปกรณ์กำเนิดภาพ(Modality) หรือเครื่องมือทางรังสีวิทยาเช่น เครื่องเอกซเรย์คอมพิวเตอร์(CT) เครื่องถ่ายภาพกำทอนแม่เหล็ก(MR) เครื่องอัลตร้าซาว์ด(US) ภาพที่ออกมาจากเครื่องมือเหล่านี้จะถูกนำไปจัดเก็บยังอุปกรณ์การจัดเก็บ การที่แพทย์จะเรียกดูหรือส่งผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพเหล่านั้นผ่านระบบเครือข่ายก็ต้องใช้ชุดคำสั่งและโปรโตคอลการสื่อสารแบบ DICOM ซึ่งจะกล่าวต่อไป โดยการบริหารและการจัดการระบบเป็นภาระหน้าที่ของ DICOM Server



รูปที่ 2.2 การเชื่อมต่อระบบการสื่อสารภาพรังสีการแพทย์เข้ากับระบบโทรเวชของโรงพยาบาล

โดยสรุประบบการสื่อสารภาพรังสีการแพทย์ประกอบไปด้วยส่วนต่างๆ ดังนี้[6]

1. แหล่งกำเนิดภาพ หมายถึง อุปกรณ์ทางการแพทย์ที่สามารถให้ภาพออกมาได้ อาจจะรวมไปถึงอุปกรณ์การจับสัญญาณภาพและรวมไปถึงข้อมูลที่อธิบายรายละเอียดของภาพด้วย โดยทั่วไปภาพที่ได้จากเครื่องมือแพทย์จะมีคุณสมบัติดังตารางที่ 2.1 ข้อมูลสามารถแบ่งย่อยได้คือ

- ภาพนิ่ง มีทั้ง 2 มิติเช่น ภาพถ่ายทรวงอก(Mamograms) และ 3 มิติ เช่นภาพจากเครื่องเอกซเรย์ คอมพิวเตอร์ (CT Scan)
- ภาพเคลื่อนไหว เช่น ภาพจากเครื่องอัลตราซาวด์ หรือภาพต่อเนื่องจากเครื่องถ่ายภาพกำทอนแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข้อมูลตัวอักษรแบบมีโครงสร้างเช่น ข้อมูลรายละเอียดของภาพ(Image Header)
- ข้อมูลตัวอักษรทั่วไป เช่น ข้อมูลผลการวินิจฉัย
- ข้อมูลเสียงเช่น ข้อมูลเสียงที่บันทึกโดยผู้บันทึกผล ข้อมูลเสียงเพื่อการเข้าจังหวะภาพ
- ข้อมูลภาพจำลองเช่น แบบจำลองของสมองที่สร้างขึ้นจากภาพ 2 มิติ

Modality	Image Dimension (pixels)	Gray Level (bits)	Avg. Size/Exam (Mbytes)
Nuclear medicine(PET, SPECT)	128x128	12	1-2
Magnetic resonance imaging	256x256	12	8-20
Ultrasound	512x512	8	5-10
Computed tomography	512x512	12	20-40
Sprial or helical CT	512x512	12	80-160
Digitized electronic microscopy	512x512	8	Varies
Digitized color microscopy	512x512	24	Varies
Digital subtraction angiography(per run)	512x512 or 1024x1024	8	100-500
Digitized X-rays	2048 x 2048	12	8
Computed radiography	2048x2048	12	8
Digitized mammography	4096x4096	12	128(4 images)

ตารางที่ 2.1คุณสมบัติของภาพจากการใช้เครื่องมือแพทย์แต่ละครั้ง

2.การประมวลผลภาพหลังจากการถ่าย คือการจัดการเกี่ยวกับภาพภายหลังจากเครื่องมือได้ให้ภาพออกมา เช่น การจัดเก็บลงหน่วยความจำสำรอง การบีบอัดภาพ

3. การสื่อสารข้อมูล คือการส่งรับข้อมูลภาพและข้อมูลต่างๆที่เป็นประโยชน์ต่อการวินิจฉัยภาพ

4. การแสดงภาพ คือการนำภาพที่ได้จากเครื่องมือไปแสดงได้ ณ จุดที่มีตัวดูภาพที่เป็นมาตรฐาน

5. การบริหารฐานข้อมูลภาพคือการจัดการเกี่ยวกับฐานข้อมูลภาพและต้องเกี่ยวเนื่องกับฐานข้อมูลหลักของโรงพยาบาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. การนำภาพไปวิเคราะห์ คือการนำภาพที่ได้ใช้วิเคราะห์ผลด้วยเครื่องมืออำนวยความสะดวกต่างๆ เช่น ซอฟต์แวร์ช่วยนับปริมาณแคลเซียมในหลอดเลือดหัวใจ[7] เป็นต้น

2.4 มาตรฐาน DICOM

DICOM ย่อมาจาก The Digital Imaging and Communications in Medicine เป็นมาตรฐานที่กำหนดในรายละเอียดของรูปแบบการจัดเก็บและวิธีแลกเปลี่ยนข้อมูลภาพทางการแพทย์และข้อมูลอื่นที่เกี่ยวข้องภาพทางการแพทย์ เปรียบเสมือนเนื้อหาความเข้าใจร่วมกันระหว่างผู้ใช้และผู้ผลิตอุปกรณ์สร้างภาพรังสีการแพทย์

2.5 หลักการของมาตรฐาน DICOM

หลักการของมาตรฐาน DICOM นั้นเปรียบเสมือนภาษาหนึ่ง กล่าวคือในภาษาหนึ่งต้องมีคำศัพท์ ไวยากรณ์ภาษา และรูปแบบการสื่อสารเป็นของตัวเอง มาตรฐาน DICOM ก็มีส่วนคำศัพท์ที่กำหนดไว้ในข้อกำหนดส่วนมีที่ 6 (Part 6 : Data Dictionary) และส่วนไวยากรณ์ที่กำหนดไว้ในส่วนที่ 5 (Part 5 : Data Structure and Encoding) ส่วนรูปแบบการสื่อสารนั้นกำหนดไว้ในส่วนที่ 7 (Part 7: DICOM Message) สำหรับส่วนประกอบของมาตรฐาน DICOM ที่กำหนดไว้อ้างอิงมี 13 ส่วนคือ

PART 1: Introduction and Overview

อธิบายโครงสร้างทั้งหมดของมาตรฐาน

PART 2 : Conformance

ระบุถึงความต้องการต่างๆ ตัวแปรพื้นฐานและข้อกำหนดต่างๆ

PART 3 : Information object Definitions

กำหนดโครงสร้าง และคุณสมบัติของข้อมูลกำหนดในการพัฒนา

PART 4 : Service Class Specifications

กำหนดการกระทำระหว่างข้อมูลจาก ส่วนที่ 3 อันแสดงถึงขอบเขตการให้บริการของข้อมูลแต่ละชนิด

PART 5 : Data Structure and Semantics

กำหนดการเข้ารหัสข้อมูลที่จะทำการแลกเปลี่ยนขึ้นอยู่กับส่วนที่ 4

PART 6 : Data Dictionary

กำหนดรายละเอียดของข้อมูลที่รวมกันเป็นข้อมูลแบบ DICOM

PART 7 : Message Exchange

กำหนดกระบวนการและโปรโตคอลที่จะแลกเปลี่ยนข่าวสารระหว่างการให้บริการ

PART 8 : Network Communication Support for Message Exchange

กำหนดการเข้าถึงโปรโตคอลชั้นขนส่ง (Transport Layer)

PART 9 : Point-to-Point Communication Support for Message Exchange

กำหนดคุณสมบัติการเชื่อมต่อระหว่างอุปกรณ์ในระดับ Hardware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PART 10 : Media Storage and File Format

กำหนดการจัดเก็บข้อมูลของแฟ้มข้อมูล

PART 11 : Application Profiles

กำหนดส่วนของผู้ผลิตที่จะเลือกใช้อุปกรณ์การจัดเก็บ

PART 12 : Media Formats and Physical Media for Data Interchange

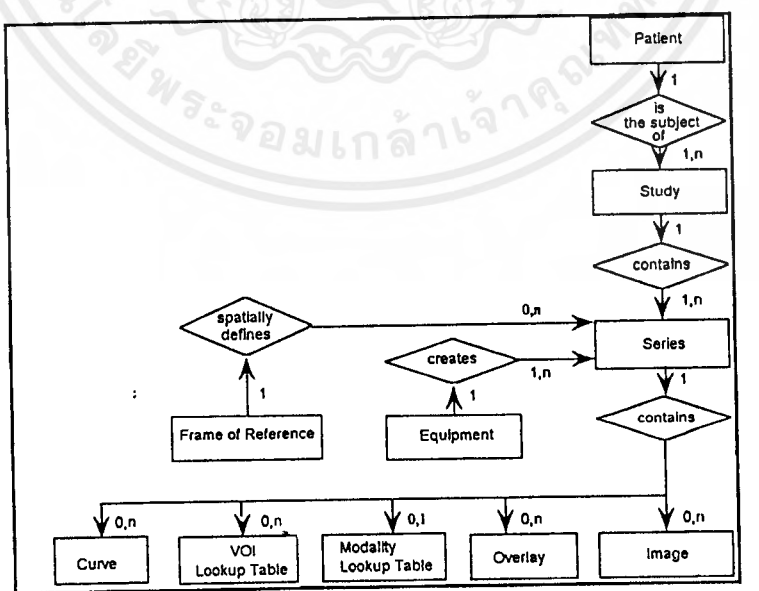
กำหนดรูปแบบการจัดเก็บและแลกเปลี่ยนลงอุปกรณ์ในระดับกายภาพ

PART 13 : Print Management Point-to-Point Communication Support

กำหนดมาตรฐานการพิมพ์

2.6 โครงสร้างข้อมูล DICOM

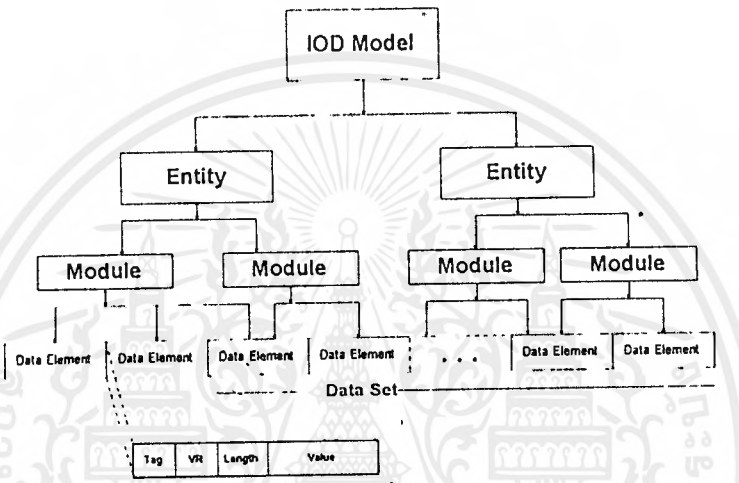
โครงสร้างข้อมูลแบบ DICOM คือกลุ่มของข้อมูลที่มีการจัดโครงสร้างตาม Information Object Definitions (IOD)[7] โครงสร้างข้อมูลแบบ DICOM สามารถอธิบายได้ด้วยแบบจำลองข้อมูลข้อมูล (ดังแสดงในรูปที่ 2.3) กล่าวคือ ข้อมูลแบบ DICOM คือข้อมูลที่มีคุณสมบัติตาม IOD ซึ่งใน IOD 1 หน่วยจะต้องประกอบไปด้วย เอนทิตี้ของข้อมูลคนไข้ (Patient Entity) 1 เอนทิตี้ เอนทิตี้ข้อมูลการรักษา (Study Entity) อย่างน้อย 1 เอนทิตี้ซึ่งแต่ละข้อมูลการรักษาประกอบไปด้วยอย่างน้อย 1 เอนทิตี้ของข้อมูลลำดับ (Series Entity) ที่เกิดจากเอนทิตี้ข้อมูลเครื่องมือแพทย์ (Equipment Entity) ในแต่ละข้อมูลลำดับสามารถประกอบไปด้วยเอนทิตี้ของภาพ (Image Entity) เอนทิตี้ของข้อมูลเส้นโค้ง (Curve Entity) เอนทิตี้ของข้อมูลตารางเครื่องมือทางการแพทย์ (Modality Lookup Table Entity) เอนทิตี้ข้อมูลซ้อนทับ (Overlay Entity) และเอนทิตี้ข้อมูลตารางการพิมพ์ (VOI Lookup Table) และใน IOD 1 หน่วยนี้อาจจะมีเอนทิตี้ของข้อมูลเฟรมอ้างอิง (Frame of Reference Entity) หรือไม่ก็ได้



รูปที่ 2.3 แบบจำลองข้อมูลแบบ DICOM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางปฏิบัติข้อมูลหน่วยที่เล็กที่สุดของโครงสร้างข้อมูลแบบ DICOM เรียกว่า Data Element [8] (ดังรูปที่ 2.4) ซึ่งใน 1 หน่วยจะมีส่วนประกอบคือ หมายเลขประจำ (Tag) ตัวแสดงวิธีการจัดเก็บ (Value Representation-VR) ตัวเลขแสดงความยาว (Value length) และข้อมูลของ Data Element (Value Field) Data Element จะถูกจัดแบ่งออกเป็นกลุ่มตามความสัมพันธ์ของข้อมูลเรียกว่าโมดูล (Module) โมดูลจะถูกจัดกลุ่มอีกครั้งหนึ่งยอมเป็นเอนทิตี้ และสุดท้ายเอนทิตี้ถูกนำมารวมกันตามแบบจำลองข้อมูลDICOM ดังที่กล่าวมา เมื่อสามารถจัดข้อมูลได้เป็นตามแบบโครงสร้างข้อมูลของมาตรฐานDICOM แล้วก็นำข้อมูลแต่ละ Data Element ที่ประกอบกันขึ้นมาเข้ารหัสแล้วจัดเรียงตามหมายเลขประจำจากน้อยไปหามาก ข้อมูลที่ถูกเข้ารหัสแล้วถูกจัดเรียงนี้เรียกว่า Data Set ซึ่งสามารถอ่านและประมวลผลได้จากโปรแกรมที่ทำงานสนับสนุนมาตรฐาน DICOM



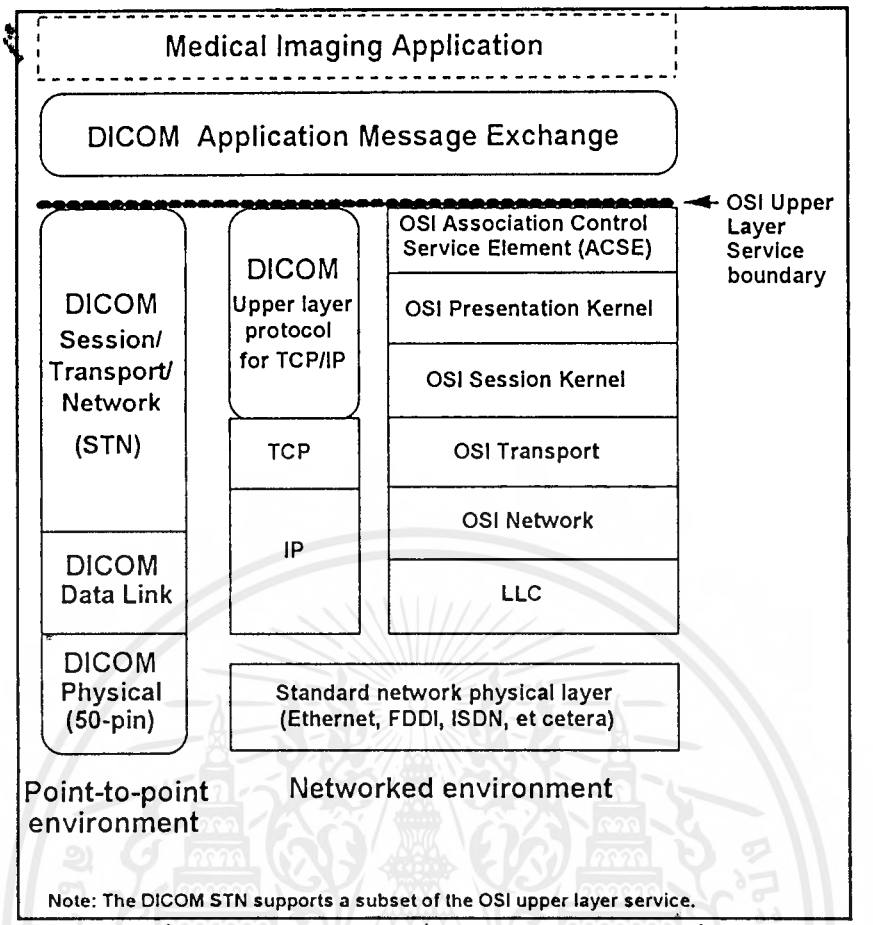
รูปที่ 2.4 โครงสร้างข้อมูลที่ประกอบกันเป็น IOD

2.7 โปรโตคอลการสื่อสาร DICOM[9]

โปรโตคอลส่วนบนแบบ DICOM สำหรับโปรโตคอลสื่อสาร TCP/IP (DICOM Upper Layer Protocol for TCP/IP) ในส่วนนี้เป็นการกล่าวถึงการใช้โปรโตคอลส่วนบนแบบ DICOM เพื่อเชื่อมต่อกับโปรโตคอลชั้น การขนส่ง(Transport Protocol) โปรโตคอลส่วนบนแบบ DICOM และการเชื่อมต่อโปรโตคอล TCP/IP เป็นความสัมพันธ์แบบหนึ่งต่อหนึ่ง โดยที่หมายเลขพอร์ทกำหนดที่เครื่องให้บริการ (DICOM Server) เป็น 104

ในระบบเครือข่าย DICOM การแลกเปลี่ยนข่าวสารข้อมูลจะทำกันระหว่างสิ่งที่เรียกว่า เอกลักษณ์ประยุกต์อันเป็นส่วนที่ใช้จัดการเกี่ยวกับการสื่อสาร ภายในเอกลักษณ์ประยุกต์นี้ประกอบไปด้วยกระบวนการของผู้ขอใช้บริการ ข้อกำหนดของการจัดสร้างการติดต่อสื่อสารและข้อมูลที่จะทำการแลกเปลี่ยน เอกลักษณ์ประยุกต์แต่ละเอกลักษณ์จะมีชื่อของประจำตัวเรียกว่า ชื่อของเอกลักษณ์ประยุกต์(Application Title) ซึ่งจะถูกใช้ในระหว่างกันจัดสร้างการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 สถาปัตยกรรมการสื่อสารข้อมูล DICOM รุ่นที่ 3.0

2.7.1 หมายตำแหน่งการแสดงตัว (Presentation Address)

ชื่อของเอกลักษณ์ประยุกต์เป็นสัญลักษณ์สำหรับกระบวนการสื่อสารแต่มันจะมีความซับซ้อนในระบบสื่อสารในระบบเครือข่ายนั้นหมายเลขเครือข่ายจะถูกนำมาใช้แทน หมายเลขเครือข่ายนี้เรียกว่า หมายเลขตำแหน่งการแสดงตัวและหมายเลขนี้จะชี้ไปยังหมายเลขที่อยู่ของเอกลักษณ์ประยุกต์แทน ที่ถูกเรียกว่าหมายเลขตำแหน่งการแสดงตัวนั้นเพราะว่า กระบวนการของผู้ใช้บริการนั้นอยู่ในชั้นประยุกต์ของแบบจำลอง OSI ซึ่งทำการเตรียมการให้กับชั้นแสดงตัวของแบบจำลอง OSI ขอบเขตของชั้นนี้คือจุดเข้าถึงของระบบเครือข่าย (Network Access Point) นั่นคือเป็นจุดที่ข้อมูลจากชั้นประยุกต์เข้าสู่ชั้นของระบบการสื่อสาร ซึ่งแต่ละจุดเข้าถึงนั้นจะมีหมายเลขประจำเพียงค่าเดียว การเปลี่ยนจากชื่อของเอกลักษณ์ไปเป็นหมายเลขตำแหน่งการแสดงตัวนั้นไม่ได้มีเพียงค่าเดียว เพราะว่าหมายเลขตำแหน่งการแสดงตัวใช้สำหรับการเริ่มต้นการสื่อสาร อย่างไรก็ตามชั้นประยุกต์ของชื่อของเอกลักษณ์ประยุกต์ใช้บอกตำแหน่งของจุดเริ่มและจุดสิ้นสุดของข้อมูลที่จะแลกเปลี่ยนรูปแบบของหมายเลขตำแหน่งการแสดงตัวนั้นเป็นไปตามรูปแบบของการเข้าถึงชั้นของโปรโตคอลการสื่อสารลงไปในที่นี้เป็น TCP/IP โปรโตคอล กล่าวคือหมายถึงหมายเลขตำแหน่งการแสดงตัวเป็นไปตามหมายเลข IP และหมายเลขซอกเกตนั่นเอง

2.7.2 การเชื่อมต่อสำหรับการแลกเปลี่ยน (Association Negotiation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อสำหรับการแลกเปลี่ยนข้อมูลระหว่างเอกลักษณ์ประยุกต์สองเอกลักษณ์เรียกว่า การเชื่อมต่อ (Association) สำหรับการเชื่อมต่อนั้นถูกกำหนดโดยคำอธิบาย (Context) มีชื่อเรียกว่า คำอธิบายประยุกต์ (Application Context) ซึ่งถูกระบุไว้ในข้อกำหนด DICOM ด้วยคำอธิบายประยุกต์ ถูกกำหนดโดย UID และระหว่างการเริ่มต้นการสื่อสารนั้น UID จะถูกส่งผ่านระหว่างคู่สื่อสาร โดยการเปรียบเทียบค่า UID ของคำอธิบายประยุกต์ เพื่อให้คู่สื่อสารนั้นตัดสินใจในการทำการสื่อสารต่อไป คำอธิบายประยุกต์ครอบคลุมถึงหน้าที่ต่างๆสำหรับการแลกเปลี่ยนข่าวสารกล่าวคือหน้าที่ต่างๆ จะถูกระบุโดย SOP Classes และ Service Classes ที่ถูกกำหนดไว้ในข้อกำหนด DICOM ส่วนที่ 4

2.7.3 คำอธิบายการแสดงตัว (Presentation Context)

สำหรับแต่ละ SOP Class ที่ถูกแลกเปลี่ยนในระหว่างเริ่มต้นการเชื่อมต่อข้อตกลงระหว่างคู่การสื่อสารถูกกำหนดโดยไวยากรณ์การส่ง (Transfer Syntax) คู่สื่อสารแต่ละด้านจะนำเอาไวยากรณ์การส่งมาพิจารณากับ SOP Class

2.7.4 การติดต่อกับโปรโตคอล TCP/IP

สืบเนื่องมาจากในข้อกำหนด TCP/IP ไม่ได้กำหนดส่วนของโปรโตคอลชั้นบนไว้ ข้อกำหนด DICOM ได้กำหนดการติดต่อกับ TCP/IP โดยผ่าน DUL (DICOM Upper Layer Protocol) ซึ่งทำหน้าที่จัดการเชื่อมต่อ TCP/IP โดยที่ หมายเลขการแสดงตัวจะถูกเปลี่ยนเป็นหมายเลขพอร์ตของ TCP ซึ่งจะนำไปรวมกันกับ หมายเลข IP และ ชื่อ HOST

2.8 โปรโตคอล TCP/IP

2.8.1 โปรโตคอล IP

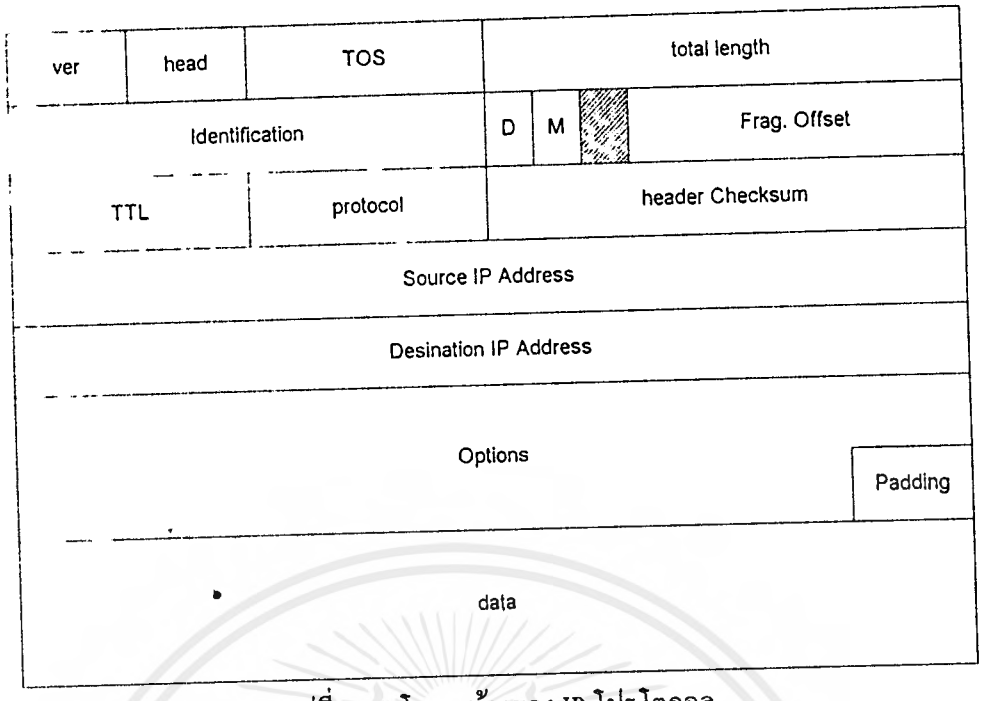
โปรโตคอล IP เป็นโปรโตคอลที่ให้บริการแก่โปรโตคอลชั้นที่สี่ ในแบบที่เรียกว่า "connectionless : CL" นั่นคือ ไม่มีการกำหนดหรือหาเส้นทางระหว่างต้นทางกับปลายทางเอาไว้ก่อน แต่การหาเส้นทางขึ้นอยู่กับอุปกรณ์การหาเส้นทาง (router) ซึ่งก็หมายความว่าภายใน IP แพคเก็ตจะต้องมีสารสนเทศที่สำคัญและเพียงพอต่ออุปกรณ์ดังกล่าว ในการนำส่ง IP แพคเก็ตไปยังปลายทาง ได้รูปแบบของโปรโตคอล IP แสดงในรูปที่ 2.6 มีรายละเอียดดังนี้

- Version (ver) มีขนาด 4 บิต แสดงถึงรุ่นของโปรโตคอล IP เนื่องจากการตีความในเขตต่างๆ ของโปรโตคอล IP ใช้ซอฟต์แวร์เป็นตัวจัดการ ดังนั้นจึงจำเป็นต้องมีการบ่งบอกถึงรุ่นของโปรโตคอล IP ไว้ด้วย ในกรณีที่รุ่นไม่ตรงกันจะไม่มีการตีความเกิดขึ้น IP แพคเก็ตที่ใช้งานอยู่ในขณะนี้เป็นรุ่นที่ 4 (ver=4)

- Header (head) มีขนาด 4 บิต บอกถึงความยาวเฉพาะส่วนหัวของ IP แพคเก็ต (ส่วนหัวของ IP แพคเก็ต เริ่มนับจากเขต version ไปจนถึงไบต์สุดท้ายก่อนที่จะถึงเขต Data) หน่วยนับเป็นจำนวนเท่าของ 4 ไบต์ เช่น ถ้า header มีค่าเท่ากับ 5 หมายถึงส่วนหัวมีขนาด 20 ไบต์ เป็นต้น

- Type of Service (TOS) มีขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 โครงสร้างของ IP โปรโตคอล

- Precedence มีขนาด 3 บิต ใช้ในการกำหนดลำดับความสำคัญของ IP แพคเก็ต ใน กรณีที่ปริมาณของแพคเก็ตมีอยู่มากในระบบอุปกรณ์หาเส้นทางจะไม่ทำการหาเส้น ทางให้กับแพคเก็ตที่มีลำดับความสำคัญต่ำสุดหรือกล่าวง่ายๆก็คือตัดแพคเก็ตนั้นออกจากระบบ

- D บิต มีขนาด 1 บิต ใช้บอกถึงความต้องการการให้บริการจากเครือข่าย ถ้าบิตนี้มีค่าเป็น 1 หมายถึงต้องการเส้นทางที่มีการหน่วงเวลา (delay time) ต่ำที่สุดเท่าที่ระบบจะหาให้ได้

- T บิต มีขนาด 1 บิต ถ้าบิตนี้มีค่าเป็น 1 หมายถึงต้องการเส้นทางที่มีความสามารถส่งผ่านข้อมูลได้ปริมาณมาก ๆ ในหนึ่งหน่วยเวลา

- R บิต มีขนาด 1 บิต ถ้าบิตนี้มีค่าเป็น 1 หมายถึงต้องการเส้นทางหรือบริการที่มีความ เชื่อถือได้สูง

- C บิต มีขนาด 1 บิต ถ้าบิตนี้มีค่าเป็น 1 หมายถึงต้องการเส้นทางที่มีค่าพารามิเตอร์ของ cost ต่ำ (ในส่วนของการหาเส้นทางนั้น cost ต่ำหมายถึงเป็นเส้นทางที่ดี)

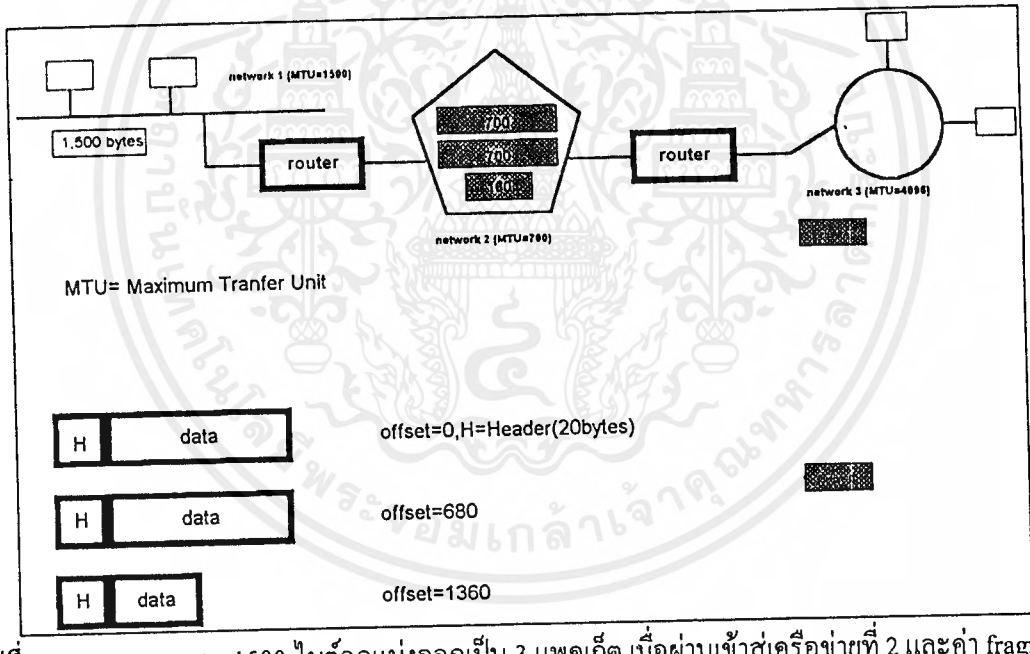
- unused มีขนาด 1 บิต ยังไม่มีการนำบิตนี้มาใช้งาน

- Total length มีขนาด 16 บิต ใช้บอกถึงความยาวของ IP แพคเก็ตทั้งหมด (ส่วนหัวและส่วนข้อมูล) หน่วยนับเป็นจำนวนเท่าของไบต์ ดังนั้น IP แพคเก็ตมีความยาวสูงสุดเท่ากับ 2^{16} หรือ 65,536 ไบต์

- Identification มีขนาด 16 บิต ใช้บอกถึงหมายเลขของ IP แพคเก็ต เขตนี้ถูกกำหนดขึ้นมาเพื่อไม่ให้สถานีปลายทางสับสนว่าตัวมันเองได้รับแพคเก็ตที่สมควรได้รับแล้วหรือไม่ เนื่องจากการส่งผ่านข้อมูลในลักษณะ connectionless นั้นมีโอกาสที่ข้อมูลจะสูญหายหรือซ้ำได้ ดังนั้นการกำหนดหมายเลขนี้ขึ้นมาก็เพื่อป้องกันการสับสนที่อาจจะเกิดขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- D บิต มีขนาด 1 บิต ถ้าบิตนี้มีค่าเป็น"1" หมายความว่าห้ามตัดทอน IP แพคเก็ตย่อยโดยเด็ดขาด แต่ถ้าบิตนี้มีค่าเป็น "0" หมายความว่าสามารถตัดทอน IP แพคเก็ตออกเป็นแพคเก็ตย่อยได้ การกระทำเช่นนี้มีข้อดีคือ เมื่อ IP แพคเก็ตผ่านจากระบบหนึ่งไปสู่ระบบอื่น ๆ ที่มีขนาดของแพคเก็ตที่ต่างกัน อุปกรณ์หาเส้นทางสามารถแบ่งแพคเก็ตขนาดใหญ่นั้นให้มีขนาดเล็กลง และพอดีกับระบบใหม่ได้ ทำให้มีความยืดหยุ่นและได้ประโยชน์สูงสุดในการส่งข้อมูลผ่านระบบเครือข่ายที่มีขนาดแพคเก็ตไม่เท่ากัน สำหรับโปรโตคอลอีเธอร์เนต และโทเก้นริงนั้น ไม่มีส่วนใดเลยที่อนุญาตให้กระทำเช่นนี้ได้ เช่นถ้าต้องการเชื่อมระบบอีเธอร์เนตซึ่งมีขนาดความยาวของข้อมูลสูงสุดเป็น 1,500 ไบต์เข้ากับระบบโทเก้นริงที่มีขนาดความยาวข้อมูล 4,096 ไบต์ แสดงในรูป 2.7 จะเห็นได้ว่าการส่งข้อมูลจากอีเธอร์เนตไปยังโทเก้นริงสามารถทำได้โดยตรง แต่ในทางตรงกันข้าม จากโทเก้นริงไปยังอีเธอร์เนตไม่สามารถทำได้แต่ถ้าต้องการก็จำเป็นต้องลดขนาดข้อมูลของโทเก้นริงให้เป็น 1,500 ไบต์เสียก่อน ซึ่งหมายความว่าทุก ๆ สถานีภายในโทเก้นริงต้องใช้ความยาวข้อมูลขนาด 1,500 ไบต์เหมือนกันหมด ถึงแม้ว่าจะเป็นการส่งระหว่างสถานีภายในระบบโทเก้นริงเองก็ตามอนึ่ง การแบ่งแพคเก็ต IP ออกเป็นแพคเก็ตย่อยนั้น แต่ละแพคเก็ตย่อยต้องมีหมายเลข IP แพคเก็ตเดียวกันกับแพคเก็ตใหญ่เสมอ



รูปที่ 2.7 แพคเกตขนาด 1500 ไบต์ถูกแบ่งออกเป็น 3 แพคเก็ต เมื่อผ่านเข้าสู่เครือข่ายที่ 2 และค่า fragment offset ที่บ่งถึงลำดับแพคเก็ต

- M บิต มีขนาด 1 บิต ถ้าบิตนี้ถูกกำหนดเป็น 1 ในแพคเก็ตใด หมายความว่าเป็นแพคเก็ตย่อยที่ถูกแบ่งออกมาจากแพคเก็ตใหญ่โดยมีหมายเลข IP แพคเก็ตเหมือนกับแพคเก็ตใหญ่ เพื่อที่ปลายทางจะสามารถรวบรวมแพคเก็ตย่อยเหล่านี้ให้กลับคืนมาเป็นแพคเก็ตใหญ่ได้ตามเดิม แต่ถ้าบิตนี้เป็น 0 หมายความว่าเป็นแพคเก็ตเดียวโดด ๆ หรืออาจจะเป็นแพคเก็ตย่อยตัวสุดท้ายของแพคเก็ตใหญ่ก็ได้ ซึ่งรายละเอียดจะต้องพิจารณาจากเขต fragment offset ประกอบด้วย อนึ่ง ตามข้อกำหนดของโปรโต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอล IP ถึงแม้ว่าอุปกรณ์หาเส้นทางสามารถแบ่งแพคเก็ตใหญ่ออกเป็นส่วนย่อยได้ก็ตาม แต่ไม่สามารถรวบรวมกลับคืนมาได้ ซึ่งหน้าที่นี้เป็นของสถานีปลายทางเท่านั้น

- Fragment offset (frag. Offset) มีขนาด 13 บิต ใช้ในการพิจารณาร่วมกับเขต identification , total length , D บิต และ M บิต เพื่อเรียงลำดับและหาขนาดความยาวของแพคเก็ตที่แท้จริงที่ถูกส่งออกมาจากสถานีต้นทางขณะที่มีการแบ่งแพคเก็ตเกิดขึ้น ค่าออฟเซตจะเปลี่ยนไป ดังนั้นเมื่อนำแพคเก็ตขนาดเล็กที่มีหมายเลขของ IP เดียวกันกับที่รับได้มาเรียงตามค่าของออฟเซตที่ถูกต้องโดยดู M บิตประกอบก็จะได้แพคเก็ตเดิมกลับคืนมา

- Time to live (TTL) มีขนาด 8 บิต เป็นเขตที่ใช้ในการกำหนดอายุของ IP แพคเก็ต โดยปกติอายุของแพคเก็ตเท่ากับ 255 (2^8-1) ทุกครั้งที่แพคเก็ตผ่านอุปกรณ์หาเส้นทาง ค่าของ TTL จะลดลงหนึ่งเสมอ ถ้าค่าของ TTL เป็นศูนย์ ก็หมายความว่าหมดอายุขัย เมื่ออุปกรณ์หาเส้นทางตัวใดพบสภาพเช่นนี้ก็จะนำแพคเก็ตนั้น ๆ ออกจากระบบทันที การกำหนดให้มีเขตนี้ก็เพื่อป้องกันการเกิดแพคเก็ตวนอยู่ในระบบตลอดกาล ซึ่งก่อให้เกิดความเสียหายแก่ระบบโดยรวม เช่น เป็นสาเหตุให้มีปริมาณแพคเก็ตในระบบมาก สถานีปลายทางอาจจะรับแพคเก็ตผิดได้ เป็นต้น

- Protocol มีขนาด 8 บิต เป็นเขตที่บอกชนิดโปรโตคอลที่ถูกห่อหุ้ม (encapsulate) โดยส่วนหัวของ IP แพคเก็ตซึ่งอาจจะเป็นโปรโตคอลชั้นที่สามหรือสี่ก็ได้

- Header checksum มีขนาด 16 บิต เป็นเขตที่ใช้ในการตรวจสอบความผิดพลาดเฉพาะส่วนหัวของ IP แพคเก็ต การคำนวนจะใช้วิธีการบวกเลข 16 บิต แบบ 1's complement เมื่อได้ผลลัพธ์แล้วก็ทำการผกผัน (inverse) ผลลัพธ์อีกครั้งหนึ่งจึงจะได้ checksum ที่แท้จริง การตรวจสอบกระทำเฉพาะส่วนหัวของข้อมูล เพราะส่วนข้อมูลของ IP แพคเก็ตจะเป็นโปรโตคอลชั้นที่สี่ซึ่งมี checksum เป็นของตัวเองอยู่แล้ว อนึ่ง checksum ของ IP แพคเก็ตถูกห่อหุ้มโดยโปรโตคอลชั้นที่สองที่ใช้ checksum ที่มีความเชื่อถือได้ เช่น อีเธอร์เนต ในกรณีที่ checksum ของ IP แพคเก็ตไม่ถูกใช้งานค่าของมันจะเป็นศูนย์ขนาด 16 บิต

- Source / Destination address มี ขนาดอย่างละ 32 บิต ซึ่งก็คือ IP แอดเดรส

- Option ส่วนของ Option นี้อาจจะมีหรือไม่มี หรือมีขนาดเท่าใดขึ้นอยู่กับชนิดของ IP แพคเก็ต เนื่องจากรายละเอียดของ Option ค่อนข้างปลีกย่อยจึงไม่ขอกล่าวในที่นี้

- Padding มีขนาด 0-3 ไบต์ ใช้เป็นส่วนที่ทำให้ขนาดของ option เป็นจำนวนเท่าของ 32 บิต

- Data เป็นส่วนของข้อมูลหรือโปรโตคอลที่อยู่ในชั้นที่สูงกว่าหรือเท่ากับ IP แพคเก็ต

2.8.2 โปรโตคอล TCP

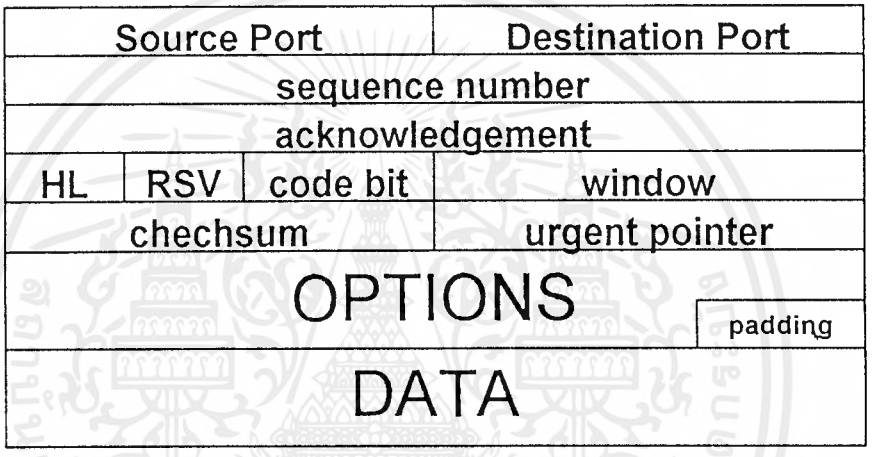
โปรโตคอล TCP เป็นโปรโตคอลที่อยู่ชั้นที่สี่ มีการให้บริการในแบบ "Connection - Oriented (CO)"ซึ่งหมายความว่าก่อนที่จะมีการรับส่งข้อมูลระหว่างต้นทางกับปลายทางจะต้องมีการสถาปนาการเชื่อมต่อขึ้นมาเสียก่อน และเมื่อสิ้นสุดการส่งผ่านข้อมูลก็จะยกเลิกการสถาปนานั้น ๆ

โปรโตคอล IP เป็น Connectionless (CL) เนื่องจากคณะผู้ออกแบบชุดโปรโตคอล TCP/IP เชื่อมั่นว่าระบบเครือข่ายที่ให้บริการในการส่งผ่านข้อมูลย่อมจะมีการผิดพลาดเกิดขึ้นได้เสมอ (ชุดโปรโตคอลนี้ถูกออกแบบในทศวรรษที่ 70 ซึ่งในขณะนั้นการสื่อสารข้อมูลยังมี Bit Error Rate (BER)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูง คือประมาณ 10^{-3} - 10^{-4}) และการทำให้แต่ละโหนดสถาปนาวงจรขึ้นมาทุกครั้งที่มีการรับส่งข้อมูลจากต้นทางไปยังปลายทางทำให้เกิดความยุ่งยากอย่างมากในระบบเครือข่าย

ด้วยสองสาเหตุหลักนี้ทำให้การออกแบบโปรโตคอลเลือกที่จะใช้ CL สำหรับ IP แพคเก็ต เมื่อโปรโตคอลในระดับล่างให้บริการในลักษณะเช่นนี้ ผู้ออกแบบต้องยอมรับสภาพการมีแพคเก็ตผิดพลาด แพคเก็ตซ้ำหรือแพคเก็ตสูญหาย เพื่อแก้ไขปัญหาเช่นนี้จึงเลือกใช้ CO สำหรับโปรโตคอลในชั้นที่อยู่เหนือกว่า ซึ่งโปรโตคอล TCP ก็เลือกใช้วิธีนี้ ส่วน UDP นั้นมีการให้บริการในลักษณะ CL ซึ่งรายละเอียดจะไม่ขอกล่าวถึง คำศัพท์ที่ใช้เรียก TCP แพคเก็ตคือ เซกเมนต์ (segment) ดังนั้นเมื่อกล่าวถึง TCP แพคเก็ตจะใช้คำว่าเซกเมนต์แทน สำหรับรายละเอียดเขตต่าง ๆ ของเซกเมนต์ TCP แสดงในรูปที่ 2.8 ซึ่งอธิบายได้ดังนี้



รูปที่ 2.8 โครงสร้างของโปรโตคอล TCP

- Source port มีขนาด 16 บิต การเข้าถึงแอพพลิเคชันของชุดโปรโตคอล TCP/IP ใช้หมายเลขพอร์ตเป็นตัวกำหนด ดังนั้น source port หมายถึง หมายเลขพอร์ตของสถานีต้นทาง

- Destination port มีขนาด 16 บิต เป็นหมายเลขพอร์ตของสถานีปลายทางที่ต้องการจะติดต่อด้วย แอพพลิเคชันมาตรฐานจะมีหมายเลขพอร์ตคงที่ เช่น Telnet ใช้พอร์ตหมายเลข 23, FTP (File Transfer Protocol) ใช้พอร์ตหมายเลข 20,21 เป็นต้น ตารางที่ 2 แสดงถึงบางส่วนของหมายเลขพอร์ตที่มีการใช้งาน สำหรับพอร์ตมาตรฐานถูกกำหนดไว้ที่หมายเลข 0-255 ในส่วนที่นอกเหนือจากนี้ผู้เขียนแอพพลิเคชันสามารถกำหนดเองได้

- Sequence number มีขนาด 32 บิต TCP เซกเมนต์ใช้หมายเลขลำดับนี้ในการบอกถึงตำแหน่งไบต์แรกของข้อมูลที่ปรากฏอยู่ ในส่วนของเขต data หมายความว่า ถ้าข้อมูล และส่วนหัวของ TCP เซกเมนต์ที่จะส่งมีความยาวเกินกว่าแพคเก็ตของ IP ที่จะยอมรับได้ (65,536 ไบต์) จึงจำเป็นต้องแบ่งข้อมูลที่จะส่งออกเป็น TCP เซกเมนต์ย่อย ๆ หลายเซกเมนต์ และเนื่องจากการมองข้อมูลของ TCP เซกเมนต์มองเป็น "stream" คือ การมองข้อมูลแบบต่อเนื่องกันไป เปรียบเสมือนการนับเลขจำนวน เมื่อข้อมูลในเซกเมนต์แรกถูกตัดตอนส่งที่ไบต์สุดท้ายตำแหน่งใด การเริ่มต้นของเซกเมนต์ใหม่ก็จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มในตำแหน่งของไบต์ที่อยู่ถัดมาเป็นเช่นนี้เรื่อย ๆ ไปจนสิ้นสุดข้อมูลทั้งหมด จะเห็นได้ว่าถ้าเลขลำดับมีลำดับที่ไม่ซ้ำ ลำดับของข้อมูลที่อยู่เรียงกันก็ย่อมไม่ซ้ำกัน

- Acknowledgement number มีขนาด 32 บิต โปรโตคอล TCP ใช้หลักการของ positive ack. ในการตอบกลับไปยังต้นทางเพื่อแสดงความปลายทางได้รับข้อมูลเรียบร้อยแล้ว โดยหมายเลขของ ack.no. จะต้องมีค่าเท่ากับ "ไบต์สุดท้ายบวกหนึ่ง" ของลำดับข้อมูลที่รับได้ (เช่น ไบต์สุดท้ายที่รับได้มีค่าเท่ากับ 1,024 เพราะฉะนั้นหมายเลขของ ack.no. จะต้องมีค่าเท่ากับ 1,025 (1,024+1) ซึ่งเป็นการบอกต้นทางอย่างเป็นนัยว่า "รับไบต์สุดท้ายได้แล้ว คาดว่าไบต์เริ่มต้นต่อไปที่จะได้รับคือไบต์สุดท้ายบวกหนึ่ง"

- Header length (HL) มีขนาด 4 บิต เป็นการบอกถึงความยาวส่วนหัวของ TCP เซกเมนต์ มีหน่วยเป็นจำนวนเท่าของ 4 ไบต์เช่นเดียวกับ IP แพคเก็ต หรืออีกนับหนึ่งก็คือเป็นการบอกถึงตำแหน่งเริ่มต้นของไบต์ข้อมูล เพราะส่วนหัวของ TCP เซกเมนต์มีความยาวที่เปลี่ยนแปลงได้ขึ้นอยู่กับเขต option โดยทั่วไปถ้าไม่มีในส่วนของ option ค่าของ HL จะเท่ากับ 5 (5x4=20 ไบต์)

- Reserved (RSV) มีขนาด 4 บิต เป็นส่วนที่ถูกสำรองไว้ใช้ในอนาคต

- Code bit ประกอบด้วย 6 เขต ย่อย ซึ่งแต่ละเขตย่อยมีขนาด 1 บิต และมีรายละเอียด ดังนี้

a) เขตย่อย URGent ถ้าบิตนี้เป็น "1" หมายความว่าเขต Urent pointer บรรจุข่าวสารที่มีความหมายจะต้องนำมาตีความด้วย การมองข้อมูลของ TCP เซกเมนต์ ว่ามีความเร่งด่วน (urgent) นั่นหมายความว่าข้อมูลนั้นไม่ใช่ข้อมูลธรรมดา แต่มีความสำคัญต่อระบบทั้งหมด ดังนั้นเมื่อปลายทางรับข้อมูลประเภทนี้ได้ต้องรีบให้ความสนใจทันที

b)เขตย่อย ACKnowledgement ถ้าบิตนี้เป็น "1" หมายความว่าเขต ACKnowledgement number มีความหมายต้องนำมาพิจารณาในการตรวจสอบลำดับในการรับส่งข้อมูลด้วย

c) เขตย่อย PUSH ถ้าบิตนี้เป็น "1" หมายความว่า ทันทีที่สถานีปลายทางรับ TCP เซกเมนต์ได้ต้องรีบส่งข้อมูลนี้ไปยังแอพพลิเคชันทันทีโดยไม่ต้องรอให้บัฟเฟอร์ในการรับข้อมูลเต็มเสียก่อน โดยทั่วไปแล้วการส่งข้อมูล จาก TCP ขึ้นไปยังแอพพลิเคชันต้องรอให้ข้อมูลเข้ามาอยู่ในบัฟเฟอร์เต็มเสียก่อนเพื่อลดจำนวน overhead ที่จะเกิดขึ้นในการส่งโปรโตคอลระหว่างชั้น

d) เขตย่อย RESET ถ้าบิตนี้เป็น "1" หมายความว่า มีการผิดพลาดเกิดขึ้นระหว่างคู่สถานีที่มีการเชื่อมต่อถึงกันอยู่และจำเป็นต้องยกเลิกการเชื่อมต่อนี้ แต่ถ้าต้องการส่งข้อมูลถึงกันอีกก็ต้องสถาปนาการเชื่อมต่อขึ้นมาใหม่

e) เขตย่อย SYNchronize ถ้าบิตนี้เป็น "1" หมายความว่าขอเริ่มต้นสถาปนาการเชื่อมต่อ และเมื่อการสถาปนาเสร็จสิ้น บิตนี้จะถูกกำหนดให้เป็น "0" และหลังจากนั้นก็สามารถส่งผ่านข้อมูลระหว่างสถานีได้

f) เขตย่อย FIN ถ้าบิตนี้เป็น "1" หมายความว่า ขอยกเลิกการเชื่อมต่อระหว่างสถานีเพราะไม่มีข้อมูลที่จะต้องส่งถึงกันอีกแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Window มีขนาด 16 บิต สถานีปลายทางใช้เขตนี้ในการบ่งบอกถึงขนาดของบัฟเฟอร์ (จำนวนไบต์)ที่สามารถรับข้อมูลจากต้นทางได้ หรือเป็นการบอกว่า "ถ้าต้นทางส่งปริมาณข้อมูลมามากกว่าที่ปลายทางจะรับได้ อาจมีบางส่วนของข้อมูลสูญหายได้" ข้อดีของเขตนี้ก็คือ เป็นการควบคุมปริมาณข้อมูลในระบบไม่ให้มีมากเกินไป ซึ่งอาจจะทำให้เกิด traffic jam ในระบบเครือข่ายได้ ในโปรโตคอลชั้นที่สอง เช่น HDLC อาจจะมีคำถามว่า window ของ TCP แตกต่างอย่างไรเมื่อเปรียบเทียบกับ HDLC ความแตกต่างที่เห็นได้ชัดคือ window ของ TCP สามารถเปลี่ยนแปลงขนาดได้ระหว่างการส่งผ่านข้อมูลครั้งหนึ่ง ๆ เพราะที่เวลาหนึ่ง ๆ สถานีปลายทางอาจจะมีขนาดของบัฟเฟอร์ไม่คงที่ แต่ window ของ HDLC จะมีขนาดคงที่ตลอดช่วงการส่งผ่านข้อมูล เพราะขณะที่เริ่มต้นสถาปนาการเชื่อมต่อในระดับโปรโตคอลชั้นที่สอง จะมีการตกลงระหว่างสถานีต้นทาง และปลายทางว่าต้องการขนาดของ window เป็นเท่าใด ซึ่งมีให้เลือกสองขนาดคือ modulo - 8 และ modulo - 128

- Checksum มีขนาด 16 บิต ใช้วิธีการเช่นเดียวกับ IP แพคเก็ต แต่จะมีส่วนที่นำมาใช้ในการคำนวณเพิ่มเติมนอกเหนือไปจากส่วนหัวของ TCP เซกเมนต์

- Urgent pointer มีขนาด 16 บิต เปรียบเสมือนเป็นค่า offset ของ sequence number โดยจะบอกถึงตำแหน่งของไบต์สุดท้ายภายใน TCP เซกเมนต์ที่เป็นข้อมูลเร่งด่วน (ข้อมูลภายใน TCP เซกเมนต์หนึ่ง ๆ อาจจะไม่มีความเร่งด่วนทั้งหมดก็ได้) ที่ต้องการให้แอพพลิเคชันพิจารณาทันที อนึ่งค่าที่ถูกบรรจุในเขตนี้จะมีความหมายก็ต่อเมื่อเขตย่อย urgent ถูกกำหนดค่าเป็น "1"

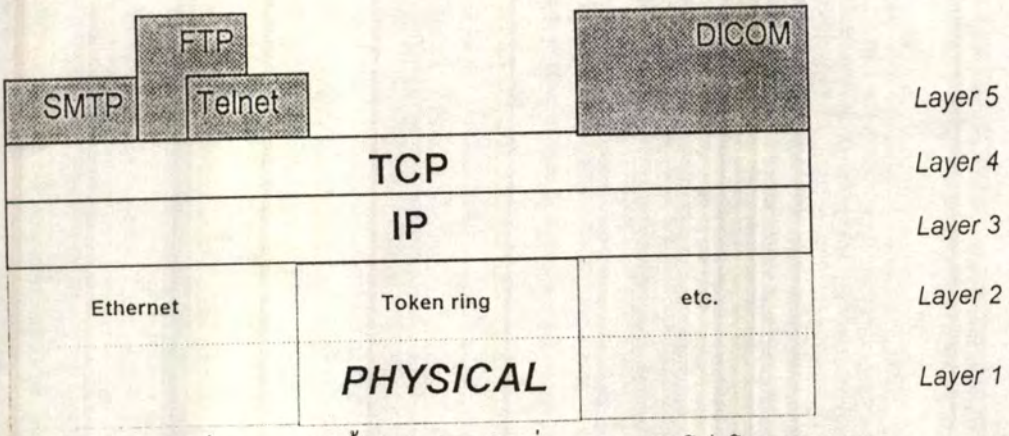
- Option ส่วนของ Option นี้อาจจะมีหรือไม่มี หรือมีขนาดเท่าใด ขึ้นอยู่กับชนิดของ TCP เซกเมนต์

- Padding มีขนาด 0-3 ไบต์ ใช้เป็นส่วนที่ทำให้ขนาดของ option เป็นจำนวนเท่าของ 32 บิต

2.8.3 ความสัมพันธ์และ การทำงานร่วมกันของ โปรโตคอลแต่ละชั้น

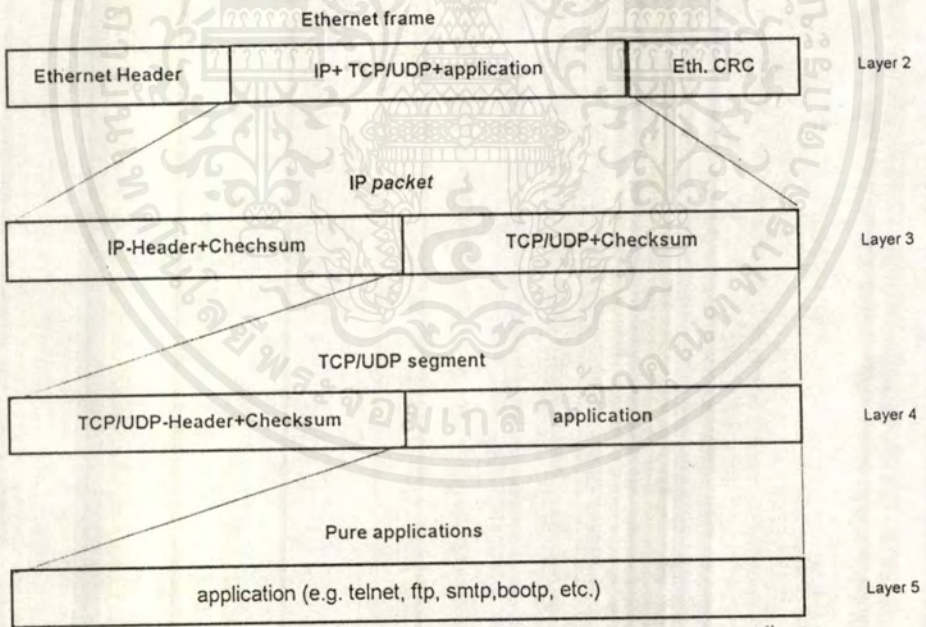
ตัวอย่างลำดับชั้นของชุดโปรโตคอล TCP/IP และแอพพลิเคชัน แสดงในรูปที่ 2.9 จากรูปจะเห็นได้ว่าแอพพลิเคชันแต่ละประเภทเลือกใช้โปรโตคอลชั้นที่ 4 เช่น SMTP เลือกใช้ TCP ในที่นี้ DICOM Upper Layer ก็ใช้ TCP และต่อมาก็คือการใช้โปรโตคอล IP และถัดจากโปรโตคอลชั้นที่ 3 ลงมาก็ขึ้นอยู่กับว่าผู้ใช้งานเลือกใช้เครือข่ายประเภทใดในการส่งผ่านชุดโปรโตคอล TCP/IP ถ้าเป็น LAN ก็อาจจะเลือกใช้อีเธอร์เนต หรือ LAN แบบอื่นที่เห็นว่าสมควร แต่ถ้าเป็น WAN ก็อาจใช้ PPP (Point-to-Point Protocol) หรือ HDLC เป็นต้น สำหรับ ชั้น กายภาพ นั้นขึ้นอยู่กับว่าเลือกใช้ LAN แบบใด เพราะ LAN แต่ละแบบจะมีอุปกรณ์เชื่อมต่อต่างกัน สำหรับ WAN นั้นมีให้เลือกหมายมาตรฐาน เช่น RS232C, CCITT V.35 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ลำดับชั้นของแอพพลิเคชั่นและชุดของโปรโตคอล TCP/IP

ในโครงงานนี้ใช้ LAN แบบอีเธอร์เนตประกอบการอธิบายรวมไปถึงในการทดลองด้วย อย่างไรก็ตามก็สามารถประยุกต์กับ LAN แบบอื่นๆได้ ถ้าเขียนความสัมพันธ์ของโปรโตคอลชั้นต่าง ๆ จะได้ดังรูปที่ 2.10 จะเห็นได้ว่าโปรโตคอลชั้นที่อยู่บนสุดถูกห่อหุ้มด้วยชั้นที่อยู่ถัดมาเป็นเช่นนี้เรื่อย ๆ ไปจนถึงชั้นที่สองซึ่งเป็นการสิ้นสุดการห่อหุ้มส่วนชั้นกายภาพนั้นก็ขึ้นอยู่กับว่าสารสนเทศจะถูกส่งออกไปในรูปแบบของสัญญาณไฟฟ้าอย่างไร



รูปที่ 2.10 โปรโตคอลในชั้นบนสุดจะถูกห่อหุ้มด้วยโปรโตคอลในชั้นที่ต่ำกว่า

จากรูปที่ 2.11 ถ้าสถานี A ต้องการต่อเข้ากับไฟล์เซอร์ฟเวอร์ (file server)ในลักษณะของเทอร์มินัลระยะไกล (remote terminal) โดยใช้แอพพลิเคชัน "telnet 158.108.1.20" นั้นมีเหตุการณ์อย่างไร

ขั้นตอนแรกสถานี A ต้องการสถาปนาการเชื่อมต่อกับไฟล์เซอร์ฟเวอร์ จะเห็นได้ว่าแอดเดรสต้นทาง (ทั้งอีเธอร์เนต และ IP แอดเดรส) เป็นของสถานี A ส่วนแอดเดรสปลายทางเป็นของไฟล์

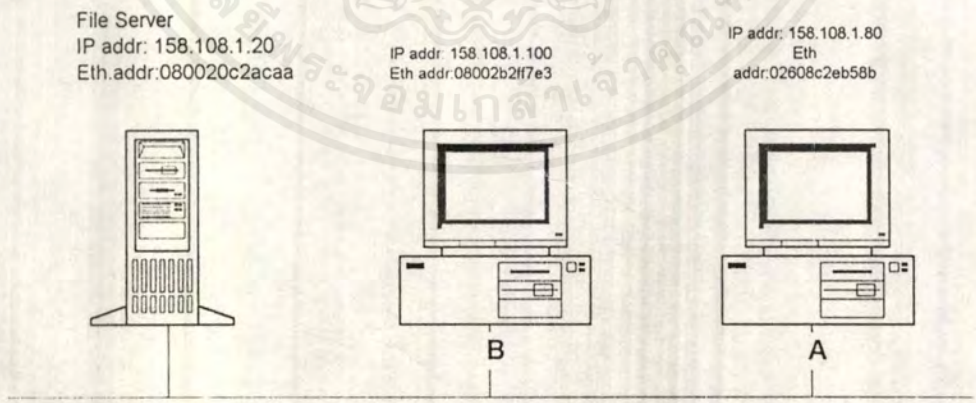
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซอร์ฟเวอร์ สำหรับหมายเลขพอร์ตต้นทางนั้น สถานี A จะทำการสุ่มเลขจำนวนเต็มบวกใด ๆ ขึ้นมา (ห้ามน้อยกว่า 255) ที่ไม่ซ้ำกับหมายเลขพอร์ตที่ปรากฏอยู่บนสถานี A ทั้งหมดในขณะนั้น เพื่อใช้เป็นจุดอ้างอิงของแอพพลิเคชัน แต่สำหรับหมายเลขพอร์ตปลายทางจะเป็นเบอร์ "$(23)_{10}$" เท่านั้น ถ้าเป็นเบอร์อื่นเช่น "$(53)_{10}$" จะกลายเป็น DOMAIN ซึ่งไม่ใช่ TELNET เนื่องจากแอพพลิเคชัน "telnet" ใช้บริการของ TCP ดังนั้นหมายเลขโปรโตคอลในส่วนของ IP แพคเก็ตจึงเป็นเบอร์ "$(6)_{10}$" และเนื่องจากอีเธอร์เนตให้บริการแก่ IP แพคเก็ต ดังนั้นส่วนของ frame type จึงมีค่าเท่ากับ $(0800)_{16}$ จะเห็นได้ว่าอีเธอร์เนตแอดเดรสของทั้งต้นทางและปลายทางเป็นแอดเดรสแบบเฉพาะเจาะจง ถึงแม้ว่าสถานี B รับอีเธอร์เนตเฟรมได้ก็จะไม่สนใจเพราะตัวมันเองไม่มีส่วนเกี่ยวข้องกับเฟรมนี้

ขั้นตอนที่สอง เมื่อไฟล์เซอร์ฟเวอร์ได้รับการขอสถาปนาการเชื่อมต่อจากสถานี A จึงตอบตกลงกลับมายังสถานี A แอดเดรสต้นทาง ปลายทาง และหมายเลขพอร์ตมีการสลับกันกับเกิดขึ้น เพราะขณะนี้เรามองสารสนเทศที่ถูกส่งจากไฟล์เซอร์ฟเวอร์ซึ่งเป็นต้นทางไปยังสถานี A ซึ่งเป็นปลายทาง

สถานี A และไฟล์เซอร์เวอร์จะสื่อสารและโต้ตอบกันอีกหลายแพคเก็ตซึ่งเป็นไปตามแผนภาพสถานะ (state diagram) ของโปรโตคอล TCP/IP เมื่อสถานี A ได้รับ "prompt" จากไฟล์เซอร์ฟเวอร์ ก็จะใส่ชื่อของผู้ใช้งานลงไป เช่น "login:s8013084" เมื่อไฟล์เซอร์ฟเวอร์ได้รับชื่อก็ส่งถามรหัสผ่านกลับมายังผู้ใช้งานอีก ผู้ใช้งานก็ต้องใส่รหัสผ่านของตัวเองลงไป เช่น "password:first98" เมื่อไฟล์เซอร์ฟเวอร์ตรวจสอบความถูกต้องแล้ว ผู้ใช้งานก็สามารถเข้าใช้ทรัพยากรของไฟล์เซอร์ฟเวอร์ได้จากสถานี A ตามที่ผู้ดูแลระบบจัดเตรียมไว้

สำหรับการทำงานด้วยโปรโตคอลชั้นบนแบบ DICOM กับ TCP/IP นั้นก็มีหลักการเช่นเดียวข้างต้นเพียงแต่ขั้นตอนการทำงานนั้นเป็นไปตามข้อกำหนด DICOM นั่นเอง



รูปที่ 2.11 สถานี A และ B ถูกกำหนดให้อยู่ในเครือข่ายเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
039036

บทที่ 3
การออกแบบและโครงสร้างของโปรแกรม

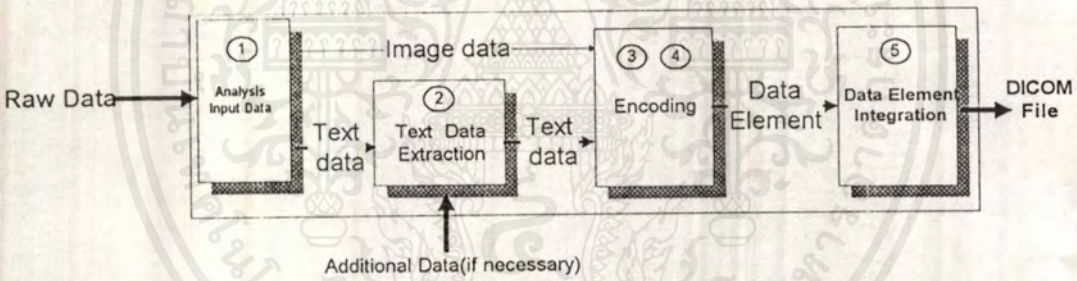
3.1 บทนำ

ในบทนี้กล่าวถึงการพัฒนางาน 2 ส่วนคือการพัฒนาตัวแปลงข้อมูลจากแฟ้มข้อมูลภาพทั่วไปไปเป็นข้อมูลแบบ DICOM และการพัฒนาการสื่อสารข้อมูลแบบ DICOM

3.2 โครงสร้างส่วนของการแปลงข้อมูลให้เข้าสู่โครงสร้างข้อมูลแบบ DICOM

การแปลงข้อมูลให้อยู่ในโครงสร้างข้อมูลแบบ DICOM มีขั้นตอนดังนี้

- การวิเคราะห์และแบ่งแยกประเภทของข้อมูล
- การรับข้อมูลอักษรเพิ่มเติม
- การจัดข้อมูลที่มีเข้าสู่แบบจำลองข้อมูล
- การเข้ารหัสข้อมูลให้อยู่ในรูปแบบ DICOM
- การจัดเก็บข้อมูลเป็นแฟ้มแบบ DICOM



รูปที่ 3.1 ขั้นตอนการแปลงข้อมูลให้อยู่ในโครงสร้างข้อมูลแบบ DICOM

3.1.1 การวิเคราะห์และแบ่งแยกประเภทของข้อมูล

ขั้นตอนนี้ทำการวิเคราะห์ข้อมูลต้นแบบที่จะนำมาทำการแปลงว่ามีลักษณะอย่างไร เช่นมีการบีบอัดมาหรือไม่ หรือมีลักษณะการจัดเก็บอย่างไร ขั้นตอนนี้จะแตกต่างกันไปตามแต่ข้อมูลต้นแบบ ส่วนนี้จึงเป็นส่วนที่ต้องเปลี่ยนแปลงเมื่อข้อมูลต้นแบบเปลี่ยนไป การทำงานเริ่มจากแบ่งแยกข้อมูลออกเป็นประเภทคือ ข้อมูลประเภทที่เป็นตัวอักษรเช่นข้อมูลประวัติคนไข้ และข้อมูลที่มาเป็นรหัสไบนารี่เช่นข้อมูลภาพจากเครื่องมือแพทย์ชนิดต่างๆ ในขั้นตอนนี้จะทำการถอดนำข้อมูลตัวอักษรออกมาให้อยู่ในรูปแบบรหัสแอสกี ส่วนข้อมูลภาพหากไม่ได้อยู่ในรูปของข้อมูลจุดภาพ (Pixel Data) ก็ต้องทำการแจ้งให้กับโปรแกรมหลักเพื่อเตรียมการถอดรหัสภาพ ในขั้นนี้จะนำเพียงแค่นำคุณสมบัติของภาพเช่น ความกว้าง ความยาว จำนวนภาพ จำนวนบิทที่จัดเก็บต่อ 1 หน่วยจุดภาพ เป็นต้น ข้อมูลคุณสมบัติภาพจะถูกเก็บเป็นข้อมูลอักษรเช่นเดียวกับข้อมูลคนไข้ สาเหตุที่ยังไม่ทำการถอดรหัสทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมดของข้อมูลภาพออกมาเป็นข้อมูลจุดในขั้นตอนนี้เนื่องมาจากต้องการประหยัดพื้นที่ในการทำงานในฮาร์ดดิสก์

3.1.2 การรับข้อมูลอักษรเพิ่มเติม

ขั้นตอนนี้มีไว้สำหรับให้ผู้ใช้ใส่ข้อมูลเพิ่มเติม ขั้นตอนนี้มีความสำคัญกับภาพต้นแบบที่ไม่มีประวัติหรือข้อมูลการรักษาติดมาด้วยหรือมาแต่ไม่ได้อยู่ในรูปแบบข้อมูลดิจิตอล สำหรับภาพต้นแบบที่มีข้อมูลติดมาเพียงแล้วอาจจะให้ผู้ใช้ข้ามขั้นตอนนี้ไปได้

3.1.3 การจัดข้อมูลที่มีเข้าสู่แบบจำลองข้อมูล

นำข้อมูลตัวอักษรที่ทำการถอดรหัสได้จากขั้นตอนที่ 1 หรือจากการนำเข้ามาใหม่ไปทำการจัดลง ให้ตรงตามรูปแบบของแบบจำลองข้อมูล DICOM จากนั้นทำการจัดเรียงข้อมูลที่จะนำไปยังขั้นตอนต่อไป ในการเขียนโปรแกรมคือการส่งค่าหมายเลขประจำและข้อมูลเข้าไปในโมดูลการเข้ารหัสตามลำดับของหมายเลขประจำจนครบ สำหรับข้อมูลภาพจะอยู่เป็นข้อมูลลำดับสุดท้ายและในขั้นนี้ข้อมูลจุดภาพก็ยังไม่ถูกแยกออกมา

3.1.4 การเข้ารหัสข้อมูลให้อยู่ในรูปแบบ DICOM

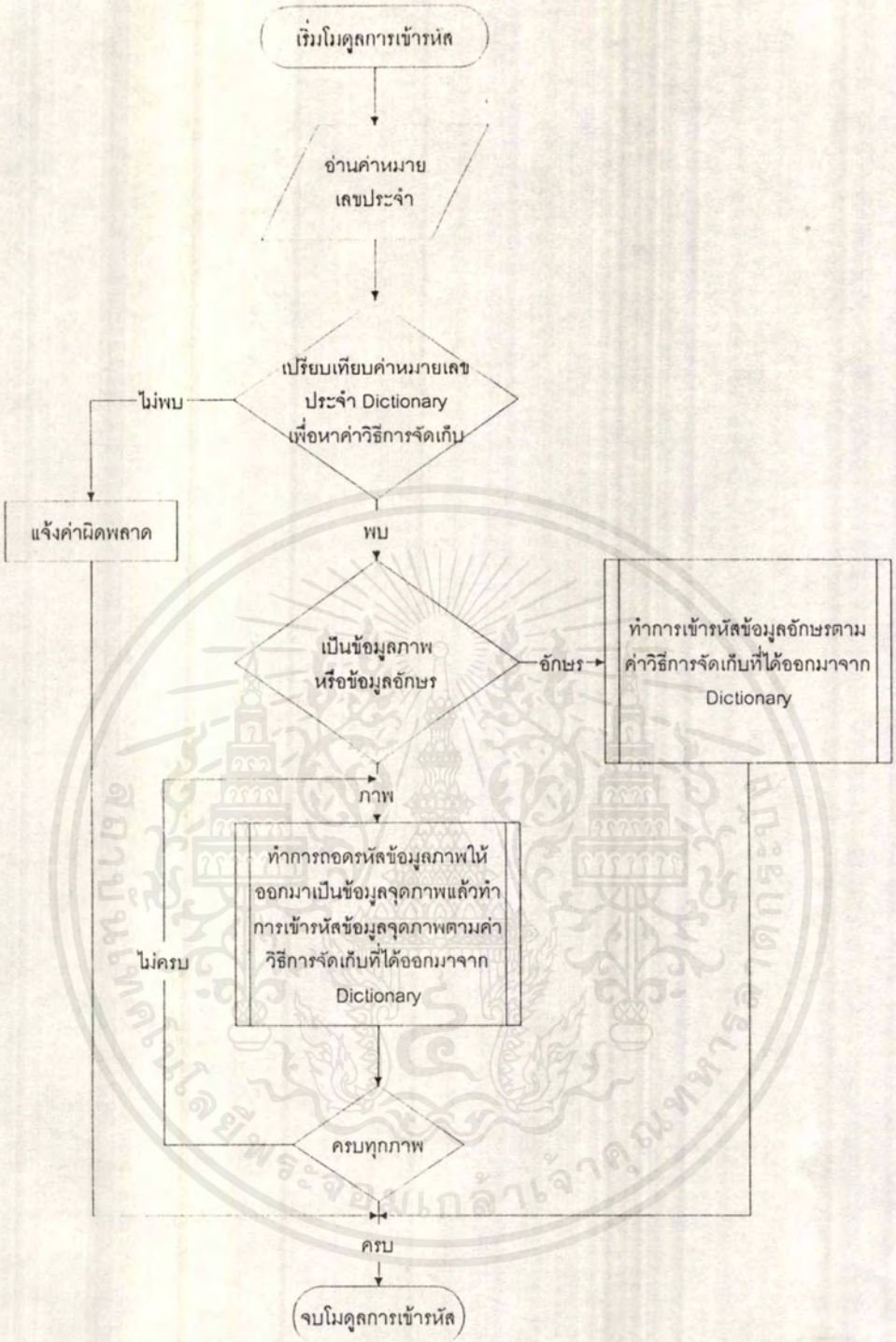
ขั้นตอนนี้จะทำการนำค่าหมายเลขประจำที่ได้มาจากขั้นตอนที่แล้วเข้าไปเปรียบเทียบในโปรแกรมส่วน Dictionary[10] ที่ถูกสร้างขึ้นตามมาตรฐานดังรูปที่ 3.2 เพื่อที่จะได้ค่าของวิธีการจัดเก็บออกมา ค่านี้จะเป็นตัวบอกว่าจะทำอย่างไรกับข้อมูลที่เข้ามาพร้อมกับหมายเลขประจำนั้นๆ เมื่อได้ค่าวิธีการจัดเก็บมาแล้วก็ไปส่งค่าหมายเลขประจำ ค่าวิธีการจัดเก็บและข้อมูลไปเข้ารหัสออกมาเป็น Data Element โดยสมบูรณ์

3.1.5 การทดสอบความถูกต้องของการแปลง

ข้อมูลที่ถูกต้องหมายถึงข้อมูลที่ผ่านการแปลงแล้วต้องมีการจัดการโครงสร้างตรงกับแบบจำลองข้อมูลแบบ DICOM แสดงเป็นรูปแบบเมื่อเก็บเป็นแฟ้มในรูปที่ 3.3 และรายละเอียดของข้อมูลที่ใส่เข้าไปครบกล่าวคือถ้าเป็นข้อมูลตัวอักษรก็ต้องอ่านได้ครบตามที่ใส่เข้าไป ส่วนข้อมูลภาพก็ต้องมีข้อมูลจุดภาพเดียวกันกับภาพก่อนที่จะทำการแปลง ขั้นตอนการตรวจสอบข้อมูลจุดภาพคือ

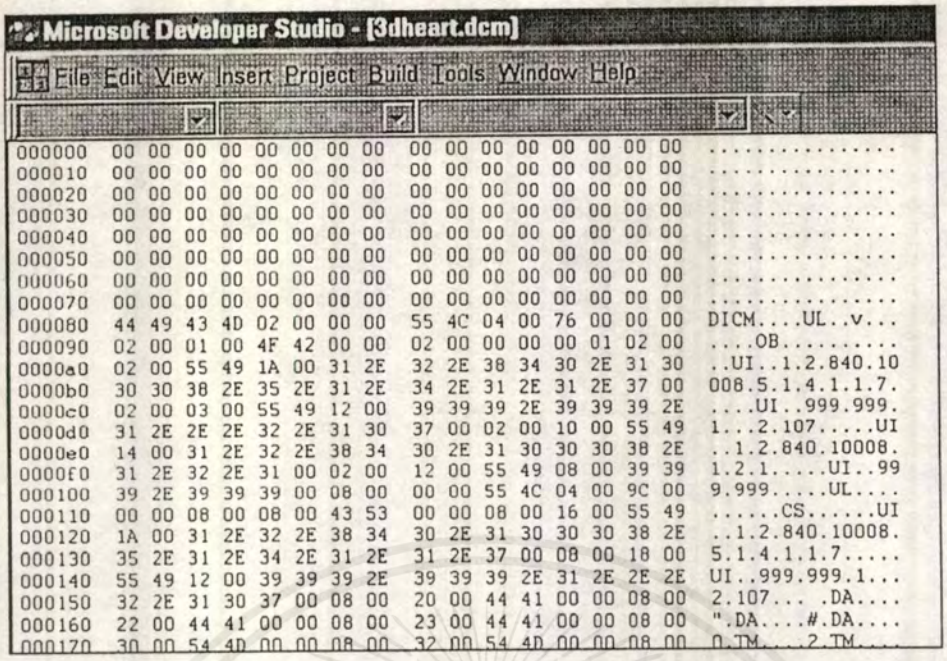
1) อ่านข้อมูลจุดภาพต้นฉบับแล้วเก็บเป็นแฟ้ม source.raw
2) อ่านข้อมูลภาพที่ผ่านการแปลงแล้วเก็บเป็นแฟ้ม target.000
3) ทำการเปรียบเทียบข้อมูลในแฟ้มทั้งสองโดยการอ่านข้อมูลจากทั้งสองแฟ้มแล้วนำมาลบกัน หรือใช้คำสั่ง fc (File compare) ของ DOS
4) ถ้าไม่มีข้อแตกต่างหมายถึงการแปลงภาพไม่ได้เปลี่ยนแปลงข้อมูลจุดภาพ หรือกล่าวได้ว่าไม่มีการเปลี่ยนแปลงข้อมูลภาพในกระบวนการแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



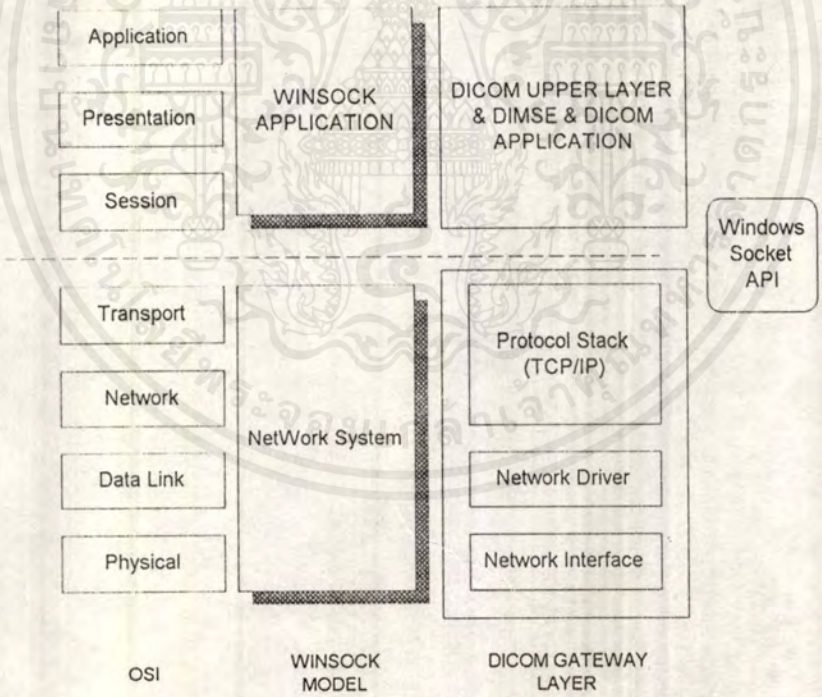
รูปที่ 3.2 ผังงานการเข้ารหัสข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000080	44	49	43	4D	02	00	00	00	55	4C	04	00	76	00	00	00	DICM....UL..v...
000090	02	00	01	00	4F	42	00	00	02	00	00	00	00	01	02	00	....OB.....
0000a0	02	00	55	49	1A	00	31	2E	32	2E	38	34	30	2E	31	30	..UI..1.2.840.10
0000b0	30	30	38	2E	35	2E	31	2E	34	2E	31	2E	31	2E	37	00	008.5.1.4.1.1.7.
0000c0	02	00	03	00	55	49	12	00	39	39	39	2E	39	39	39	2E	....UI..999.999.
0000d0	31	2E	2E	2E	32	2E	31	30	37	00	02	00	10	00	55	49	1...2.107.....UI
0000e0	14	00	31	2E	32	2E	38	34	30	2E	31	30	30	30	38	2E	..1.2.840.10008.
0000f0	31	2E	32	2E	31	00	02	00	12	00	55	49	08	00	39	39	1.2.1.....UI..99
000100	39	2E	39	39	39	00	08	00	00	00	55	4C	04	00	9C	00	9.999.....UL....
000110	00	00	08	00	08	00	43	53	00	00	08	00	16	00	55	49	......CS......UI
000120	1A	00	31	2E	32	2E	38	34	30	2E	31	30	30	30	38	2E	..1.2.840.10008.
000130	35	2E	31	2E	34	2E	31	2E	31	2E	37	00	08	00	18	00	5.1.4.1.1.7.....
000140	55	49	12	00	39	39	39	2E	39	39	39	2E	31	2E	2E	2E	UI..999.999.1...
000150	32	2E	31	30	37	00	08	00	20	00	44	41	00	00	08	00	2.107... .DA....
000160	22	00	44	41	00	00	08	00	23	00	44	41	00	00	08	00	".DA....#.DA....
000170	30	00	54	4D	00	00	08	00	32	00	54	4D	00	00	08	00	0.TM....2.TM....

รูปที่ 3.3 โครงสร้างแฟ้มข้อมูลหลังผ่านการแปลงเมื่อแสดงในรูปแบบเลขฐาน 16



รูปที่ 3.4 การวางตัวของชั้นโปรโตคอลส่วนบนแบบ DICOM

3.3 การพัฒนาการสื่อสารข้อมูล

การวางตัวของ DICOM Upper Layer Protocol รูปที่ 3.4 เมื่อสนับสนุนการทำงานบนโปรโตคอลชั้นขนส่งแบบ TCP/IP ในการพัฒนาใช้มาตรฐาน Windows Socket Specification กล่าวคือการติดต่อสื่อสารไปยัง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอลชั้นขนส่งให้ติดต่อผ่าน Winsock API นั่นคือการส่งผ่านโครงสร้างข้อมูลและชุดคำสั่งแบบ DICOM ผ่าน Winsock API ไปยังโปรโตคอลชั้นขนส่งนั่นเอง

3.3.1 ขั้นตอนการติดต่อกับโปรโตคอลชั้นขนส่ง

การติดต่อกับโปรโตคอลชั้นขนส่งเป็นขั้นตอนแรกของการเริ่มต้นการสื่อสาร ทำการติดต่อโดยใช้วิธีตามมาตราฐานการเข้าถึงของ Window socket Specification โดยขอการเชื่อมต่อเพื่อให้ได้หมายเลขซอกเก็ตแล้วนำซอกเก็ตนั้นไปใช้เป็นช่องทางการสื่อสารนั่นเอง

3.3.2 การพัฒนาชุดคำสั่ง

การเริ่มต้นการสื่อสารจะเริ่มต้นด้วยการอุปกรณ์ที่เป็นตัวของใช้บริการจะส่งโครงสร้างข้อมูลที่เรียกว่า A-ASSOCIATE-RQ (A-ASSOCIATE-REQUEST) ไปยังตัวให้บริการ แล้วรอรับโครงสร้างข้อมูลที่เรียกว่า A-ASSOCIATE-AC (A-ASSOCIATE-ACCEPT) ถ้าหากไม่ได้รับโครงสร้างข้อมูลนี้ก็จะยกเลิกการสื่อสารทั้งหมด หรือทำการขอการเชื่อมต่อใหม่จนกว่าจะได้รับการตอบรับ เมื่อได้รับการตอบรับแล้วจึงทำการสร้างชุดคำสั่งตามแต่ที่ต้องการ ชุดคำสั่งต่างๆถูกกำหนดไว้ในข้อกำหนด DICOM ส่วนที่ 4 (Part 4: Service Class Specification) โครงสร้างของชุดคำสั่งก็มีลักษณะเดียวกับโครงสร้างข้อมูลแบบ DICOM ดังแสดงไว้ในบทที่ 2 เมื่อสร้างชุดคำสั่งเสร็จแล้วก็ทำการส่งผ่านโครงสร้างข้อมูลที่เรียกว่า PDV ผ่านลงไปยัง Winsock API เพื่อดำเนินการส่งไปยังระบบเครือข่ายต่อไป ถ้าต้องการยกเลิกการเชื่อมต่อก็ให้ส่งโครงสร้าง A-RELEASE-REQUEST ไปยังตัวให้บริการ สำหรับผังงานของทั้ง 3 ส่วนแสดงในรูปที่ 3.5, 3.6, 3.7 ตามลำดับ

ในโครงงานนี้ทำการทดลองชุดคำสั่ง 2 แบบ คือ Verification Service Class หรือคำสั่งการตรวจสอบการเชื่อมต่อกับ DICOM Server กับ Storage Service Class User หรือคำสั่งการของส่งข้อมูลภาพไปเก็บที่ DICOM Server การทำงานของชุดคำสั่งเขียนเป็นคำสั่งจำลองได้คือ

หมายเหตุ สำหรับโครงสร้างชุดคำสั่งโดยละเอียดหาได้จาก PART 7 : Message Exchange[11]

1). Verification Service Class

```
/* send Echo to DICOM Server */
INITIALIZE Socket                    /* Initial TCP/IP Service /
   IF FAIL
        GOTO END
SEND  A-ASSOCIATE-RQ (Socket)
LOOP  wait for A-ASSOCIATE-RQ RSP
IF  TIME OUT
   GOTO  END
IF  A-ASSOCAITE-RQ AC
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

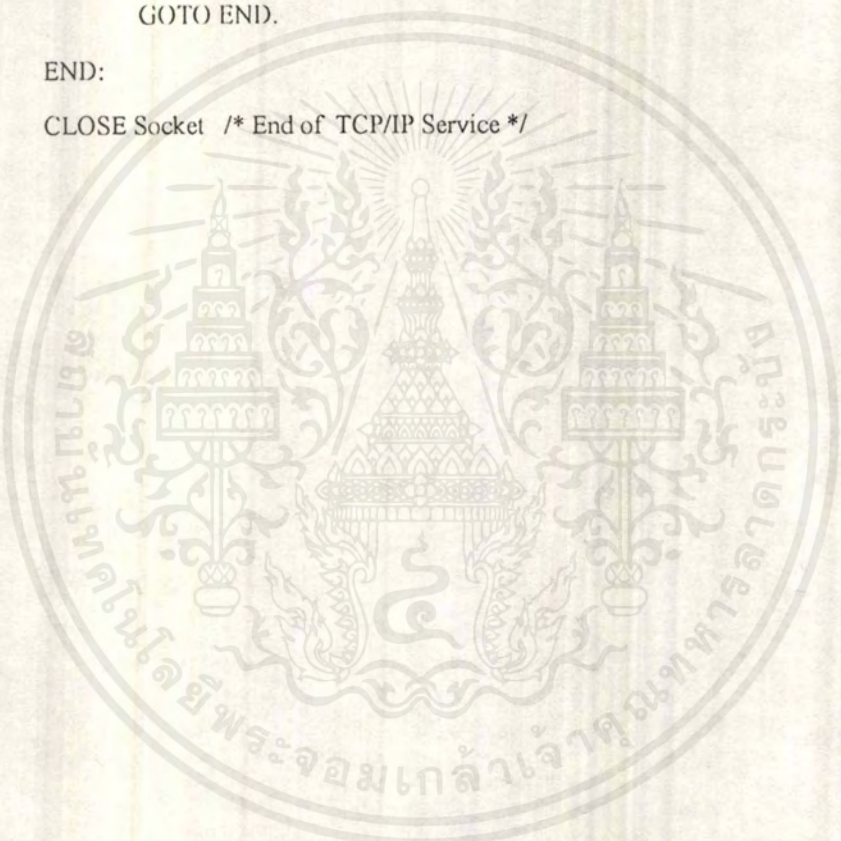
```
            CONSTRUCT PDV
            SEND Verification Structure by PDV(Socket)
            LOOP wait Verification RSP
            SHOW RSP Message
            SEND A-RELEASE-RQ(Socket)
            GOTO END.
    }
    IF  A-ASSOCAITE-RQ RJ
    GOTO END.

    END:
    CLOSE  Socket   /* End of  TCP/IP Service */
```

2) Storage Service Class User

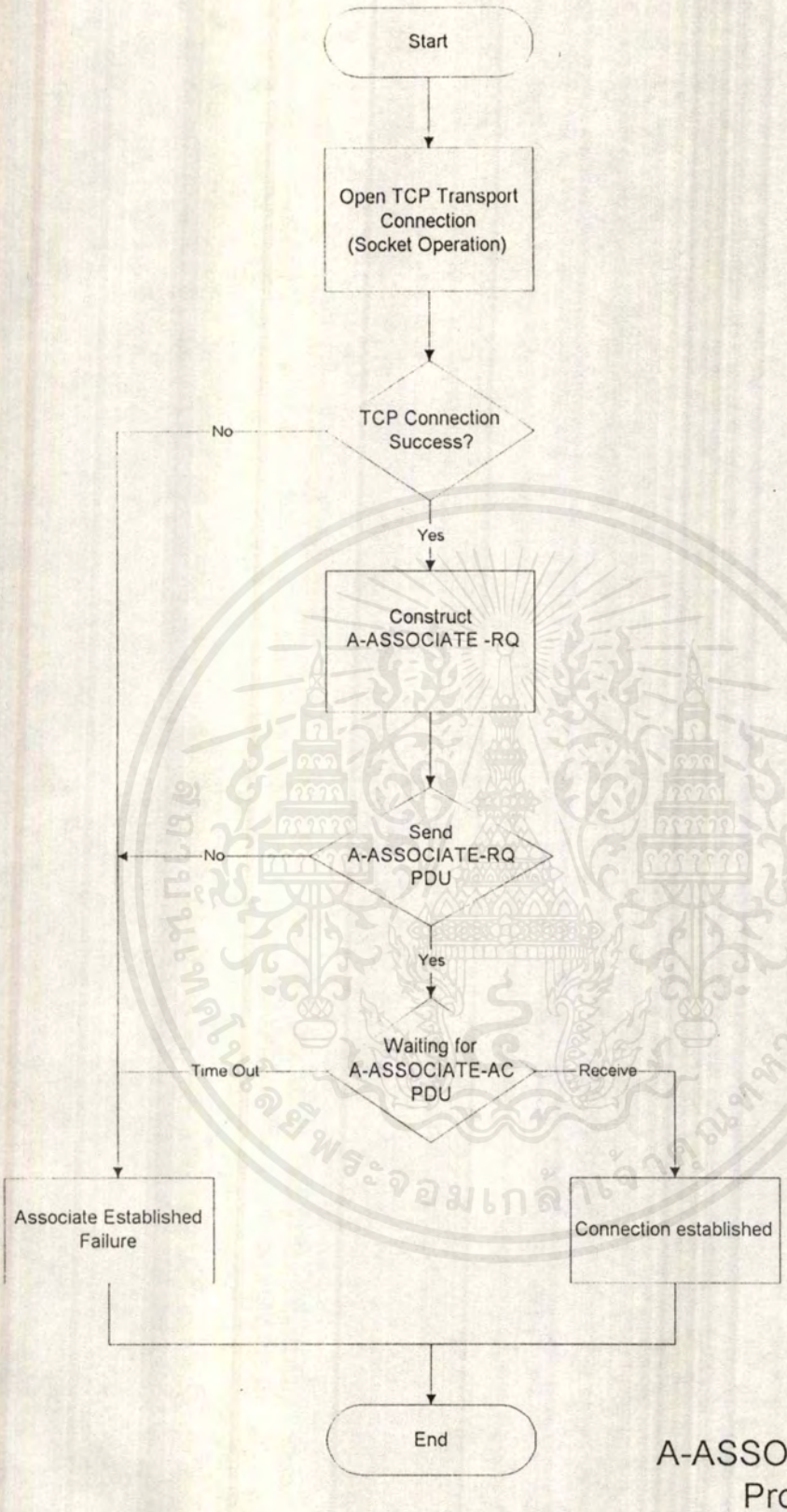
```
    /* Send DICOM Object  to DICOM Server ( XXX.DCM ) */
    INITIALIZE  Socket          /* Initial TCP/IP Service /
    IF FAIL
            GOTO END
    OPEN XXX.DCM          /* Open DICOM File Set to send */
    IF FAIL
            GOTO END
    SEND A-ASSOCIATE-RQ
    LOOP  wait for A-ASSOCIATE-RQ RSP
    IF  TIME OUT
       GOTO  END
    IF A-ASSOCAITE-RQ AC
    {
            CONSTRUCT PDV
            SEND Storage Service User  Class by PDV
            LOOP wait Storage Service User  Class  RSP
            SHOW RSP
            IF Storage Service User  Class  RSP -AC
            {
                    READ  XXX.DCM
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CONSTRUCT PDV
                SEND XXX.DCM by PDV
                LOOP End of XXX.DCM
                SEND A-RELEASE-RQ
                GOTO END
        }
        GOTO END.
}
IF  A-ASSOCAITE-RQ RJ
        GOTO END.
END:
CLOSE Socket   /* End of  TCP/IP Service */
```

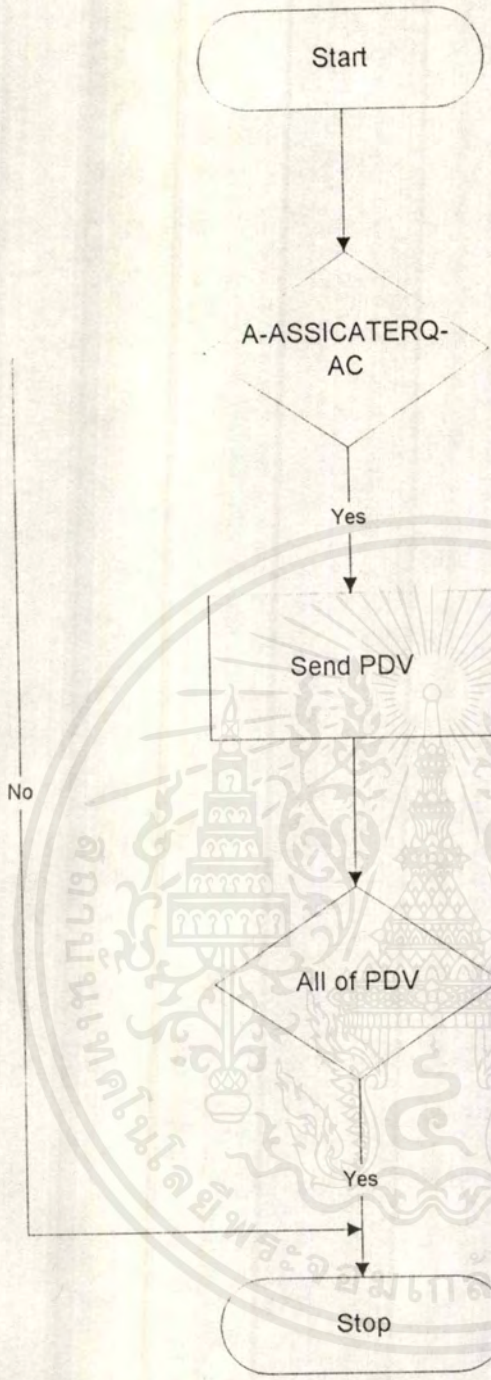


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 การขอการเชื่อมต่อ

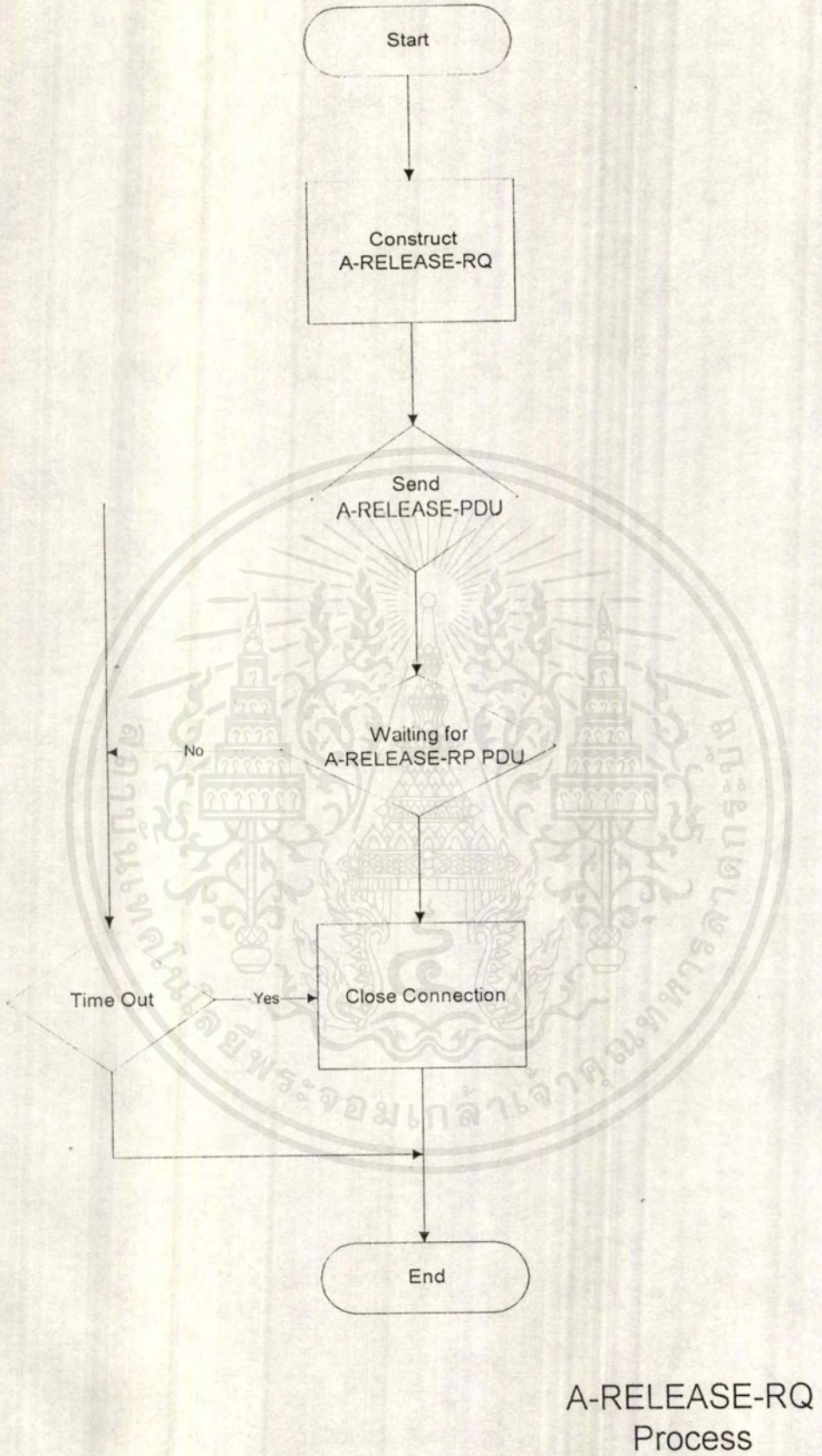
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Send Protocol Data Value

รูปที่ 3.6 การส่งข้อมูลผ่าน PDV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 การขอยกเลิกการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 การทดสอบการสื่อสารข้อมูล

ความถูกต้องของข้อมูลที่ทำการสื่อสารทดสอบโดยใช้หลักการเดียวกันกับการตรวจสอบข้อมูลภาพคือทำการอ่านข้อมูลของจุดภาพของภาพต้นทางต้นทางเก็บเป็นแฟ้มไว้ แล้วเปรียบเทียบกับข้อมูลที่ส่งผ่านไปบนระบบเครือข่าย ในที่นี้จะทดสอบแต่เพียงส่วนที่เรียกว่า DICOM Upper Layer ซึ่งตรงกับชั้นโปรโตคอลที่ 5, 6, 7 ตามมาตรฐาน OSI การทดสอบจึงหมายถึงการทดสอบความถูกต้องของชุดคำสั่ง (อ่านได้จากการส่งค่ากลับมาของ DICOM Server) และการทดสอบข้อมูลจุดภาพ ซึ่งก็ใช้หลักการเดียวกันกับการทดสอบข้อมูลจุดภาพที่ทดสอบในตอนที่แล้ว ขั้นตอนการตรวจสอบมีดังนี้คือ

1) ทดสอบชุดคำสั่งโดยการส่งคำสั่งที่ต้องการออกไปยัง DICOM Server
2) อ่านค่าที่ Server ส่งกลับออกมา
3) เปรียบเทียบค่าที่อ่านได้กับมาตราฐานที่กำหนดไว้ใน Part 7 และ Part 8
4) ทดสอบความถูกต้องของข้อมูลจุดภาพโดยการส่งข้อมูลไปยัง DICOM Server
5) นำข้อมูลจาก DICOM Server กลับมายังเครื่องคอมพิวเตอร์ที่ทำการส่ง
6) เก็บข้อมูลจุดภาพต้นฉบับลงแฟ้ม อ่านค่าจุดภาพของข้อมูลที่ส่งกลับมาลงแฟ้ม
7) เปรียบเทียบข้อมูลระหว่างแฟ้มทั้งสอง

หมายเหตุ การทดสอบการส่งนี้ทำบนสมมติฐานที่ว่าภายใน Layer ที่ต่ำกว่าทำงานถูกต้องกล่าวคือในชั้นกายภาพและในส่วนของการเข้าถึงสื่อกลางมีกระบวนการตรวจสอบข้อมูลที่ถูกต้อง รวมไปถึง Winsock API หรือส่วนที่จัดการการเข้าถึง TCP/IP ทำงานได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

4.1 บทนำ

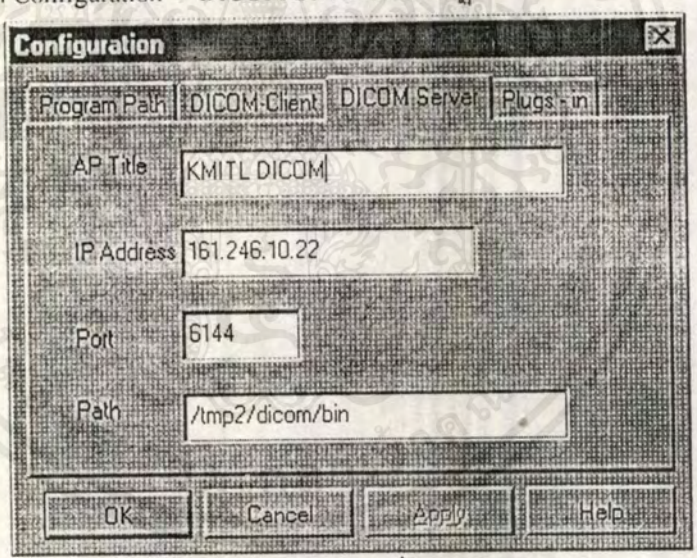
ในบทนี้จะกล่าวถึงการนำโปรแกรมที่เขียนขึ้นในบทที่แล้วไปใช้งานเพื่อนำภาพจากเครื่องมือแพทย์ที่ไม่ได้ใช้มาตรฐาน DICOM นำมาแปลงและส่งความสู่ระบบการจัดเก็บและจัดการภาพรังสีการแพทย์ กล่าวคือ ทดสอบหุนสมบัติของภาพที่แปลงเสร็จสิ้นแล้วโดยการนำไปแสดงด้วยโปรแกรม Osiris ซึ่งเป็นโปรแกรมมาตรฐานที่ใช้แสดงภาพมาตรฐาน DICOM และทำการส่งข้อมูลที่แปลงแล้วไปยัง RSNA DICOM Server เพื่อทดสอบโปรโตคอลส่วนบนแบบ DICOM

4.2 การทดลองแปลงข้อมูล

4.2.1 การทดลองแปลงข้อมูลที่มีส่วนขยายแบบ BMP

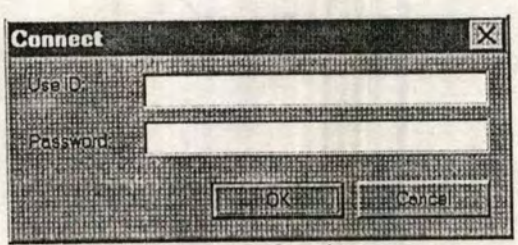
มีวิธีการทดลองดังนี้

1) เริ่มโปรแกรม จะปรากฏกรอบข้อความเพื่อเชื่อมต่อเข้าระบบเครือข่าย หากต้องการเปลี่ยน IP ให้กดยกเลิกแล้วกด Configuration- > DICOM Server จะปรากฏ



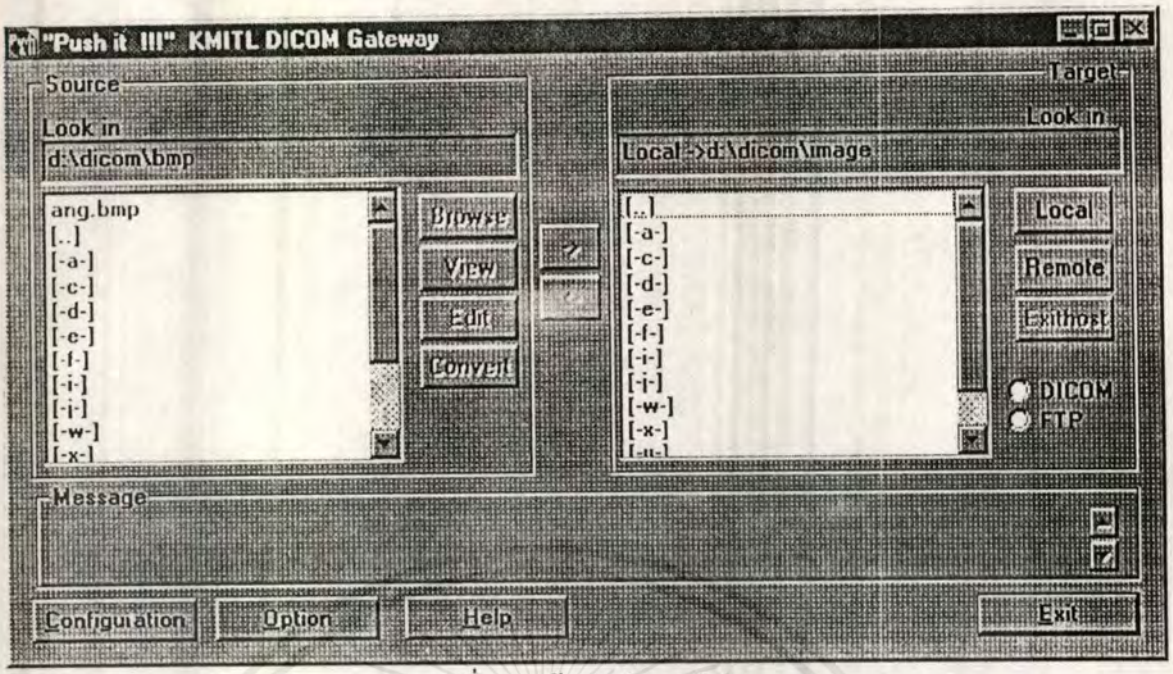
รูปที่ 4.1 กรอบข้อความเพื่อกำหนดเลข IP

2) ในที่นี้ยังไม่ต้องการส่งข้อมูลจึงเลือกปุ่มยกเลิก (Cancel) เพื่อปฏิเสธการเชื่อมต่อไปยัง DICOM Server



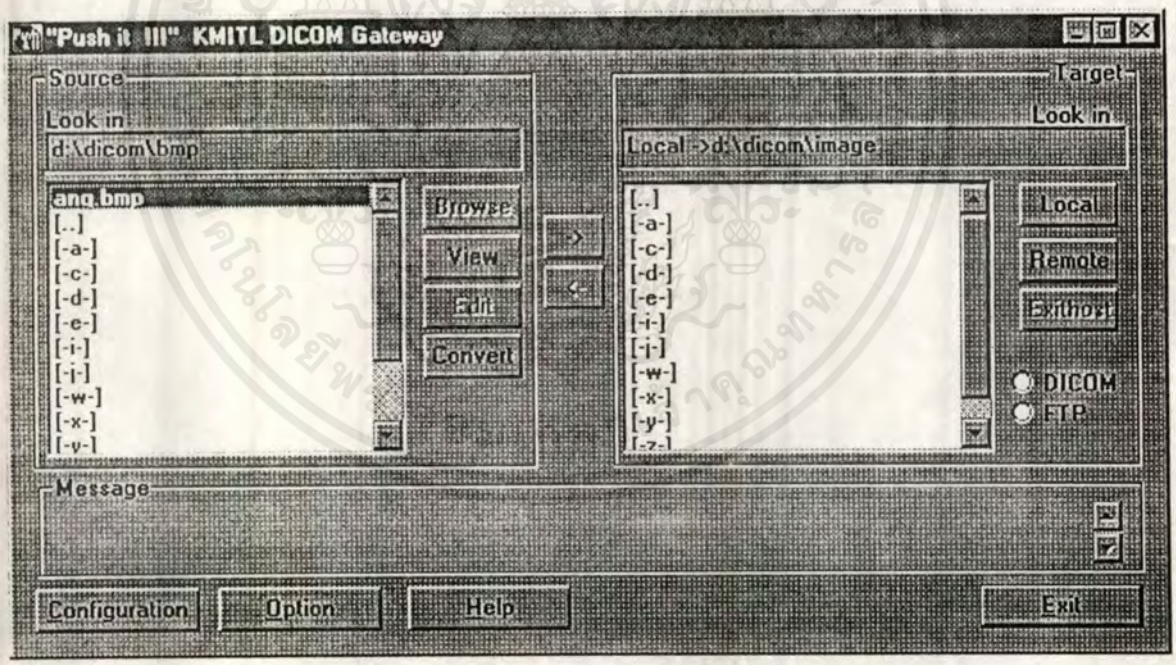
รูปที่ 4.2 กรอบข้อความเพื่อเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 หน้าจอหลักของโปรแกรม

4) ทดสอบการแปลงข้อมูลแบบ BMP เลือกแฟ้มต้นฉบับที่มีส่วนขยายเป็น .BMP

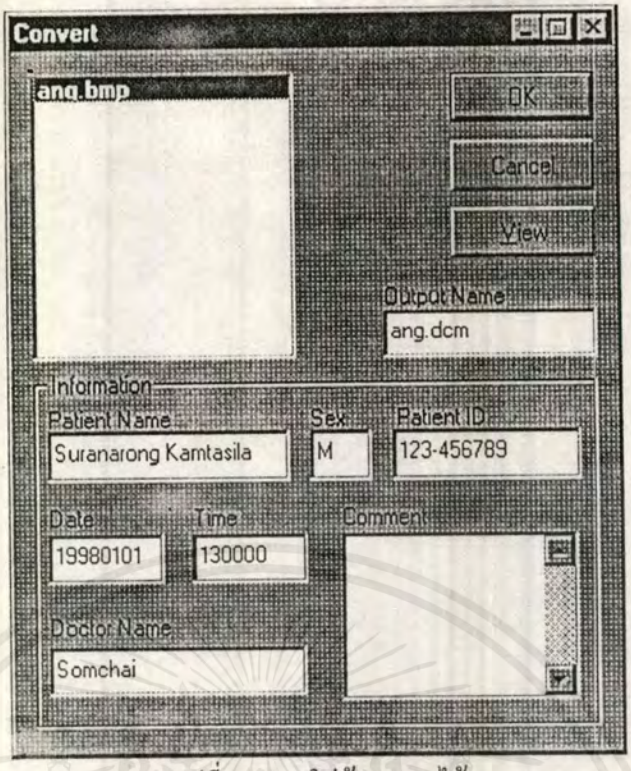


รูปที่ 4.4 เลือกแฟ้มต้นทาง

5) ทำเลือกที่ปุ่มแปลง (Convert)

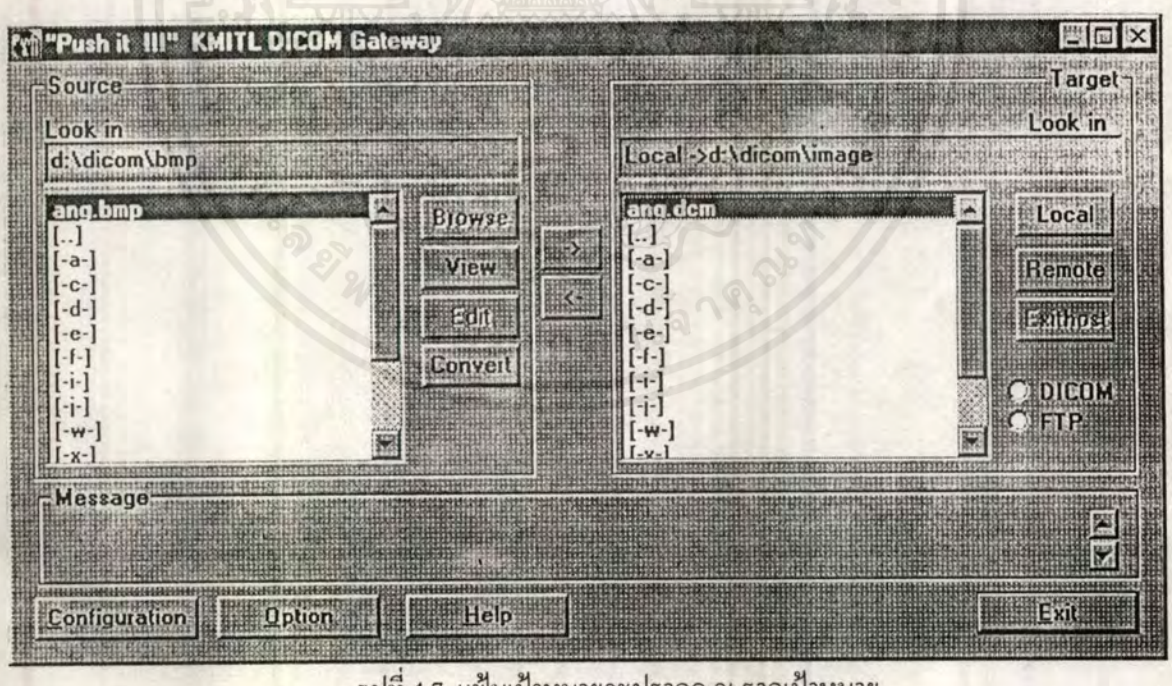
6) จะปรากฏกรอบข้อความเพื่อให้ใส่ข้อมูลคนไข้ลงไป กรอกข้อความจนครบเลือกปุ่มตกลง (Ok)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 การใส่ข้อมูลคนไข้

7) โปรแกรมจะทำการแปลงข้อมูลแล้วเก็บยังเป้าหมายที่ได้เลือกไว้ในข้อ 3)

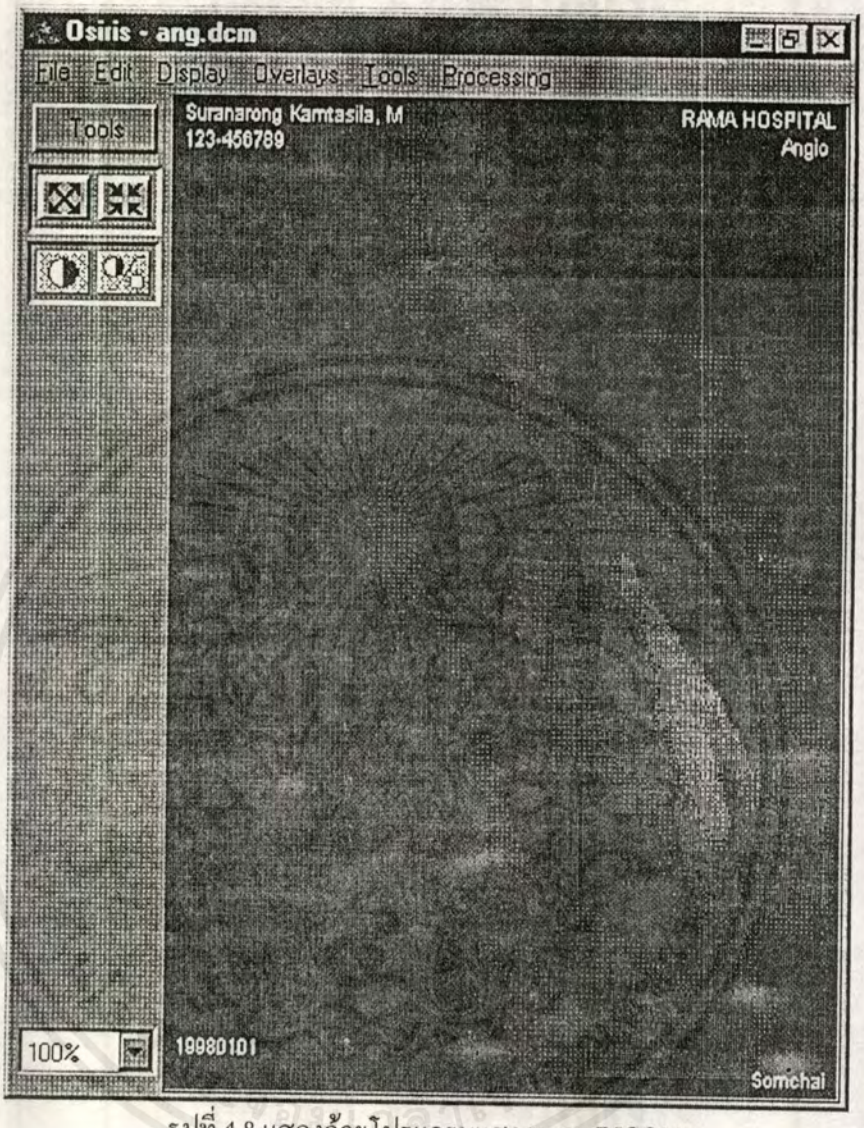


รูปที่ 4.7 แฟ้มเป้าหมายจะปรากฏ ณ รากเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

ทดสอบความถูกต้องโดยการแสดงบนโปรแกรม Osiris

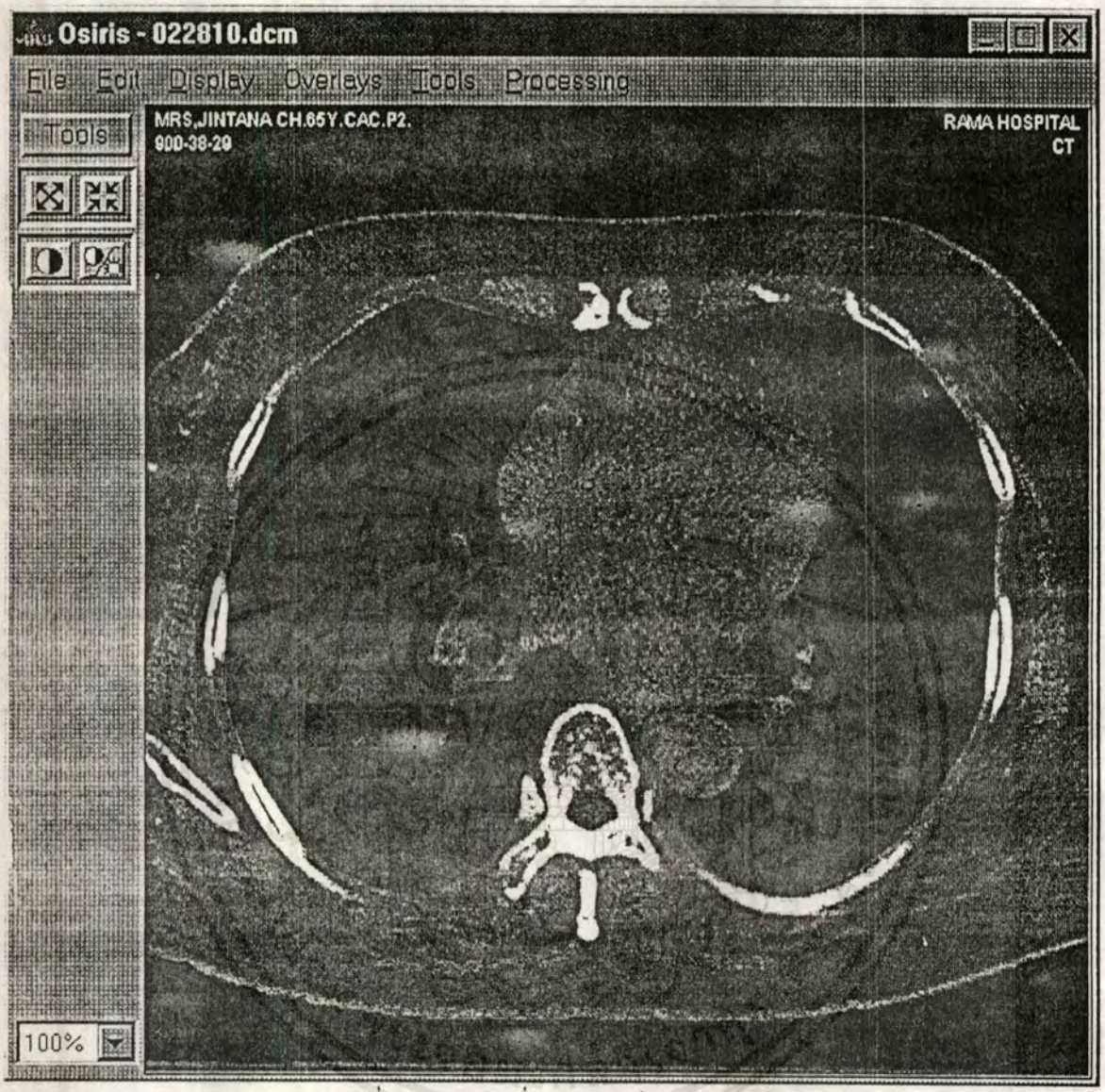


รูปที่ 4.8 แสดงด้วยโปรแกรมแสดงภาพ DICOM

4.2.2 การแปลงภาพแบบอื่น ๆ

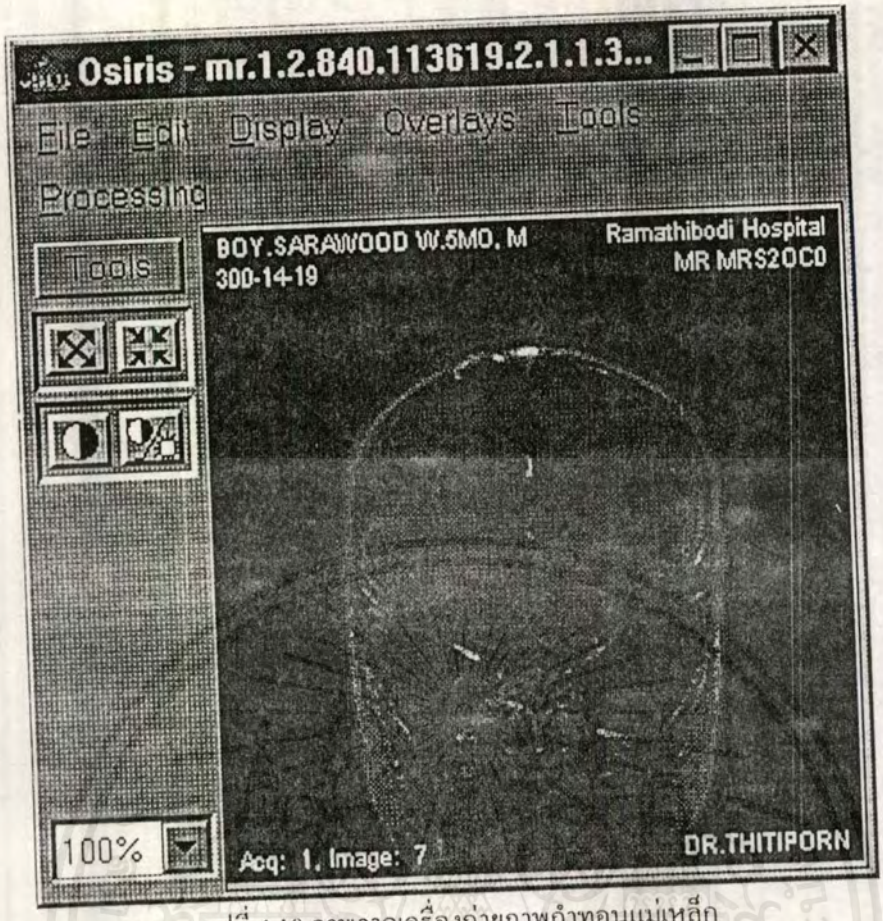
ด้วยวิธีเดียวกันสามารถทำการทดลองกับภาพอื่นๆ ได้ โดยผลที่ได้มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผลจากเครื่องเอกซเรย์คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

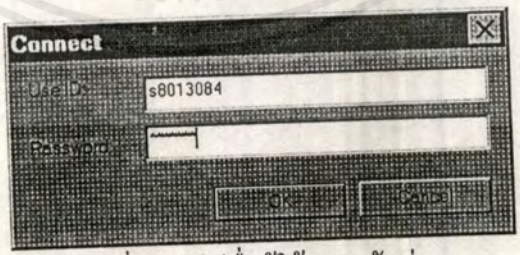


รูปที่ 4.10 ภาพจากเครื่องถ่ายภาพกำทอนแม่เหล็ก

4.3 การทดสอบโปรโตคอลส่วนบนแบบ DICOM

มีวิธีการทดลองดังนี้

1) ทำการเริ่มโปรแกรม RSNA DICOM Server
2) เริ่มโปรแกรม
3) กรอกรายชื่อผู้ใช้และรหัสผ่านลงในช่องที่กำหนด



รูปที่ 4.11 ใส่ชื่อผู้ใช้และรหัสผ่าน

4) ที่กรอบด้านแหล่งภาพ (Source) เลือกแฟ้มข้อมูลที่มีส่วนขยายเป็น DCM
5) เลือกปุ่มส่ง (->)
6) โปรแกรมจะทำการส่งข้อมูลไปยัง DICOM Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

ข้อความด้านล่างเป็นข้อความที่อุปกรณ์ที่ให้บริการการเก็บภาพสื่อสารกับโปรแกรมที่เขียนขึ้น ผลที่ได้แสดงถึงขั้นตอนแต่ละขั้นในการส่งข้อมูล ข้อความดังกล่าวเป็นขั้นการเริ่มต้นการขอส่งข้อมูล

```
DUL_InitializeNetwork, Type: NETWORK TCP, Mode: AE ACCEPTOR

***BEFORE LISTEN***
***AFTER LISTEN***

DUL_Receive Association RQ

*************BEFORE ACCEPT*************
*************AFTER ACCEPT(sock: 4/errno: 0)*************

DUL FSM Table: State: 1 Event: 4
DUL Event: Transport connection indication
DUL Action: AE 5 Transport Connect Response
Read PDU HEAD TCP: 01 00 00 00 00 cd
Read PDU HEAD TCP: type: 1, length: 205 (cd)
DUL FSM Table: State: 2 Event: 5
DUL Event: A-ASSOCIATE-RQ PDU (on tranport)
DUL Action: AE 6 Examine Associate Request
PDU Type: Associate Request PDU Length: 211
 01 00 00 00 00 cd 00 01 00 00 41 4c 59 2d 53 43
 50 20 20 20 20 20 20 20 20 20 4b 4d 49 54 4c 5f
 44 49 43 4f 4d 5f 76 31 30 20 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 10 00 00 15 31 2c
 32 2c 38 34 30 2c 31 30 30 30 38 2e 33 2e 31 2e
 31 2c 31 20 00 00 2c 01 00 ff 00 30 00 00 11 31
 2c 32 2c 38 34 30 2c 31 30 30 30 38 2c 31 2c 31
 40 00 00 11 31 2c 32 2c 38 34 30 2c 31 30 30 30
 38 2c 31 2e 32 50 00 00 3a 51 00 00 04 00 00 40
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00 52 00 00 1b 31 2e 32 2e 32 37 36 2e 30 2e 37

32 33 30 30 31 30 2e 33 2e 30 2e 33 2e 32 2e 31

55 00 00 0f 4b 4d 49 54 4c 5f 44 49 43 4f 5d 5f

56 31 30

Parsing an A-ASSOCIATE PDU

PDU type: 1 (A-ASSOCIATE RQ), PDU Length: 205

DICOM Protocol: 1

Called AP Title: ANY-SCP

Calling AP Title: KMITL_DICOM_v10

Parsing remaining 137 bytes of A-ASSOCIATE PDU

Next item type: 10

Subitem parse: Type 10, Length 21, Content: 1.2.840.10008.3.1.1.1

Successfully parsed Application Context

Parsing remaining 112 bytes of A-ASSOCIATE PDU

Next item type: 20

Parsing Presentation Context: (20), Length: 46

Presentation Context ID: 1

Parsing remaining 42 bytes of Presentation Ctx

Next item type: 30

Subitem parse: Type 30, Length 17, Content: 1.2.840.10008.1.1

Successfully parsed Abstract Syntax

Parsing remaining 21 bytes of Presentation Ctx

Next item type: 40

Subitem parse: Type 40, Length 17, Content: 1.2.840.10008.1.2

Successfully parsed Transfer Syntax

Successfully parsed Presentation Context

Parsing remaining 62 bytes of A-ASSOCIATE PDU

Next item type: 50

Parsing user info field (50), Length: 58

Parsing remaining 58 bytes of User Information

Next item type: 51

Maximum PDU Length: 16384

Successfully parsed Maximum PDU Length

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Parsing remaining 50 bytes of User Information

Next item type: 52

Subitem parse: Type 52, Length 27, Content: 1.2.276.0.7230010.3.0.3.2.1

Parsing remaining 19 bytes of User Information

Next item type: 55

Subitem parse: Type 55, Length 15, Content: KMITL_DICOM_V10

Successfully parsed User Information

Association Received

Parameters:

Our Implementation Class UID: 1.2.276.0.7230010.3.0.3.2.1

Our Implementation Version Name: KMITL_DICOM_V10

Their Implementation Class UID: 1.2.276.0.7230010.3.0.3.2.1

Their Implementation Version Name: KMITL_DICOM_V10

Application Context Name: 1.2.840.10008.3.1.1.1

Calling Application Name: KMITL_DICOM_v10

Called Application Name: ANY-SCP

Responding Application Name:

Our Max PDU Receive Size: 16384

Their Max PDU Receive Size: 16384

Presentation Contexts:

Context ID: 1 (Proposed)

Abstract Syntax: =VerificationSOPClass

Proposed SCP/SCU Role: Default

Accepted SCP/SCU Role: Default

Proposed Transfer Syntax(es):

=LittleEndianImplicit

DUL_Acknowledge Association RQ

DUL FSM Table: State: 3 Event: 6

DUL Event: A-ASSOCIATE resp prim (accept)

DUL Action: AE 7 Send Associate AC

Constructing Associate AC PDU

Association Acknowledged (Max Send PDV: 16372)

Our Implementation Class UID: 1.2.276.0.7230010.3.0.3.2.1

Our Implementation Version Name: KMITL_DICOM_V10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Their Implementation Class UID:    1.2.276.0.7230010.3.0.3.2.1
Their Implementation Version Name: KMITL_DICOM_V10
Application Context Name:    1.2.840.10008.3.1.1.1
Calling Application Name:    KMITL_DICOM_v10
Called Application Name:     ANY-SCP
Responding Application Name: STORESCP
Our Max PDU Receive Size: 16384
Their Max PDU Receive Size: 16384
Presentation Contexts:
 Context ID:       1 (Accepted)
  Abstract Syntax: =VerificationSOPClass
  Proposed SCP/SCU Role: Default
  Accepted SCP/SCU Role: Default
  Accepted Transfer Syntax: =LittleEndianImplicit
DIMSE receiveCommand
Read PDU HEAD TCP:  04 00 00 00 00 4a
Read PDU HEAD TCP: type: 4, length: 74 (4a)
DUL  FSM Table: State:  6 Event:  9
DUL  Event:  P-DATA-TF PDU (on transport)
DUL  Action: DT 2 Indicate P DATA PDU Received
BEGIN-DcmDataset::read(xfer=0,glenc=1)
 BEGIN-DcmItem::read(xfer=0)
   TagAndLength read of: (0x0000,0x0000) "UL" [0x00000004] "CommandGroupLength"
   BEGIN-dcitem:DcmItem::insert(DcmElement*=d96f0,BOOL replaceOld=0)
    element (0x0000,0x0000) / VR="UL" at beginning inserted
   END---dcitem:DcmIt...
   TagAndLength read of: (0x0000,0x0002) "UI" [0x00000012] "AffectedSOPClassUID"
   BEGIN-dcitem:DcmItem::insert(DcmElement*=d9720,BOOL replaceOld=0)
    element (0x0000,0x0002) / VR="UI" inserted
   END---dcitem:DcmIt...
   TagAndLength read of: (0x0000,0x0100) "US" [0x00000002] "CommandField"
   BEGIN-dcitem:DcmItem::insert(DcmElement*=d9760,BOOL replaceOld=0)
    element (0x0000,0x0100) / VR="US" inserted
   END---dcitem:DcmIt...
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TagAndLength read of: (0x0000,0x0110) "US" [0x00000002] "MessageID"
BEGIN-dcitem:DcmItem::insert(DcmElement*=d9790,BOOL replaceOld=0)
 element (0x0000,0x0110) / VR="US" inserted
END---dcitem:DcmIt...
TagAndLength read of: (0x0000,0x0800) "US" [0x00000002] "DataSetType"
BEGIN-dcitem:DcmItem::insert(DcmElement*=d97c0,BOOL replaceOld=0)
 element (0x0000,0x0800) / VR="US" inserted
END---dcitem:DcmIt...
errorFlag=(0)
END---DcmItem::rea...
errorFlag=(0)
END---DcmDataset::...
DIMSE receiveCommand: 1 pdv's (68 bytes), presID=1
DIMSE Command Received:
# Dcom Data Set
# Used TransferSyntax: LittleEndianImplicit
(0000,0002) UI  =VerificationSOPClass          #  18, 1 AffectedSOPClassUID
(0000,0100) US  48                             #   2, 1 CommandField
(0000,0110) US  1                              #   2, 1 MessageID
(0000,0800) US  257                            #   2, 1 DataSetType
BEGIN-dcitem:DcmItem::remove(tag=(0000,0100))
 element p=d9760 removed, but not deleted
END---dcitem:DcmIt...
BEGIN-dcitem:DcmItem::remove(tag=(0000,0110))
 element p=d9790 removed, but not deleted
END---dcitem:DcmIt...
BEGIN-dcitem:DcmItem::remove(tag=(0000,0800))
 element p=d97c0 removed, but not deleted
END---dcitem:DcmIt...
BEGIN-dcitem:DcmItem::remove(tag=(0000,0002))
 element p=d9720 removed, but not deleted
END---dcitem:DcmIt...
Received C-Echo RQ: MsgID 1
 AffectedSOPClassUID: =VerificationSOPClass
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Data Set: Not Present
BEGIN-DcmUnsignedShort::put(slong=32816,num=0)
END---DcmUnsignedS...
BEGIN-dcitem:DcmItem::insert(DcmElement*=d97b8,BOOL replaceOld=1)
 element (0x0000,0x0100) / VR="US" at beginning inserted
END---dcitem:DcmIt...
BEGIN-DcmUnsignedShort::put(slong=1,num=0)
END---DcmUnsignedS...
BEGIN-dcitem:DcmItem::insert(DcmElement*=d97c8,BOOL replaceOld=1)
 element (0x0000,0x0120) / VR="US" inserted
END---dcitem:DcmIt...
BEGIN-DcmUnsignedShort::put(slong=257,num=0)
END---DcmUnsignedS...
BEGIN-dcitem:DcmItem::insert(DcmElement*=d9818,BOOL replaceOld=1)
 element (0x0000,0x0800) / VR="US" inserted
END---dcitem:DcmIt...
BEGIN-DcmUnsignedShort::put(slong=0,num=0)
END---DcmUnsignedS...
BEGIN-dcitem:DcmItem::insert(DcmElement*=d9848,BOOL replaceOld=1)
 element (0x0000,0x0900) / VR="US" inserted
END---dcitem:DcmIt...
BEGIN-dcitem:DcmItem::insert(DcmElement*=d9878,BOOL replaceOld=1)
 element (0x0000,0x0002) / VR="UI" at beginning inserted
END---dcitem:DcmIt...
DIMSE Command To Send:

# Dicom-Data-Set
# Used TransferSyntax: UnknownTransferSyntax
(0000,0002) UI  =VerificationSOPClass          #  18, 1 AffectedSOPClassUID
(0000,0100) US  32816                          #   2, 1 CommandField
(0000,0120) US  1                              #   2, 1 MessageIDBeingRespondedTo
(0000,0800) US  257                            #   2, 1 DataSetType
(0000,0900) US  0                              #   2, 1 Status
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจสอบการทำงานของโปรโตคอลส่วนบนแบบ DICOM

จากข้อความที่ผ่านมาเมื่อพิจารณาค่าที่ได้จาก DICOM Server เมื่อตีความส่วนที่เป็นชุดคำสั่งเปรียบเทียบกับมาตรฐานพบว่าตรงกับมาตราฐานที่กำหนดไว้ (Part 8)

สำหรับการทดสอบส่วนของข้อมูลจุดภาพโดยอาศัยหลักการส่งไปยัง DICOM Server แล้วนำข้อมูลส่งกลับมาจากนั้นทำการเปรียบเทียบข้อมูลจุดภาพแล้วผลที่ได้คือ

ชื่อแฟ้ม	ขนาดแฟ้ม (byte)	ขนาดข้อมูลจุดภาพก่อนส่ง (byte)	ส่งครั้งที่	ขนาดของแฟ้มที่ DICOM Server (byte)	ขนาดของจุดภาพ (byte)	ผลต่าง
3dheart.dcm	254720	254016	1 2 3	254764 254764 254764	254016 254016 254016	0 0 0
mr.dcm	132176	131072	1 2 3	132504 132504 132504	131072 131072 131072	0 0 0
us001.dcm	310008	307200	1 2 3	310124 310124 310124	307200 307200 307200	0 0 0

ตารางที่ 4.1 การเปรียบเทียบผลของจุดภาพก่อนและหลังการส่ง

จากผลทำให้เราทราบว่าผลของจุดภาพไม่มีการเปลี่ยนแปลงหลังจากผ่านการส่งข้อมูลแล้ว ผลที่แตกต่างกันของข้อมูลระหว่างการส่งข้อมูลแต่ละครั้งคือ ความเร็วของการส่งแต่ละครั้งมีค่าไม่เท่ากัน ซึ่งเกิดจากสภาวะแวดล้อมภายในระบบเครือข่ายเอง แต่ในโครงงานนี้สนใจถึงความถูกต้องของข้อมูลภายใต้สมมติฐานว่าโปรโตคอลชั้นที่ต่ำกว่าทำงานได้ถูกต้องตามกลไกของมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและแนวทางการพัฒนา

5.1 สรุปผลการทดลอง

จากการทดลองในบทที่ผ่านมาพบว่าในเบื้องต้นสามารถนำข้อมูลจากเครื่องมือแพทย์ที่ไม่ได้สนับสนุนมาตรฐาน DICOM ให้สามารถใช้งานร่วมกับเครื่องมือที่ใช้มาตราฐาน DICOM ได้ โดยแบ่งการพิจารณาเป็น 2 ส่วนคือการใช้ข้อมูลและการใช้โปรโตคอลสื่อสาร

ในส่วนของการใช้ข้อมูลภาพและข้อมูลคนไข้ที่ผ่านการแปลงสามารถแสดงได้ถูกต้องบนโปรแกรมแสดงภาพมาตราฐาน DICOM

ในส่วนของการสื่อสารข้อมูล สามารถส่งข้อมูลจากเครื่องมือที่ให้ภาพที่มีอยู่ทั่วไปเข้าไปในระบบการจัดเก็บแบบ DICOM ได้

5.2 แนวทางการพัฒนาขั้นต่อไป

ในการใช้งานถ้าจะเพิ่มประสิทธิภาพในการใช้งานหากสามารถลดขนาดของภาพต้นฉบับลงโดยที่ข้อมูลภาพยังอยู่ครบเมื่อทำการแสดงภาพก็จะทำให้สามารถส่งข้อมูลได้ความเร็วต่อภาพมากขึ้น อันส่งผลให้ประสิทธิภาพในการสื่อสารข้อมูลสูงขึ้นนั่นเอง ดังนั้นการบีบบอัดข้อมูลภาพโดยคำนึงถึงคุณภาพของภาพเป็นหลักจึงมีความจำเป็นและเป็นแนวทางในการพัฒนาต่อไป รวมไปถึงการพัฒนาระบบการจัดเก็บหรือฐานข้อมูลให้มีความยืดหยุ่นและสะดวกในการใช้งาน

หมายเหตุ สำหรับแฟ้มต้นฉบับในโครงงานนี้ทั้งหมดสำหรับ Visual C++ 5.0 สามารถดาวโหลดได้ที่ ftp://161.246.11.197 หรือ http://161.246.10.21/~s8013084

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
////////////////////////////////////////////////////
//
// GatewayDlg.h :header file
//              :DICOM Gateway Main Screen
//

#if !defined(AFX_GATEWAYDLG_H__B32644C8_8E4D_11D0_8B02_0060083A12C4__IN
CLUDED_)
#define AFX_GATEWAYDLG_H__B32644C8_8E4D_11D0_8B02_0060083A12C4__INCLUDE
D_

#if _MSC_VER >= 1000
#pragma once
#endif // _MSC_VER >= 1000


#include "option.h"
#include "config.h"
#include "programpath.h"
#include "configdicom.h"
#include "configdicomserver.h"
#include "configplugin.h"

#include "converter.h"
#include "connect.h"


#include "localfile.h"



////////////////////////////////////////////////////////////////////////
//////
// CGatewayDlg dialog
//
class CGatewayDlg : public CDialog
{
// Construction
public:
    CString m_Configuration_Path_Bin;
    CString m_Configuration_Path_Plugin;
    CString m_Configuration_Path_Log;
    CString m_Configuration_Path_Temp;
    UINT    m_Configuration_DICOMServer_Port;
    CString m_Configuration_DICOMServer_APtitle;
    CString m_Configuration_DICOMClient_APtitle;
    CString m_Configuration_DICOMClient_StartUpDir;
    CString m_Configuration_DICOMServer_IP;
    CString m_Configuration_DICOMServer_Path;
    BOOL m_Option_AutoConvert;
    BOOL m_Option_ShowTargetPath;
    BOOL m_Option_ConnectDicomServer;
    CGatewayDlg(CWnd* pParent = NULL);  //standard constructor


private:
    void AddItem(CString sItem);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int count;
    int nGetDriveNumber(CString driveletter);

    BOOL m_AreWeConnect;
    void AddLog(CString Log);
    void AddLogMessage(CString message);
    void SaveINI();
    void ReadINI();

    CString m_StartLocaldir;

    void UpdateLocalDir();
    void UpdateTargetLocalDir();
    void StartUpControl();
    int  GetLocalDirForWnd(void);


// Dialog Data
    //{{AFX_DATA(CGatewayDlg)
    enum { IDD = IDD_GATEWAY_DIALOG };
    CEdit   m_TargetLookin;
    CEdit   m_SourceLookin;
    CListBox    m_localfile;
    CListBox    m_targetfile;
    CButton     m_TargetRemote;
    CButton     m_TargetLocal;
    CButton     m_SourceView;
    CButton     m_SourceEdit;
    CButton     m_SourceConvert;
    CButton     m_SourceBrowse;
    CButton     m_Put;
    CButton     m_Option;
    CButton     m_Help;
    CButton     m_Get;
    CButton     m_Configuration;
    CString     m_message;
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CGatewayDlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV supp
ort
    //}}AFX_VIRTUAL

// Implementation
protected:
    HICON m_hIcon;

    // Generated message map functions
    //{{AFX_MSG(CGatewayDlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    afx_msg void OnButtonOption();
    afx_msg void OnButtonConfiguration();
    afx_msg void OnButtonSourceConvert();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        afx_msg void OnButtonPut();
        afx_msg void OnButtonTargetRemote();
        afx_msg void OnButtonTargetLocal();
        afx_msg void OnDblclkListSourcefile();
        afx_msg void OnSelchangeListSourcefile();
        afx_msg void OnDblclkListTargetfile();
        afx_msg void OnSelchangeListTargetfile();
        afx_msg void OnButtonGet();
        afx_msg void OnButtonSourceView();
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Developer Studio will insert additional declarations immed
iately before the previous line.

#endif // !defined(AFX_GATEWAYDLG_H__B32644C8_8E4D_11D0_8B02_0060083A12
C4__INCLUDED_)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
////////////////////////////////////////////////////////////
//
//
// FILE :GatewayDlg.cpp : implementation file
//      :DICOM Gateway Main Screen
//
//

#include "stdafx.h"
#include "Gateway.h"
#include "GatewayDlg.h"

#include <io.h>
#include <direct.h>
#include <afxinet.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////////////////////////////////////////////
//////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
    //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CAboutDlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV supp
ort
    //}}AFX_VIRTUAL

// Implementation
protected:
    //{{AFX_MSG(CAboutDlg)
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{AFX_DATA_INIT(CAboutDlg)
    //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
        // No message handlers
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
//////
// CGatewayDlg dialog

CGatewayDlg::CGatewayDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CGatewayDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CGatewayDlg)
    m_message = _T("");
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in
Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    count = 0;
}

void CGatewayDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CGatewayDlg)
    DDX_Control(pDX, IDC_EDIT_TARGET_LOOKIN, m_TargetLookin);
    DDX_Control(pDX, IDC_EDIT_SOURCE_LOOKIN, m_SourceLookin);
    DDX_Control(pDX, IDC_LIST_SOURCEFILE, m_localfile);
    DDX_Control(pDX, IDC_LIST_TARGETFILE, m_targetfile);
    DDX_Control(pDX, IDC_BUTTON_TARGET_REMOTE, m_TargetRemote);
    DDX_Control(pDX, IDC_BUTTON_TARGET_LOCAL, m_TargetLocal);
    DDX_Control(pDX, IDC_BUTTON_SOURCE_VIEW, m_SourceView);
    DDX_Control(pDX, IDC_BUTTON_SOURCE_EDIT, m_SourceEdit);
    DDX_Control(pDX, IDC_BUTTON_SOURCE_CONVERT, m_SourceConvert);
    DDX_Control(pDX, IDC_BUTTON_SOURCE_BROWSE, m_SourceBrowse);
    DDX_Control(pDX, IDC_BUTTON_PUT, m_Put);
    DDX_Control(pDX, IDC_BUTTON_OPTION, m_Option);
    DDX_Control(pDX, IDC_BUTTON_HELP, m_Help);
    DDX_Control(pDX, IDC_BUTTON_GET, m_Get);
    DDX_Control(pDX, IDC_BUTTON_CONFIGURATION, m_Configuration);
    DDX_Text(pDX, IDC_EDIT_MESSAGE, m_message);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CGatewayDlg, CDialog)
    //{{AFX_MSG_MAP(CGatewayDlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON_OPTION, OnButtonOption)
    ON_BN_CLICKED(IDC_BUTTON_CONFIGURATION, OnButtonConfiguration)
    ON_BN_CLICKED(IDC_BUTTON_SOURCE_CONVERT, OnButtonSourceConvert)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ON_BN_CLICKED(IDC_BUTTON_PUT, OnButtonPut)
    ON_BN_CLICKED(IDC_BUTTON_TARGET_REMOTE, OnButtonTargetRemote)
    ON_BN_CLICKED(IDC_BUTTON_TARGET_LOCAL, OnButtonTargetLocal)
    ON_LBN_DBLCLK(IDC_LIST_SOURCEFILE, OnDblclkListSourcefile)
    ON_LBN_SELCHANGE(IDC_LIST_SOURCEFILE, OnSelchangeListSourcefile)
    ON_LBN_DBLCLK(IDC_LIST_TARGETFILE, OnDblclkListTargetfile)
    ON_LBN_SELCHANGE(IDC_LIST_TARGETFILE, OnSelchangeListTargetfile)
    ON_BN_CLICKED(IDC_BUTTON_GET, OnButtonGet)
    ON_BN_CLICKED(IDC_BUTTON_SOURCE_VIEW, OnButtonSourceView)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()


////////////////////////////////////////////////////////////////////////////
//////
// CGatewayDlg message handlers
BOOL CGatewayDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu)
;
        }
    }

    // Set the icon for this dialog.  The framework does this automatic
ally
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);         // Set big icon
    SetIcon(m_hIcon, FALSE);        // Set small icon


    ReadINI();
    StartUpControl();
    UpdateLocalDir();



    if(m_Option_ConnectDicomServer == TRUE)
    {
        CConnect dlg;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(dlg.DoModal() == IDOK)
            {
                m_AreWeConnect = TRUE;

            }
            m_AreWeConnect = FALSE;
            UpdateTargetLocalDir();
        }

        return TRUE;  //return TRUE unless you set the focus to a control
    }


    void CGatewayDlg::OnSysCommand(UINT nID, LPARAM lParam)
    {
        if ((nID & 0xFFF0) == IDM_ABOUTBOX)
        {
            CAboutDlg dlgAbout;
            dlgAbout.DoModal();
        }
        else
        {
            CDialog::OnSysCommand(nID, lParam);
        }
    }


    void CGatewayDlg::OnPaint()
    {
        if (IsIconic())
        {
            CPaintDC dc(this); // device context for painting

            SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

            // Center icon in client rectangle
            int cxIcon = GetSystemMetrics(SM_CXICON);
            int cyIcon = GetSystemMetrics(SM_CYICON);
            CRect rect;
            GetClientRect(&rect);
            int x = (rect.Width() - cxIcon + 1) / 2;
            int y = (rect.Height() - cyIcon + 1) / 2;

            // Draw the icon
            dc.DrawIcon(x, y, m_hIcon);
        }
        else
        {
            CDialog::OnPaint();
        }
    }


    //The system calls this to obtain the cursor to display while the user
    drags
    //the minimized window.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HCURSOR CGatewayDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

//////////////////////////////////////////////////////
//  SETTING AREA
//
//////////////////////////////////////////////////////


//////////////////////////////////////////////////////
//Press Option Button then call Option Dialog
//
void CGatewayDlg::OnButtonOption()
{
    COption dlg;

    dlg.Setting(m_Option_ConnectDicomServer,
                m_Option_ShowTargetPath,
                m_Option_AutoConvert);

    if(dlg.DoModal() == IDOK)
    {
        m_Option_ConnectDicomServer = dlg.m_connectdicomserver;
        m_Option_ShowTargetPath     = dlg.m_showpath;
        m_Option_AutoConvert        = dlg.m_autoconvert;
        SaveINI();
    }

}


//////////////////////////////////////////////////////////////////
//Event : Press Configuration Button then call Configuration Dialog
//
//
//
void CGatewayDlg::OnButtonConfiguration()
{
    CConfig               sheet("Configuration");
    CProgramPath          page1;
    CConfigDicom          page2;
    CConfigDicomServer    page3;
    CConfigPlugin         page4;

    sheet.AddPage(&page1);
    sheet.AddPage(&page2);
    sheet.AddPage(&page3);
    sheet.AddPage(&page4);

    page1.m_bin     = m_Configuration_Path_Bin;
    page1.m_plugin  = m_Configuration_Path_Plugin;
    page1.m_log     = m_Configuration_Path_Log;
    page1.m_temp    = m_Configuration_Path_Temp;

    page2.m_aptitle     = m_Configuration_DICOMClient_APtitle;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        page2.m_startupdir = m_Configuration_DICOMClient_StartUpDir;

        page3.m_aptitle     = m_Configuration_DICOMServer_APtitle;
        page3.m_port        = m_Configuration_DICOMServer_Port;
        page3.m_serverpath  = m_Configuration_DICOMServer_Path;
        page3.m_ip          = m_Configuration_DICOMServer_IP;

        if(sheet.DoModal() == IDOK)
        {
            m_Configuration_Path_Bin            = page1.m_bin;
            m_Configuration_Path_Plugin         = page1.m_plugin;
            m_Configuration_Path_Log            = page1.m_log;
            m_Configuration_Path_Temp           = page1.m_temp;

            m_Configuration_DICOMClient_APtitle    = page2.m_aptitle;
            m_Configuration_DICOMClient_StartUpDir = page2.m_startupdir;

            m_Configuration_DICOMServer_APtitle = page3.m_aptitle;
            m_Configuration_DICOMServer_Port    = page3.m_port;
            m_Configuration_DICOMServer_Path    = page3.m_serverpath;
            m_Configuration_DICOMServer_IP      = page3.m_ip;
            SaveINI();
        }

    }



    ////////////////////////////////////////////////////
    //  SOURCE AREA
    //
    ////////////////////////////////////////////////////
    void CGatewayDlg::OnButtonSourceConvert()
    {

        CConverter dlg;
        if(dlg.DoModal() == IDOK)
        {
            dlg.Convert();
        };

    }



    void CGatewayDlg::OnDblclkListSourcefile()
    {
        CString Selected, drive;
        int     drivenumber, nResult;

        m_localfile.GetText(m_localfile.GetCurSel(), Selected);

        if(strcmp("[..]", Selected) == 0)
        {
            _chdir("..//");
            UpdateLocalDir();
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if(Selected[0] == '[')
{
 if(Selected[3] == '-')
 {
      drive = Selected[2];
      drivenumber = nGetDriveNumber(drive);
      nResult = _chdrive(drivenumber);
      if(nResult == -1)
      {
        AfxMessageBox("Cannot find specified drive");
      }
      UpdateLocalDir();
 }
 else if(Selected[1] == '-')
 {
     AfxMessageBox("Device");
 }
  else
  {
      int  len, count, i;

      len = strlen(Selected);

      CString temp;

      for(count = 1, i =0; count <len -1;count ++, i++)
      {
          if(count == 1)
          {
              temp = Selected[1];
          }
          else
              temp = temp + Selected[count];
      }
      _chdir(temp);
      UpdateLocalDir();
   }
}
else
{
   CString    logpath;
   CString    fullselected;
   CString    TargetDir;
   char       selectpath[80] = "";

   getcwd(selectpath, 80);
   logpath      = m_Configuration_Path_Log + "\\log00";
   fullselected = selectpath;
   fullselected = fullselected + "\\";
   fullselected = fullselected + Selected;

   m_TargetLookin.GetWindowText(TargetDir);

   if(strlen(TargetDir) != 3)
   {
        TargetDir = TargetDir + "\\";
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        WritePrivateProfileString("Source", "input1", fullselected, logp
    ath);
        WritePrivateProfileString("Output", "FileName", TargetDir, logpa
    th);

        CConverter dlg;

        dlg.Getlog(logpath);
        dlg.DoModal();
    }
}


void CGatewayDlg::OnSelchangeListSourcefile()
{
    CString Selected;

    m_localfile.GetText(m_localfile.GetCurSel(), Selected);

    if(strcmp("[..]", Selected) == 0)
    {
        m_SourceConvert.EnableWindow(FALSE);
        m_SourceView.EnableWindow(FALSE);
        m_Put.EnableWindow(FALSE);
        m_Get.EnableWindow(FALSE);
    }
    else if(Selected[0] == '[')
    {
     if(Selected[3] == '-')
      {
        m_SourceConvert.EnableWindow(FALSE);
        m_Put.EnableWindow(FALSE);
        m_Get.EnableWindow(FALSE);
        m_SourceView.EnableWindow(FALSE);
     }
     else if(Selected[1] == '-')
     {
        m_SourceConvert.EnableWindow(TRUE);
        m_Put.EnableWindow(TRUE);
        m_Get.EnableWindow(FALSE);
        m_SourceView.EnableWindow(FALSE);
     }
      else
      {
        m_SourceConvert.EnableWindow(FALSE);
        m_Put.EnableWindow(FALSE);
        m_Get.EnableWindow(FALSE);
      }
    }
    else
    {
        m_SourceConvert.EnableWindow(FALSE);
        m_Put.EnableWindow(TRUE);
        m_Get.EnableWindow(FALSE);
        m_SourceView.EnableWindow(TRUE);
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//////////////////////////////////////////////////////
//  TARGET AREA
//
//////////////////////////////////////////////////////
void CGatewayDlg::OnButtonTargetRemote()
{

    CConnect            dlg;
    CInternetSession    *inet = NULL;
    CFtpConnection      *ftp  = NULL;
    CFtpFileFind        *ftpfilefind = NULL;

    CString  user;
    CString  pass;
    DWORD    dwServiceType = INTERNET_SERVICE_FTP;

    if(dlg.DoModal() == IDOK)
    {
        user = dlg.m_user;
        pass = dlg.m_pass;

        if(ftp != NULL)
        {
            delete ftp;
            ftp = NULL;
        }


        if(inet != NULL)
        {
            delete inet;
            inet  = NULL;
        }

        inet = new CInternetSession(NULL, PRE_CONFIG_INTERNET_ACCESS);
        if(inet == NULL)
        {
            AfxMessageBox("Cannot open internet connection");
        }


      if((dwServiceType == INTERNET_SERVICE_FTP) && !m_Configuration_
DICOMServer_IP.IsEmpty())
        {
            try
            {
                CWaitCursor cursor;
                ftp = inet->GetFtpConnection(m_Configuration_DICOMServe
r_IP, user, pass);
            }
            catch(CInternetException* pEx)
            {
                pEx->Delete();
                ftp = NULL;
                AfxMessageBox("Cannot connect to server");
                return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
        }
        else
            return;


        CWaitCursor cursor;
        CString     sDir = "";

        ftp->GetCurrentDirectory(sDir);

        CString  strSearchDir = sDir + _T("/*");

        BOOL bContinue = ftpfilefind->FindFile(strSearchDir);


        if(!bContinue)
        {
            ftpfilefind->Close();
        }


        while(bContinue)
        {
            bContinue = ftpfilefind->FindNextFile();
            CString strFileName = ftpfilefind->GetFileName();
            if(ftpfilefind->IsDirectory())
            {
                AddItem(strFileName);
            }
        }


    }


}


void CGatewayDlg::OnButtonTargetLocal()
{
    UpdateTargetLocalDir();
}


void CGatewayDlg::OnDblclkListTargetfile()
{
    CString Selected, drive;
    int     drivenumber, nResult;

    m_targetfile.GetText(m_targetfile.GetCurSel(), Selected);

    if(strcmp("[..]", Selected) == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            _chdir("..//");
            UpdateTargetLocalDir();
        }
        else if(Selected[0] == '[')
        {
         if(Selected[3] == '-')
         {
              drive = Selected[2];
              drivenumber = nGetDriveNumber(drive);
              nResult = _chdrive(drivenumber);
              if(nResult == -1)
              {
                AfxMessageBox("Cannot find specified drive");
              }
              UpdateTargetLocalDir();
         }
          else
          {
              int  len, count, i;

              len = strlen(Selected);

              CString temp;

              for(count = 1, i =0; count <len -1;count ++, i++)
              {
                   if(count == 1)
                   {
                        temp = Selected[1];
                   }
                   else
                        temp = temp + Selected[count];
              }
              _chdir(temp);
              UpdateTargetLocalDir();
          }
        }
        else
        {
            AfxMessageBox("Selected file"+Selected);
        }
    }


    void CGatewayDlg::OnSelchangeListTargetfile()
    {
        CString Selected;

        m_targetfile.GetText(m_targetfile.GetCurSel(), Selected);

        if(strcmp("[..]", Selected) == 0)
        {
            m_Get.EnableWindow(FALSE);
        }
        else if(Selected[0] == '[')
        {
         if(Selected[3] == '-')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          m_Get.EnableWindow(FALSE);
        }
        else if(Selected[1] == '-')
        {
          m_Get.EnableWindow(FALSE);
        }
         else
         {
          m_Get.EnableWindow(FALSE);
         }
       }
       else
       {
          m_Get.EnableWindow(TRUE);
       }
    }


    ////////////////////////////////////////////////////////////////////
    //
    //  Communication Function
    //
    //
    ////////////////////////////////////////////////////////////////////
    void CGatewayDlg::OnButtonPut()
    {
        CString Selected;
        CString arg;
        CDicomnetwork dcm;

        char    porttemp[20] = "";
        char    cmdline[80]  = "";

        m_localfile.GetText(m_localfile.GetCurSel(), Selected);

        sprintf(porttemp, " %d  ", m_Configuration_DICOMServer_Port);

        arg = m_Configuration_DICOMServer_IP + porttemp;

        arg = arg + Selected;
        dcm.Storscu(arg);
    }



    ////////////////////////////////////////////////////////////////////
    //
    //  Service Function
    //  Define as private:
    //
    //
    ////////////////////////////////////////////////////////////////////
    void CGatewayDlg::ReadINI()
    {
     char temp[80] = "";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int  porttemp;

    GetPrivateProfileString("Configuration", "Server Port", "", temp, size
   of(temp), "pushit.ini");
    porttemp= atoi(temp);
    m_Configuration_DICOMServer_Port     = porttemp;

    GetPrivateProfileString("Configuration", "Server IP", "", temp, sizeof
   (temp), "pushit.ini");
    m_Configuration_DICOMServer_IP       = temp;

    GetPrivateProfileString("Configuration", "Server AP_TITLE", "", temp,
   sizeof(temp), "pushit.ini");
    m_Configuration_DICOMServer_APtitle = temp;

    GetPrivateProfileString("Configuration", "Server Path", "", temp, size
   of(temp), "pushit.ini");
    m_Configuration_DICOMServer_Path     = temp;

    GetPrivateProfileString("Configuration", "Client AP_TITLE", "", temp,
   sizeof(temp), "pushit.ini");
    m_Configuration_DICOMClient_APtitle = temp;

    GetPrivateProfileString("Configuration", "Client Startup Dir", "", tem
   p, sizeof(temp), "pushit.ini");
    m_Configuration_DICOMClient_StartUpDir  = temp;

    GetPrivateProfileString("Configuration", "Client Path Bin", "", temp,
   sizeof(temp), "pushit.ini");
    m_Configuration_Path_Bin     = temp;

    GetPrivateProfileString("Configuration", "Client Path Plugin", "", tem
   p, sizeof(temp), "pushit.ini");
    m_Configuration_Path_Plugin  = temp;

    GetPrivateProfileString("Configuration", "Client Path Log", "", temp,
   sizeof(temp), "pushit.ini");
    m_Configuration_Path_Log  = temp;

    GetPrivateProfileString("Configuration", "Client Path Temp", "", temp,
    sizeof(temp), "pushit.ini");
    m_Configuration_Path_Temp  = temp;

    GetPrivateProfileString("Option", "Connect DICOM", "", temp, sizeof(te
   mp), "pushit.ini");
    if (strcmpi(temp, "TRUE") == 0)
    {
       m_Option_ConnectDicomServer = TRUE;
    }
    else
       m_Option_ConnectDicomServer = FALSE;

    GetPrivateProfileString("Option", "Show Target Path", "", temp, sizeof
   (temp), "pushit.ini");
    if (strcmpi(temp, "TRUE") == 0)
    {
       m_Option_ShowTargetPath = TRUE;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    else
       m_Option_ShowTargetPath = FALSE;


    GetPrivateProfileString("Option", "Autoconvert", "", temp, sizeof(temp
    ), "pushit.ini");
    if (strcmpi(temp, "TRUE") == 0)
    {
       m_Option_AutoConvert = TRUE;
    }
    else
       m_Option_AutoConvert = FALSE;


    AddLog("-> Load Pushit.ini");
   }



   /*
    *  Funcition : SaveINI()
    *  Purpose   : Save Configuation/Option to pushit.ini
    */
   void CGatewayDlg::SaveINI()
   {
     char temp[10] = "";

     sprintf(temp, "%d", m_Configuration_DICOMServer_Port);

     WritePrivateProfileString("Configuration", "Server Port", temp, "push
   it.ini");
     WritePrivateProfileString("Configuration", "Server IP", m_Configurati
   on_DICOMServer_IP, "pushit.ini");
     WritePrivateProfileString("Configuration", "Server AP_TITLE", m_Confi
   guration_DICOMServer_APtitle, "pushit.ini");
     WritePrivateProfileString("Configuration", "Server Path", m_Configura
   tion_DICOMServer_Path, "pushit.ini");

     WritePrivateProfileString("Configuration", "Client AP_TITLE", m_Confi
   guration_DICOMClient_APtitle, "pushit.ini");
     WritePrivateProfileString("Configuration", "Client StartUp Dir", m_Co
   nfiguration_DICOMClient_StartUpDir, "pushit.ini");

     WritePrivateProfileString("Configuration", "Client Path Bin", m_Confi
   guration_Path_Bin, "pushit.ini");
     WritePrivateProfileString("Configuration", "Client Path Plugin", m_Co
   nfiguration_Path_Plugin, "pushit.ini");
     WritePrivateProfileString("Configuration", "Client Path Log", m_Confi
   guration_Path_Log, "pushit.ini");
     WritePrivateProfileString("Configuration", "Client Path Temp", m_Conf
   iguration_Path_Temp, "pushit.ini");

     if(m_Option_ConnectDicomServer == TRUE)
         sprintf(temp, "TRUE");
     else
         sprintf(temp, "FALSE");
     WritePrivateProfileString("Option", "Connect DICOM", temp, "pushit.in
   i");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(m_Option_ShowTargetPath == TRUE)
        sprintf(temp, "TRUE");
    else
        sprintf(temp, "FALSE");
    WritePrivateProfileString("Option", "Show Target Path", temp, "pushit
.ini");

    if(m_Option_AutoConvert == TRUE)
        sprintf(temp, "TRUE");
    else
        sprintf(temp, "FALSE");
    WritePrivateProfileString("Option", "Autoconvert", temp, "pushit.ini"
);

    AddLog("-> Save Pushit.ini");
}


void CGatewayDlg::StartUpControl()
{
    char temp[80] = "";

    m_TargetRemote.EnableWindow(TRUE);
    m_TargetLocal.EnableWindow(TRUE);
    m_SourceView.EnableWindow(FALSE);
    m_SourceEdit.EnableWindow(FALSE);
    m_SourceConvert.EnableWindow(TRUE);
    m_SourceBrowse.EnableWindow(FALSE);
    m_Put.EnableWindow(FALSE);
    m_Get.EnableWindow(TRUE);
    m_Option.EnableWindow(TRUE);
    m_Help.EnableWindow(TRUE);
    m_Configuration.EnableWindow(TRUE);

    sprintf(temp, "%s", m_Configuration_DICOMClient_StartUpDir);
    _chdir(temp);
    AddLog("-> Setting Startup Control");
}


void CGatewayDlg::UpdateLocalDir()
{
    char temp[80] = "";

    DlgDirList("*.*", IDC_LIST_SOURCEFILE, NULL, DDL_DIRECTORY | DDL_DR
IVES);
    m_localfile.AddString("[-Snappy-]");
    getcwd(temp, 80);
    m_SourceLookin.SetWindowText(temp);
    AddLog("-> Update Source Directory");
}


void CGatewayDlg::UpdateTargetLocalDir()
{
    char temp[80] = "";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    char tmp[90]  = "";

    DlgDirList("*.*", IDC_LIST_TARGETFILE, NULL, DDL_DIRECTORY | DDL_DR
IVES);
    getcwd(temp, 80);
    sprintf(tmp, "%s", temp);
    m_TargetLookin.SetWindowText(tmp);
    AddLog("-> Update Target Directory");
}


void CGatewayDlg::AddLog(CString Log)
{
    count = count++;
    m_message += Log + "\r\n";
    UpdateData(FALSE);
    if(count > 2)
    {
       SendDlgItemMessage(IDC_EDIT_MESSAGE, WM_VSCROLL, SB_BOTTOM, NULL
);
    }

}


int CGatewayDlg::nGetDriveNumber(CString driveletter)
{
    switch(driveletter[0])
    {
        case 0x61:return 1; //a
        case 0x62:return 2; //b
        case 0x63:return 3; //c
        case 0x64:return 4; //d
        case 0x65:return 5; //e
        case 0x66:return 6; //f
        case 0x67:return 7; //g
        case 0x68:return 8; //h
        case 0x69:return 9; //i
        case 0x6a:return 10;//j
        case 0x6b:return 11;//k
        case 0x6c:return 12;//l
        case 0x6d:return 13;//m
        case 0x6e:return 14;//n
        case 0x6f:return 15;//o
        case 0x70:return 16;//p
        case 0x71:return 17;//q
        case 0x72:return 18;//r
        case 0x73:return 19;//s
        case 0x74:return 20;//t
        case 0x75:return 21;//u
        case 0x76:return 22;//v
        case 0x77:return 23;//w
        case 0x78:return 24;//x
        case 0x79:return 25;//y
        case 0x7a:return 26;//z
    }
    return 0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void CGatewayDlg::AddItem(CString sItem)
{
    SendDlgItemMessage(IDC_LIST_TARGETFILE, LB_ADDSTRING, 0, (long)(LPC
STR)sItem);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดีเนื่องมาจากความกรุณาจากหลายฝ่าย ขอขอบพระคุณผศ.ดร.สุวิพล สิทธิชีวภาพ อาจารย์ที่ปรึกษาที่ได้ให้คำแนะนำในการทำงานโดยเฉพาะอย่างยิ่งความรู้ในทางการสื่อสารข้อมูล ขอพระคุณอาจารย์เกรียงไกร โรจนาภรณ์ ที่ให้ความกรุณาแนะนำในการเขียนโปรแกรมและได้เอื้อเฟื้อเครื่องคอมพิวเตอร์เพื่อติดตั้งโปรแกรมที่ในการทดลอง ขอบคุณ ดร.สุธี ผู้เจริญชนะชัย ที่ชักนำและได้ให้โอกาสสำคัญอันเป็นจุดเริ่มต้นในการทำโครงงานนี้ อาจารย์สมชาย เกรียงอารีกุล ที่สอนการทำงานอย่างเป็นระบบให้แก่ผู้จัดทำ อาจารย์บุญชัย เรืองสุขนุกูลให้คำแนะนำอันเป็นประโยชน์มากเกี่ยวกับระบบปฏิบัติการยูนิกส์ นายแพทย์ราเมศว์ วัชรสินธุ์ ที่ได้ให้โอกาสผู้จัดทำได้ทำการทดลองกับเครื่องมือแพทย์อันทันสมัยไม่ว่าจะเป็นเครื่องเอกซเรย์คอมพิวเตอร์ความเร็วสูง เครื่องถ่ายภาพกำทอนแม่เหล็ก ณ ภาควิชารังสีวิทยา โรงพยาบาลรามาธิบดี

ขอบคุณเจ้าหน้าที่ห้องทดลองเครื่องคอมพิวเตอร์ถ่ายภาพตัดขวางร่างกาย และเจ้าหน้าที่สำนักวิจัยและบริการคอมพิวเตอร์ทุกท่าน คุณอดิศักดิ์ แข็งสาริกิจ คุณวิโรจน์ แสงธงทอง รุ่นพี่ที่ได้ให้คำแนะนำต่างๆ คุณปรเมศว์ ห่อแก้ว รุ่นน้องผู้มากความสามารถที่ได้ร่วมงานกันมาหลายโครงการ คุณธีรเดช คุณอรรถพล และทุกๆคนที่ไม่ได้กล่าวถึงที่ได้สร้างสีสรร และบรรยากาศอันดีในการทำงาน

ขอบคุณ คุณพ่อ คุณแม่ และทุกคนในครอบครัว คุณจันทณี อาชาพานิช ที่ได้สนับสนุนและเข้าใจในสิ่งที่ผู้จัดทำ ทำตลอดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] NEMA, "Digital Imaging and Communications in Medicine Part 1: Introduction and Overview", March 27, 1992, NEMA Standards Publication

[2] ผศ.ชวลิต วงษ์เอก, "Hello DICOM", วารสารชมรมข้อมูล ข่าวสารทาง การแพทย์และสาธารณสุขไทย, ฉบับที่ 1 พ.ศ. 2539, หน้า 55-62.

[3] สมชาย เกรียงอารีกุล, สุธี ผู้เจริญชนะชัย, สุรณรงค์ คามตะศิลา, "เทคนิคการแปลงและจัดเก็บภาพทางการแพทย์ด้วยมาตรฐาน DICOM", ประชุมวิชาการทางวิศวกรรมไฟฟ้าครั้งที่ 20

[4] Osman Ratib, Yves Ligier, Christian Giard, Marianne Logean, Ronald Welz, Silvia Dragomir, "Osiris Medical Imaging Software", http://expasy.hcuge.ch/UIN.

[5] Radiological Society of North America, "www.rsna.com"

[6] Stephen T.C. Wong, H.K. Huang, "Networked Multimedia for Medical Imaging", IEEE Multimedia in Medicine April-June 1997

[7] NEMA, "Digital Imaging and Communications in Medicine Part 5:Data Structures and Encoding", October 29, 1993, NEMA Standards Publication

[8] NEMA, "Digital Imaging and Communications in Medicine Part 3: Information Object Definitions", October 29, 1993, NEMA Standards Publication

[9] NEMA, "Digital Imaging and Communications in Medicine Part 8: Network Communications Support for Message Exchange", December 13, 1993, NEMA Standards Publication

[10] NEMA, "Digital Imaging and Communications in Medicine Part 6: Data Dictionary", October 29, 1993, NEMA Standards Publication

[11] NEMA, "Digital Imaging and Communications in Medicine Part 7: Message Exchange", October 29, 1993, NEMA Standards Publication

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้