

## LỜI CẢM ƠN

Trước hết, em xin chân thành gửi lời cảm ơn sâu sắc đến cô giáo **ThS.Nguyễn Thị Xuân Hương**, người đã tận tình hướng dẫn và tạo mọi điều kiện cho em trong quá trình làm tốt nghiệp.

Em xin chân thành cảm ơn các thầy cô giáo trong khoa Công Nghệ Thông Tin Trường Đại Học Dân Lập Hải Phòng đã truyền đạt những kiến thức quý báu và giúp đỡ em trong suốt bốn năm học và trong quá trình làm tốt nghiệp vừa qua.

Em xin trân trọng cảm ơn **GS.TS.NGƯT Trần Hữu Nghị** - Hiệu trưởng trường Đại Học Dân Lập Hải Phòng đã ủng hộ, động viên, và tạo mọi điều kiện tốt nhất cho chúng em trong thời gian học tập tại trường.

Cuối cùng tôi xin gửi lời cảm ơn chân thành tới tất cả những người thân cùng bạn bè đã động viên, giúp đỡ và đóng góp nhiều ý kiến quý báu cho tôi trong quá trình học tập cũng như khi làm tốt nghiệp.

*Hải Phòng, tháng 7 năm 2010*

*Sinh viên*

**Nguyễn Mạnh Tuân**

# MỤC LỤC

## LỜI CẢM ƠN

LỜI MỞ ĐẦU .....	1
------------------	---

## Chương 1: TỔNG QUAN VỀ CÁC CHUẨN LUU TRỮ ẢNH TRONG

Y KHOA .....	3
--------------	---

1.1. Analysis of Functional NeuroImaging – AFNI .....	3
---	---

1.2. Analyse.....	4
-------------------	---

1.3. DICOM .....	5
------------------	---

Chương 2: CHUẨN ẢNH DICOM .....	7
---------------------------------	---

2.1. Giới thiệu chung.....	7
----------------------------	---

2.2. Chuẩn ảnh DICOM .....	8
----------------------------	---

2.2.1. File DICOM .....	9
-------------------------	---

2.2.2. Giao thức DICOM.....	16
-----------------------------	----

2.2.2.1. Tổng quan về giao thức .....	16
---------------------------------------	----

2.2.2.2. Dịch vụ DICOM .....	18
------------------------------	----

2.2.2.2.1. Dịch vụ DIMSE.....	19
-------------------------------	----

2.2.2.2.2. Dịch vụ Association .....	20
--------------------------------------	----

2.2.2.2.3. Dịch vụ Upper Layer.....	21
-------------------------------------	----

Chương 3: GIỚI THIỆU VỀ NGÔN NGỮ LẬP TRÌNH C#.....	27
--	----

3.1. Lịch sử ra đời của ngôn ngữ lập trình C# .....	27
---	----

3.2. Đặc điểm của ngôn ngữ lập trình C# .....	27
---	----

3.3. Bảng từ khóa C#.....	28
---------------------------	----

3.4. Công nghệ .NET .....	29
---------------------------	----

3.4.1. Giới thiệu .....	29
-------------------------	----

3.4.2. .NET FRAMEWORK.....	30
----------------------------	----

Chương 4: CÀI ĐẶT VÀ THỬ NGHIỆM CHƯƠNG TRÌNH .....	33
--	----

4.1. Môi trường cài đặt.....	33
------------------------------	----

4.2. Tập ảnh thử nghiệm .....	33
-------------------------------	----

4.3. Giao diện của chương trình.....	34
--------------------------------------	----

4.3.1.Giao diện chính của chương trình.....	34
---	----

4.3.2.Giao diện hiển thị ảnh DICOM.....	35
---	----

4.3.3. Giao diện hiển thị thông số của ảnh DICOM.....	36
---	----

4.3.4. Giao diện lưu ảnh sang định dạng PNG .....	37
---	----

KẾT LUẬN .....	38
----------------	----

TÀI LIỆU THAM KHẢO .....	39
--------------------------	----

## LỜI MỞ ĐẦU

Ngày nay, lĩnh vực y tế đang giữ một vai trò vô cùng quan trọng đối với sự phát triển của con người cũng như toàn xã hội. Với việc dân số toàn cầu không ngừng tăng lên kèm theo đó là sự xuất hiện của nhiều căn bệnh mới đòi hỏi ngành y tế cần phải nỗ lực hơn nữa trong việc chăm sóc sức khỏe cho cộng đồng. Để làm được điều này, việc ứng dụng công nghệ thông tin vào trong lĩnh vực y tế là rất cần thiết. Tuy nhiên trong một thời gian dài, việc ứng dụng công nghệ thông tin trong ngành y tế mới chỉ dừng lại ở công tác quản lý hành chính và viện phí mà chưa đáp ứng được nhu cầu khám chữa bệnh của người dân.

Thực tế đã bắt đầu thay đổi, nhiều bệnh viện đặc biệt là các bệnh viện tư mới ra đời đã nhanh chóng thiết đặt các hệ thống hiện đại phục vụ cho công tác quản lý đồng thời không ngừng tăng cường đầu tư máy móc công nghệ hiện đại như máy chụp cắt lớp, máy chụp Xquang, chụp cộng hưởng từ để hỗ trợ cho việc chuẩn đoán bệnh. Ảnh được chụp từ các loại máy này được lưu trữ theo các chuẩn ảnh để phục vụ trong lĩnh vực y tế. Các ảnh này được các bác sĩ sử dụng để đọc các thông tin phục vụ cho việc chẩn đoán bệnh. Một trong những chuẩn ảnh được sử dụng phổ biến nhất hiện nay là chuẩn ảnh DICOM (Digital Imaging and Communications in Medicine). Việc tìm hiểu chuẩn lưu trữ của ảnh DICOM và có thể đọc để từ đó có thể áp dụng các kỹ thuật xử lý ảnh để xử lý dữ liệu, từ đó hỗ trợ cho việc chuẩn đoán bệnh là một công việc có ý nghĩa rất lớn.

Đó cũng là lý do em chọn đề tài nghiên cứu” Tìm hiểu về chuẩn lưu trữ ảnh DICOM và viết chương trình đọc ảnh DICOM” cho luận văn tốt nghiệp của mình.

Luận văn gồm 4 chương:

Chương 1: Tổng quan về các chuẩn lưu trữ ảnh trong y khoa: trình bày những kiến thức cơ bản nhất về các chuẩn AFNI, Analyze và chuẩn DICOM.

Chương 2: Chuẩn ảnh DICOM: trình bày về lịch sử ra đời, file DICOM, giao thức DICOM và dịch vụ DICOM.

Chương 3: Giới thiệu về ngôn ngữ lập trình C#: trình bày về lịch sử ra đời, các đặc điểm của ngôn ngữ lập trình C# và công nghệ .NET.

Chương 4: Cài đặt thử nghiệm chương trình.

Cuối cùng là phần kết luận.

# Chương 1: TỔNG QUAN VỀ CÁC CHUẨN LUU TRỮ ẢNH TRONG Y KHOA

## 1.1. Analysis of Functional NeuroImaging – AFNI

- AFNI (Analysis of Functional NeuroImaging) là một môi trường xử lý, phân tích và hiển thị fMRI data – một kỹ thuật mô phỏng hoạt động của bộ não con người. AFNI chạy trên hệ thống Unix+X11+MOTIF, bao gồm cả SGI và Linux.

- ANFI được viết bằng ngôn ngữ C, được phát triển rất mạnh ở đại học y dược Wisconsin vào năm 1994 và sau này Robert W. Cox phát triển thêm. Việc phát triển này mang lại nhiều điểm nhấn trong NIH (National Institutes of Health) vào năm 2001 và tiếp tục phát triển ở NIMH Scientific and Statistical Computing Core.

- AFNI lưu trữ thông tin vào 2 file:
  - File BRIK lưu trữ dữ liệu.
  - File ACII HEAD lưu trữ các thông tin header.



Hình 1: Chương trình phần mềm AFNI

## 1.2. Analyse

- Analyse là chương trình phần mềm mạnh do BIR (Biomedical Imaging Resource) ở Mayo Clinic phát triển, dùng trong hiển thị, xử lí và đo đặc các ảnh đa chiều trong y khoa. Analyse được sử dụng để lấy các ảnh chụp từ MRI, CT và PET.

- Định dạng file trong Analyse 7.5 đã được sử dụng sâu rộng trên lĩnh vực xử lí ảnh não bộ thần kinh, và các chương trình khác như SPM (Statistical Parametric Mapping), AIR, MRIcro có thể đọc và ghi định dạng đó. Những file có thể được sử dụng để lưu trữ những hình khối đa chiều.

- Một mục dữ liệu gồm hai file :
  - Một file chứa dữ liệu kiểu binary với phần mở rộng .img
  - Một file chứa metadata với phần mở rộng .hdr



Hình 2: Chương trình phần mềm Analyse

### 1.3. DICOM

- DICOM (Digital Imaging and Communications in Medicine) là tập hợp các chuẩn dùng trong xử lý, truyền tải thông tin, lưu trữ và in ảnh y khoa. Chuẩn này bao gồm định dạng file và giao thức truyền tin qua mạng. File DICOM được trao đổi giữa 2 chương trình và các chương trình này có thể nhận ảnh và dữ liệu bệnh nhân theo định dạng DICOM.

- DICOM cho phép tích hợp máy scan, server, trạm làm việc, máy in và các thiết bị mạng từ nhiều nhà cung cấp vào thành một hệ thống truyền tải và lưu trữ ảnh. Ngày nay, hầu hết các bệnh viện trên thế giới đều áp dụng DICOM vào trong các thiết bị y khoa, máy trạm, server, các hệ thống quản lý trong hoạt động khám và chữa bệnh.

- Các Modality hỗ trợ DICOM.

Viết tắt	Tên đầy đủ	Viết tắt	Tên đầy đủ
AS	Angioscopy	LS	Laser Surface Scan
BI	Biomagnetic Imaging	MA	Magnetic Resonance Angiography
CD	Color Flow Doppler	MR	Magnetic Resonance
CP	Culposcopy	MS	Magnetic Resonance Spectroscopy
CR	Computed Radiography	NM	Nuclear Medicine
CS	Cystoscopy	PT	Positron Emission Tomography
CT	Computed Tomography	RF	Radio Fluoroscopy
DD	Duplex Doppler	RG	Radiographic Imaging
DG	Diaphanography	RTDOSE	Radiotherapy Dose

DM	Digital Microscopy	RTIMAGE	Radiotherapy Image
DS	Digital Subtraction Angiography	RTPLAN	Radiotherapy Plan
DX	Digital Radiography	RTSTRUCT	Radiotherapy Structure Set
EC	Echocardiography	ST	Single-photon Emission Computed Tomography
ES	Endoscopy	TG	Thermography
FA	Fluorescein Angiography	US	Ultrasound
FS	Fundoscopy	XA	X-Ray Angiography
HC	Hard Copy	ECG	Electrocardiograms
LP	Laparoscopy		

## **Chương 2: CHUẨN ẢNH DICOM**

### **2.1. Giới thiệu chung**

- Vào năm 1970, trước sự ra đời của phương pháp chụp ảnh CT (Computed Tomography) cùng với các phương pháp chụp ảnh số dùng trong chẩn đoán y khoa khác, và sự gia tăng nhanh chóng ứng dụng tin học trong các lĩnh vực y khoa lâm sàng, hai tổ chức ACR (American College of Radiology) và NEMA (National Electrical Manufacturers Association) đã nhận ra yêu cầu cần thiết phải có một phương pháp chuẩn dùng trong truyền tải ảnh và thông tin liên quan đến ảnh đó giữa các nhà sản xuất thiết bị y khoa, mặc dù những thiết bị đó lại cho ra các định dạng ảnh khác nhau. Trong năm 1983, ACR và NEMA thành lập một ủy ban chung để phát triển phương pháp chuẩn này với mục đích:

- Tăng cường khả năng giao tiếp thông tin ảnh số của thiết bị y khoa bất chấp thiết bị đó là của nhà sản xuất nào.
  - Giúp cho việc phát triển và mở rộng các hệ thống truyền tải và lưu trữ ảnh trở nên dễ dàng hơn, từ đó các hệ thống này sẽ là nơi giao tiếp với các hệ thống thông tin bệnh viện khác.
  - Cho phép tạo ra thông tin cở sở chẩn đoán, từ đó nhiều loại thiết bị chẩn bệnh sẽ sử dụng và tra cứu thông tin này.
- ACR-NEMA công bố "ACR-NEMA Standards Publication" phiên bản 1.0 vào năm 1985. Và năm 1988, ủy ban này công bố tiếp "ACR-NEMA Standards Publication" phiên bản 2.0. Tài liệu "ACR-NEMA Standards Publication" đặc tả giao tiếp phần cứng, số lượng tối thiểu các lệnh phần mềm và các định dạng dữ liệu.
- Chuẩn DICOM (**Digital Imaging and Communications in Medicine**) đưa ra nhiều cải tiến qua trọng so với 2 phiên bản của chuẩn ACR-NEMA trước:

- Chuẩn DICOM này áp dụng được trong môi trường mạng vì chúng dùng giao thức mạng chuẩn là TCP/IP. Chuẩn ACR-NEMA chỉ có thể áp dụng cho mạng point-to-point.
- Chuẩn DICOM áp dụng cho môi trường lưu trữ off-line, DICOM dùng các thiết bị lưu trữ chuẩn như CD-R, MOD và filesystem luận lý như ISO 9660 và FAT16. Chuẩn ACR-NEMA không đặc tả định dạng file, thiết bị lưu trữ vật lý hay filesystem luận lý.
- Chuẩn DICOM đặc tả các thiết bị y khoa cần tuân theo chuẩn DICOM sẽ phải đáp ứng lệnh và dữ liệu như thế nào. Chuẩn ACR-NEMA bị giới hạn về truyền tải dữ liệu, DICOM dùng khái niệm Service Classes để mô tả ngữ nghĩa lệnh và dữ liệu đi kèm.
- DICOM có kèm đặc tả về yêu cầu, quy tắc cho các nhà sản xuất thiết bị y khoa sản xuất sản phẩm tuân theo chuẩn DICOM. Chuẩn ACR-NEMA đặc tả rất ít về điều này.

- Hướng phát triển hiện thời: chuẩn DICOM luôn phát triển và do *Procedures of the DICOM Standards Committee* quản lý. Đề nghị nâng cấp trong tương lai của các thành viên trong ủy ban DICOM dựa trên thông tin từ các những người đã dùng qua chuẩn DICOM. Các ý kiến được xem xét để đưa vào phiên bản tiếp theo của DICOM và các thay đổi của DICOM phải đảm bảo tương thích tốt với phiên bản trước.

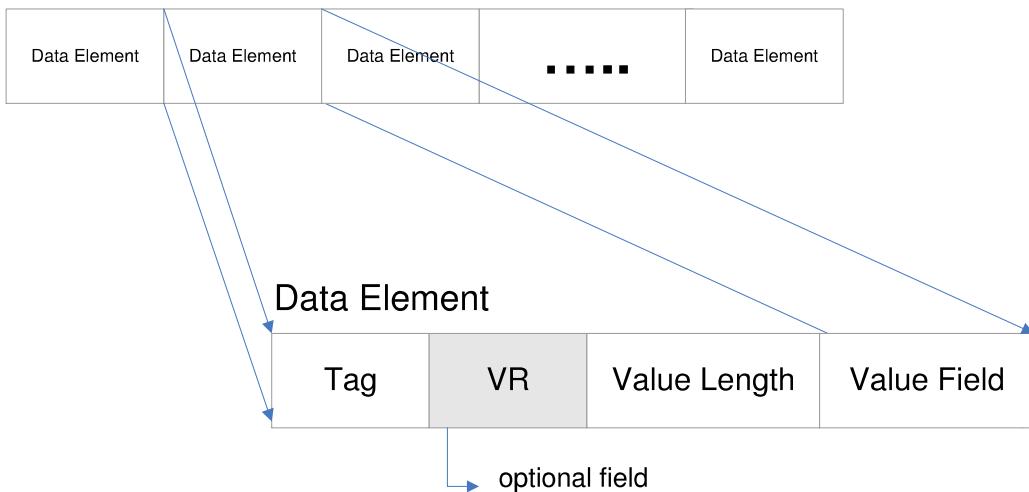
## 2.2. Chuẩn ảnh DICOM

- Đặc tả DICOM áp dụng cho:
  - Định dạng file ảnh dùng trong y khoa.
  - Giao thức truyền thông dữ liệu DICOM.

### 2.2.1. File DICOM

- File DICOM là file lưu trữ theo định dạng DICOM. File này lưu trữ những thông tin sau:
  - Thông tin bệnh nhân.
  - Thông tin về lần khám của ảnh.
  - Thông tin lượt viếng thăm.
  - Thông tin về thiết bị y khoa đã sinh ra ảnh.
  - Ảnh của bệnh nhân.
- DICOM hỗ trợ các định dạng ảnh JPEG, JPEG Lossless, JPEG 2000, LZW và Run-length encoding (RLE).
- Cấu trúc căn bản của file DICOM là Data Set.

Data Set



Hình 3: Cấu tạo Data Set

- Các khái niệm trong DICOM.

Khái niệm	Ý nghĩa
<b>Data Set</b>	<ul style="list-style-type: none"> <li>- Là tập hợp nhiều Data Element trong một file DICOM.</li> </ul>
<b>Data Element</b>	<ul style="list-style-type: none"> <li>- Là một đơn vị thông tin trong DICOM file. Data Element chứa <u>một thông tin đầy đủ</u>. Các field trong Data Element có nhiệm vụ đặc tả đầy đủ một thông tin, đặc tả bao gồm: ý nghĩa, giá trị, chiều dài của tin và định dạng dữ liệu của tin.</li> </ul>
<b>Tag</b>	<ul style="list-style-type: none"> <li>- Là 2 số nguyên không dấu, mỗi số 16 bit. Cặp số nguyên này xác định ý nghĩa của Data Element như tên bệnh nhân, chiều cao của ảnh, số bit màu, ... Một số xác định Group Number và số kia xác định Element Number.</li> <li>- Giá trị của Group Number và Element Number cho biết Data Element nói lên thông tin nào. Các thông tin (Data Element) cùng liên quan đến một nhóm ngữ nghĩa sẽ có chung số Group Number.</li> </ul>
<b>VR</b> (Value Representation)	<ul style="list-style-type: none"> <li>- Đây là field tùy chọn, tùy vào giá trị của Transfer Syntax mà VR có mặt trong Data Element hay không.</li> <li>- Giá trị của VR cho biết kiểu dữ liệu và định dạng giá trị của Data Element.</li> </ul>
<b>VM</b> (Value Multiplicity)	<ul style="list-style-type: none"> <li>- Cho biết số lượng Value của Value Field nếu Value Field có nhiều giá trị.</li> </ul>

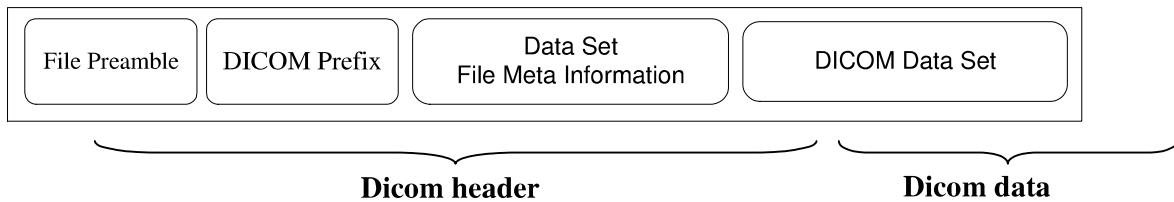
	<ul style="list-style-type: none"> <li>- Nếu số lượng Value không xác định, VM sẽ có dạng “a-b” với a số giá trị Value nhỏ nhất và b là số Value lớn nhất có thể có của Data Element.</li> </ul> <p>VD: VM = “6-10”: Value Field có ít nhất là 6 giá trị và nhiều nhất là 10 giá trị.</p> <ul style="list-style-type: none"> <li>- Data Element với Value Field có nhiều giá trị sẽ <ul style="list-style-type: none"> <li>▪ Với chuỗi kí tự, dùng kí tự 5Ch (\") làm kí tự phân cách.</li> <li>▪ Với giá trị nhị phân, không có kí tự phân cách.</li> </ul> </li> </ul>
<b>Value Length</b>	<ul style="list-style-type: none"> <li>- Là một số nguyên không dấu, có độ dài là 16 hay 32 bit. Giá trị của Value Length cho biết độ lớn (tính theo byte) của Value Field (không phải là độ lớn của toàn bộ Data Element).</li> <li>- Giá trị của Value Length là FFFFFFFFh (32 bit) hàm ý không xác định được chiều dài (Undefined Length).</li> </ul>
<b>Value Field</b>	<ul style="list-style-type: none"> <li>- Là nội dung thông tin (Data Element). Kiểu dữ liệu của field này do VR quy định và độ lớn (tính theo byte) nằm trong Value Length.</li> </ul>
<b>Transfer Syntax</b>	<ul style="list-style-type: none"> <li>- Transfer Syntax là các quy ước định dạng dữ liệu. Giá trị của Transfer Syntax cho biết cách dữ liệu được định dạng và mã hóa trong DICOM đồng thời cũng cho biết VR sẽ có tồn tại trong Data Element hay không.</li> </ul>

	<ul style="list-style-type: none"> <li>- Mặc định ban đầu, Transfer Syntax của file DICOM là <b>Explicit VR LittleEndian Transfer Syntax</b>.</li> </ul>
<b>Information Object Definition (IOD)</b>	<ul style="list-style-type: none"> <li>- IOD đại diện cho một đối tượng chứa thông tin và đối tượng này có tồn tại trong thế giới thực. Thông tin của đối tượng IOD là thông tin của đối tượng trong thế giới thực.</li> <li>- Có 2 loại IOD: <ul style="list-style-type: none"> <li>▪ Composite IOD: là IOD đại diện cho những phần khác nhau của các đối tượng khác nhau trong thế giới thực.</li> <li>▪ Normalized IOD: là IOD cho duy nhất một đối tượng trong thế giới thực.</li> </ul> </li> </ul>
<b>Lớp Service-Object Pair (SOP)</b>	<ul style="list-style-type: none"> <li>- Lớp SOP được tạo ra khi ghép một IOD với DIMSE Service dành cho IOD đó.</li> <li>- Có 2 loại lớp SOP: <ul style="list-style-type: none"> <li>▪ Lớp Normalized SOP: được tạo ra khi ghép Normalized IOD với các dịch vụ DIMSE-N.</li> <li>▪ Lớp Composite SOP: được tạo ra khi ghép Composite IOD với các dịch vụ DIMSE-C.</li> </ul> </li> </ul>

- Thứ tự của chuỗi byte: một giá trị sẽ được lưu thành một hay nhiều byte trong file. Có 2 quy ước quy định thứ tự xuất hiện của các byte của một giá trị nào đó trong file DICOM.

<b>LittleEndian</b>	<ul style="list-style-type: none"> <li>- Đối với số nhị phân gồm nhiều byte thì byte có trọng số thấp nhất (Least Significant Byte) sẽ nằm trước, những byte còn lại có trọng số tăng dần nằm tiếp sau đó.</li> <li>- Đối với chuỗi kí tự, các kí tự sẽ nằm theo thứ tự xuất hiện trong chuỗi (từ trái sang phải).</li> </ul>
<b>BigEndian</b>	<ul style="list-style-type: none"> <li>- Đối với số nhị phân gồm nhiều byte thì byte có trọng số lớn nhất (Most Significant Byte) sẽ nằm trước, những byte còn lại có trọng số giảm dần nằm tiếp sau đó.</li> <li>- Đối với chuỗi kí tự, các kí tự sẽ nằm theo thứ tự xuất hiện trong chuỗi (từ trái sang phải).</li> </ul>

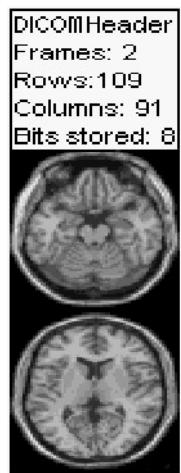
- Cấu trúc file DICOM:



- Một file DICOM đơn bao gồm phần header (lưu trữ thông tin về tên bệnh nhân, kiểu ảnh, kích thước ảnh, ...) cũng như phần data (có thể chứa đựng thông tin trong không gian 3D). Sự khác nhau bắt nguồn từ việc phân tích định dạng ảnh phổ biến, đó là dữ liệu ảnh được lưu trữ trong một file (\*.img) và dữ liệu header được lưu trữ trong một file khác (\*.hdr). Một điểm khác nhau nữa là dữ liệu ảnh DICOM có thể được nén để giảm bớt kích thước ảnh.

## DICOM header

- Hình bên hiển thị một giả thuyết về file ảnh DICOM. Trong ví dụ này, 794 bytes đầu tiên được sử dụng cho phần header của file ảnh DICOM, nó mô tả kích thước ảnh và giữ lại thông tin văn bản khác về sự nội soi cắt lớp. Kích thước của header này thay đổi phụ thuộc vào thông tin header được lưu trữ. Ở đây, header định nghĩa một file ảnh có kích thước 109x91x2 pixels



với độ phân giải dữ liệu 1byte/pixel (như vậy toàn bộ kích thước của ảnh sẽ là 19838). Dữ liệu ảnh đi theo thông tin header (header và dữ liệu ảnh được lưu trữ trong cùng một file).

- Thông tin header bao gồm thông tin bộ định danh dữ liệu được đưa vào file, nó bắt đầu bởi 128 bytes Preamble được đưa về 00H. Sau đó là 4 bytes ký tự “DICM” dùng để nhận dạng file DICOM. Các thành phần dữ liệu đầu file bắt buộc với mọi file DICOM gồm các trường tag (nhãn) có dạng (0002, xxxx) với xxxx là số thành phần bất kỳ, VR – giá trị thể hiện được mã hóa theo cú pháp chuyển đổi giá trị thể hiện ẩn I-VR (Implicit Value Representation), VL – chiều dài giá trị, gồm 4 bytes chứa chiều dài trường giá trị Value (Value Length), trường Value gồm một số chẵn bytes chứa giá trị của thành phần dữ liệu. Thứ tự các byte quy định theo kiểu Little.

Chi tiết cấu trúc DICOM header:

Tên Data Element	Tag	Mô tả
File Preamble	Không có	Đây là chuỗi byte đầu tiên của file, có chiều dài là 128 byte dành cho chương trình xử lý file DICOM sử dụng. Nếu không sử dụng thì 128 byte này đều có nội dung là 00h.

DICOM Prefix	Không có	4 byte là chuỗi “DICM”. Prefix này để xác định file có phải là DICOM file hay không.
<u>File Meta Information</u>		
Group Length	(0002,0000)	Độ lớn của Data Set File Meta Information (tính theo byte). Số byte này được tính từ Data Element theo ngay sau Data Element Group Length này.
File Meta Information Version	(0002,0001)	Xác định phiên bản của File Meta Information.
Media Storage SOP Class UID	(0002,0002)	Chuỗi UID cho SOP Class xác định định dạng lưu trữ của file DICOM.
Media Storage SOP Instance UID	(0002,0003)	Chuỗi UID cho bản thân file DICOM.
Transfer Syntax UID	(0002,0010)	Chuỗi UID cho Transfer Syntax sẽ dùng cho các Data Element nằm ở Data Set sau Data Set File Meta Information.
Implementation Class UID	(0002,0012)	Chuỗi UID của chương trình đã tạo ra file DICOM này.
Implementation Version Name	(0002,0013)	Phiên bản của chương trình tạo file DICOM có UID như trên.
Source Application Entity Title	(0002,0016)	Chuỗi tiêu đề cho Application Entity đã tạo ra file DICOM.
Private Information Creator UID	(0002,0100)	Chuỗi UID của người cung cấp thông tin riêng tư (xem bên dưới).
Private Information	(0002,0102)	Thông tin riêng tư.

- Các Data Element ở đầu file cung cấp một số thông tin ban đầu quan trọng. Chúng nằm trong một Data Set tên File Meta Information. Sau Data Set File Meta Information là đến những Data Element bình thường, các Data Element này là nội dung DICOM file (gồm hình ảnh, thông tin hình ảnh, thông tin khám, thông tin bệnh nhân).

- Ban đầu các Data Set File Meta Information được định dạng, mã hóa theo Transfer Syntax là Explicit VR Little Endian Transfer Syntax. Các Data Element nằm trong Data Set ngay sau Data Set File Meta Information sẽ có định dạng và được mã hóa theo Transfer Syntax quy định bởi UID của Transfer Syntax trong File Meta Information.

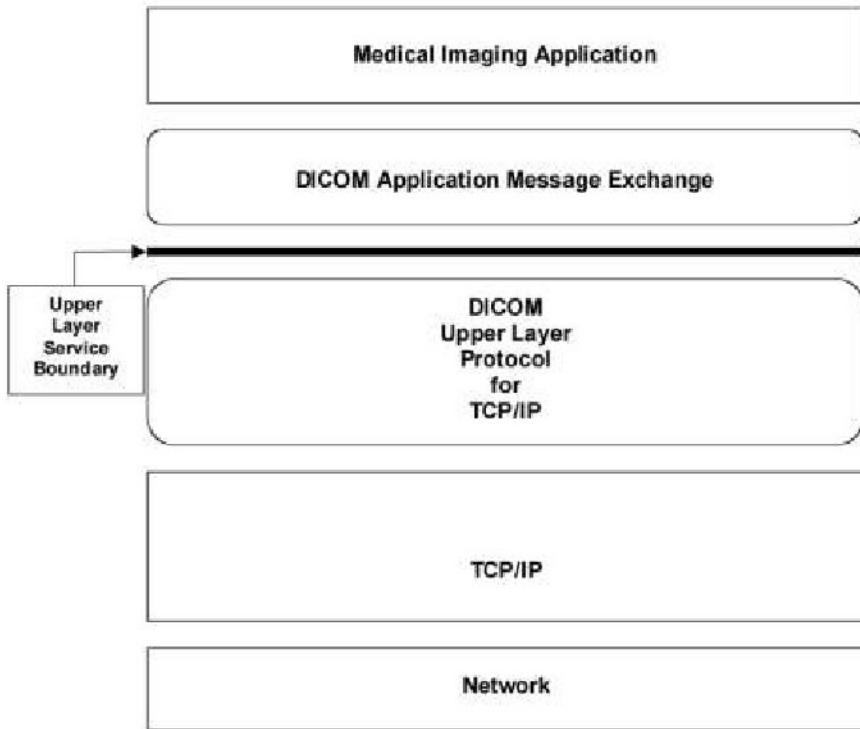
- Với các Transfer Syntax quy ước không cần VR trong Data Element, cần tra cứu trong Data Dictionary để biết VR mặc định của từng Data Element.

## **2.2.2. Giao thức DICOM**

### **2.2.2.1. Tổng quan về giao thức**

- Các ứng dụng DICOM (xem, xử lý và quản lý ảnh DICOM) giao tiếp thông tin với nhau qua các dịch vụ DICOM và sử dụng giao thức DICOM để truyền tải thông tin. Giao thức DICOM dựa trên TCP/IP để truyền tải dữ liệu.

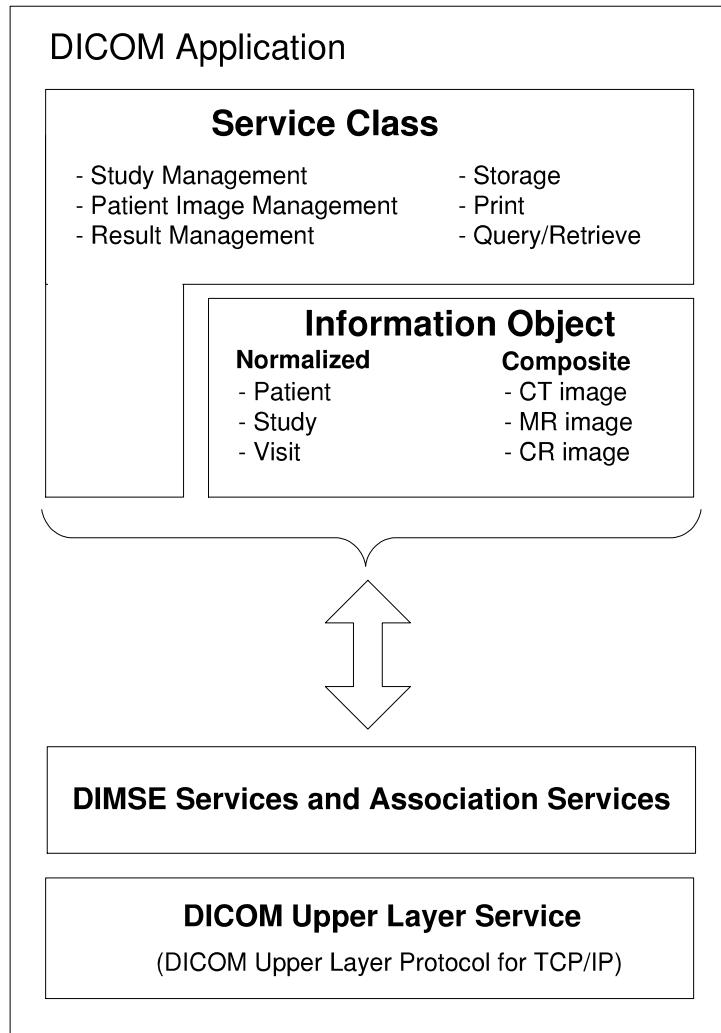
- Kiến trúc của giao thức DICOM:



*Hình 4: Kiến trúc của giao thức DICOM*

### 2.2.2.2. Dịch vụ DICOM

- Mô hình dịch vụ DICOM.



Hình 5: Mô hình dịch vụ DICOM

- Các ứng dụng DICOM giao tiếp và hoạt động trong môi trường mạng đều thông qua các dịch vụ DICOM. Mỗi dịch vụ DICOM phục vụ cho một công việc cụ thể.
- Khi ứng dụng DICOM trao đổi dữ liệu qua mạng thì cần sử dụng dịch vụ tương ứng, chương trình cung cấp một dịch vụ DICOM cụ thể gọi là Service Provider. Ứng dụng DICOM trao đổi dữ liệu với Service Provider để

lấy thông tin hay yêu cầu thực hiện một công việc cụ thể. Service Provider có thể tự thực hiện yêu cầu từ ứng dụng DICOM hay gửi yêu cầu cho một Service Provider khác, lúc đó Service Provider gửi yêu cầu đóng vai trò là một ứng dụng DICOM bình thường.

- Chuẩn DICOM đặc tả giao tiếp mạng thông qua 2 lớp dịch vụ:
  - Dịch vụ DIMSE và Association: ứng dụng DICOM trao đổi dữ liệu trực tiếp với lớp này.
  - Dịch vụ Upper Layer.

#### **2.2.2.2.1. Dịch vụ DIMSE**

- Dịch vụ DIMSE hỗ trợ 2 loại dịch vụ:
  - Dịch vụ loại Notification: cho phép ứng dụng DICOM thông báo cho ứng dụng khác biết về một sự kiện hay sự thay đổi trạng thái.
  - Dịch vụ loại Operation: cho phép ứng dụng DICOM yêu cầu ứng dụng DICOM khác thực hiện một công việc trên đối tượng SOP mà ứng dụng này đang quản lý.
- Dịch vụ DIMSE được chia làm 2 nhóm:
  - Dịch vụ DIMSE-N: dịch vụ này chỉ thao tác trên đối tượng Normalized SOP.
  - Dịch vụ DIMSE-C: dịch vụ này chỉ thao tác trên đối tượng Composite SOP.
- Các dịch vụ DIMSE:

Dịch vụ	Nhóm	Loại dịch vụ
C-STORE	DIMSE-C	Operation
C-GET	DIMSE-C	Operation
C-MOVE	DIMSE-C	Operation
C-FIND	DIMSE-C	Operation
C-ECHO	DIMSE-C	Operation

N-EVENT-REPORT	DIMSE-N	Notification
N-GET	DIMSE-N	Operation
N-SET	DIMSE-N	Operation
N-ACTION	DIMSE-N	Operation
N-CREATE	DIMSE-N	Operation
N-DELETE	DIMSE-N	Operation

#### 2.2.2.2.2. Dịch vụ Association

- Trước khi dùng dịch vụ DIMSE để truyền tải dữ liệu, ứng dụng DICOM cần được cung cấp thông tin ban đầu như Transfer Syntax dùng trong lúc truyền, tên ứng dụng DICOM sẽ giao tiếp, ... Những thông tin này được cung cấp qua dịch vụ Association. Dịch vụ này sẽ cung cấp các thông tin cần thiết trước khi truyền dữ liệu. Một Association giữa ứng dụng DICOM sẽ giúp 2 bên biết các thông tin ban đầu trước khi truyền dữ liệu. Khi truyền dữ liệu đi, cả bên truyền và bên nhận đều cung cấp Application Association Information trong request primitive và response primitive.

- Dịch vụ Association đi cùng với dịch vụ DIMSE là dịch vụ ở mức tổng quát so với các dịch vụ Association do Upper Layer cung cấp. Tại mức này dịch vụ Association sử dụng dịch vụ A-ASSOCIATE của Upper Layer.

- Dịch vụ Association sẽ tạo một association cho 2 ứng dụng DICOM để bắt đầu sử dụng các dịch vụ DIMSE.

- Các thông tin dịch vụ Association cần phải có:
  - Application context.
  - Các yêu cầu về presentation và session.
  - Thông tin về ứng dụng DICOM sử dụng dịch vụ.
  - Application Association Information.

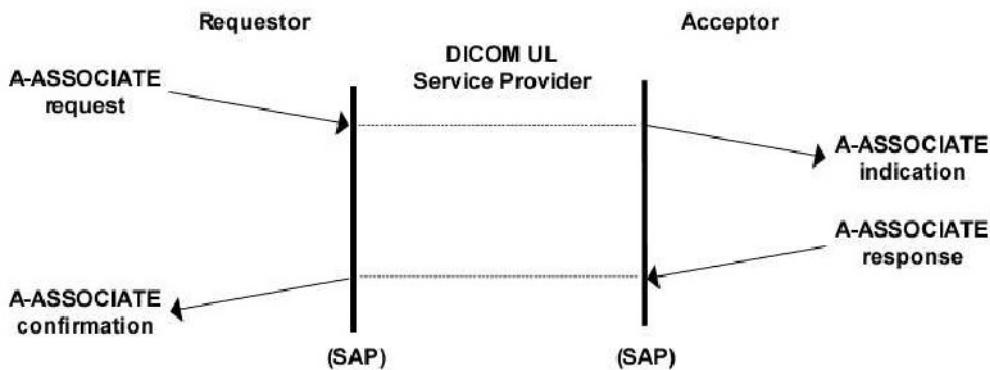
### 2.2.2.3. Dịch vụ Upper Layer

- Các dịch vụ Upper Layer được sử dụng bởi 2 dịch vụ ở mức trên là Association và DIMSE. Upper Layer chịu trách nhiệm đưa thông tin từ những dịch vụ trên thành các chuỗi byte để truyền qua mạng và nhận chuỗi byte từ mạng, sau đó đóng gói thành thông tin truyền cho dịch vụ trên.

- Các dịch vụ Upper Layer cung cấp:

#### A-ASSOCIATE

Thiết lập một association giữa hai ứng dụng DICOM thông qua các message A-ASSOCIATE request, A-ASSOCIATE indication, A-ASSOCIATE response và A-ASSOCIATE confirmation.

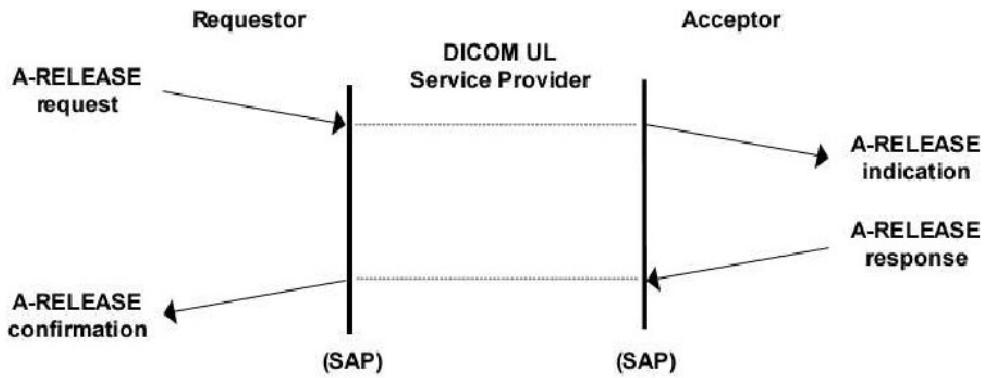


*Hình minh họa thiết lập association giữa 2 ứng dụng DICOM*

#### A-RELEASE

Khi một trong 2 bên muốn hủy association thì sẽ dùng dịch vụ này để hủy bỏ association giữa hai ứng dụng DICOM thông qua các message A-RELEASE request, A-RELEASE indication, A-RELEASE response và A-RELEASE confirmation.

Cả hai ứng dụng DICOM đều chấp nhận hủy bỏ association để giải phóng tài nguyên.

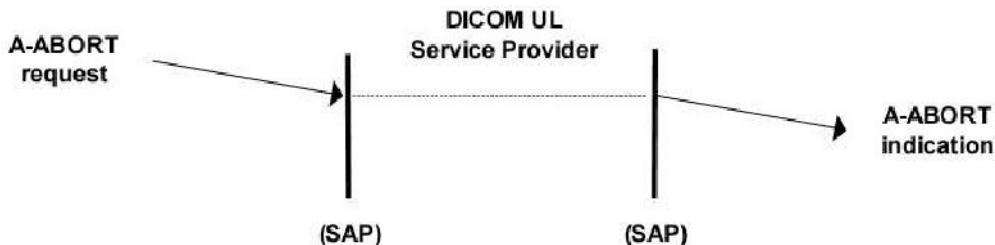


*Hình minh họa hủy bỏ association giữa 2 ứng dụng DICOM*

### A-ABORT

Ứng dụng DICOM cần ngắt đột ngột association với phía bên kia. Dịch vụ này không cần phải xác nhận lại kết quả thực hiện.

Tuy nhiên, yêu cầu indication từ ứng dụng DICOM không đảm bảo là sẽ đến với ứng dụng kia. Trong những trường hợp như vậy, cả hai ứng dụng đều biết rằng association đã bị ngắt. Việc ngắt được thực hiện thông qua các message A-ABORT request và A-ABORT indication.

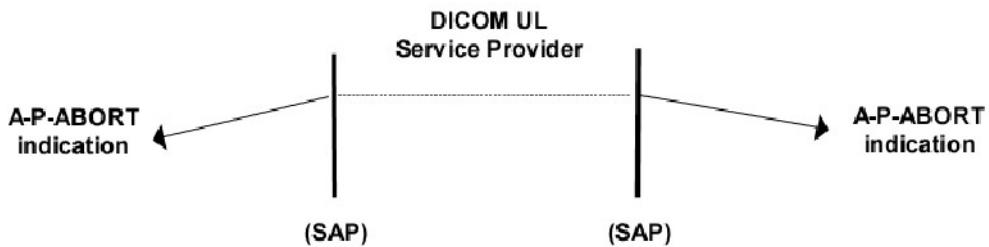


*Hình minh họa ngắt đột ngột association giữa 2 ứng dụng DICOM*

### A-P-ABORT

Service Provider phát tín hiệu ngắt association ngay mà không đợi một trong hai ứng dụng DICOM yêu cầu ngắt. Lý do của việc ngắt là do có các dịch vụ khác gặp trực trặc ở lớp Presentation hay lớp trên.

Việc ngắt đột ngột sẽ gây mất thông tin đang truyền.



*Hình minh họa ngắt association với yêu cầu ngắt từ Service Provider*

### P-DATA

Ứng dụng DICOM dùng dịch vụ này để trao đổi thông tin với nhau (truyền tải DICOM Message). Một association cho phép truyền và nhận P-DATA request primitive, P-DATA indication primitive đồng thời.

Dịch vụ DIMSE được sử dụng trong dịch vụ này.



*Hình minh họa truyền tải dữ liệu dựa trên association đã thiết lập giữa 2 ứng dụng*

- Các dịch vụ Upper Layer dùng giao thức TCP và truyền / nhận dữ liệu tại port 104 (là port chuẩn cho giao thức DICOM).
- Định dạng của một đơn vị thông tin giao tiếp giữa 2 peer trong giao thức Upper Layer là PDU (Protocol Data Unit). PDU được tạo từ các field có kích thước cố định và các field tùy chọn, những field tùy chọn sẽ chứa một hay nhiều item hay sub-item.
- Có 7 loại PDU trong giao thức DICOM Upper Layer:
  - A-ASSOCIATE-RQ PDU.
  - A-ASSOCIATE-AC PDU.
  - A-ASSOCIATE-RJ PDU.

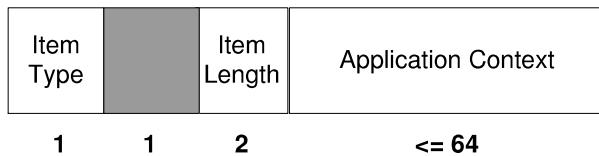
- P-DATA-TF PDU.
  - A-RELEASE-RQ PDU.
  - A-RELEASE-RP PDU.
  - A-ABORT PDU.
- Chỉ có header của PDU là có thứ tự byte Big Endian còn định dạng fragment của PDV (Presentation Data Values) message trong P-DATA-TF PDU là tuân theo giá trị của Transfer Syntax.
- Định dạng của PDU có đặc tả như sau:
    - Kiểu của PDU do một hay nhiều byte chỉ định với byte đầu tiên có số thứ tự thấp nhất.
    - Mỗi byte trong PDU có 8 bit đánh số từ 0-7 với bit 0 là bit có trọng số thấp.
    - Những byte liên tục dùng biểu diễn số nhị phân, byte có số thứ tự thấp thì có trọng số lớn.
    - Byte có số thứ tự thấp nhất sẽ được truyền đầu tiên trong luồng truyền dữ liệu.
  - Sau đây là cấu trúc của các PDU, độ lớn mỗi field tính theo byte, các ô màu đậm là dùng để dự trù.

### **PDU A-ASSOCIATE-RQ và PDU A-ASSOCIATE-AC**

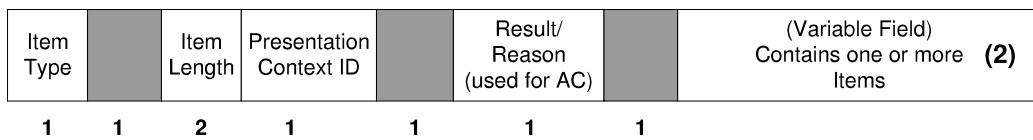
PDU Type		PDU Length	Protocol Version		Called Entity Title	Calling Entity Title		(Variable Field) Contains one or more (1) Items
1	1	4	2	2	16	16	32	

### *PDU A-ASSOCIATE-RQ và PDU A-ASSOCIATE-AC*

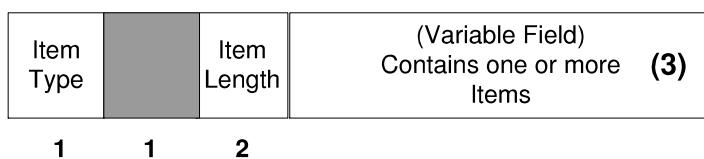
- Variable Field (1) gồm các Item con sau:
  - Application Context Item: chỉ một item.
  - Presentation Context Item: một hay nhiều item.
  - User Info Item: chỉ một item.



### *Application Context Item*

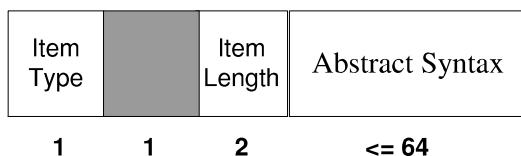


### *Presentation Context Item*

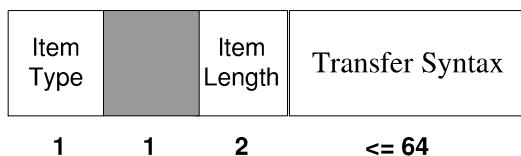


### *User Info Item*

- Variable Field (2) gồm các Item con sau:
  - Abstract Syntax Item: chỉ một item trên mỗi RQ, không có item này trong AC.
  - Transfer Syntax Item: một hay nhiều item trong RQ, chỉ một item trong AC.



### *Abstract Syntax Item*



### *Transfer Syntax Item*

- Variable Field (3) gồm các Item con sau:
  - Maximum Length Item.

Item Type		Item Length	Max Length Received
1	1	2	4

*Maximum Length Item*

**PDU A-ASSOCIATE-RJ PDU, PDU A-RELEASE-RQ, PDU A-RELEASE-RP PDU và PDU A-ABORT**

PDU Type		PDU Length		Result *	Source *	Reason / Diag *
1	1	4	1	1	1	1

*PDU A-ASSOCIATE-RJ PDU, PDU A-RELEASE-RQ*

*PDU A-RELEASE-RP PDU và PDU A-ABORT*

- (\*): tùy thuộc vào PDU cụ thể mà field này sẽ dùng hay để dự trữ.

**P-DATA-TF PDU**

PDU Type		PDU Length	(Variable Field) Contains one or more Presentation Data Value Items
1	1	4	

*P-DATA-TF PDU*

- Variable Field (4) là các DICOM Message.

Item Length	Presentation Context ID	Presentation-data-Values (PDV) DICOM message Command or Data Set Information
4	1	

*Presentation Data Value item*

## **Chương 3: GIỚI THIỆU VỀ NGÔN NGỮ LẬP TRÌNH C#**

### **3.1. Lịch sử ra đời của ngôn ngữ lập trình C#**

C# là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft, là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

### **3.2. Đặc điểm của ngôn ngữ lập trình C#**

C#, theo một hướng nào đó, là ngôn ngữ lập trình phản ánh trực tiếp nhất đến .NET Framework mà tất cả các chương trình .NET chạy, và nó phụ thuộc mạnh mẽ vào Framework này. Mọi dữ liệu cơ sở đều là đối tượng, được cấp phát và hủy bỏ bởi trình dọn rác Garbage-Collector (GC), và nhiều kiểu trùu tượng khác chẳng hạn như class, delegate, interface, exception, v.v, phản ánh rõ ràng những đặc trưng của .NET runtime.

- Trong ngôn ngữ C# mọi thứ liên quan đến khai báo lớp điều được tìm thấy trong phần khai báo của nó.
- Định nghĩa một lớp trong ngôn ngữ C# không đòi hỏi phải chia ra tập tin header và tập tin nguồn giống như trong ngôn ngữ C++.
- Hơn thế nữa, ngôn ngữ C# hỗ trợ kiểu XML, cho phép chèn các tag XML để phát sinh tự động các document cho lớp.
- Ngôn ngữ C# khá đơn giản, chỉ khoảng 80 từ khóa và hơn mươi mấy kiểu dữ liệu được xây dựng sẵn.
- Tuy nhiên, ngôn ngữ C# có ý nghĩa cao khi nó thực thi những khái niệm lập trình hiện đại. C# bao gồm tất cả những hỗ trợ cho cấu trúc, thành phần component, lập trình hướng đối tượng.
- Những tính chất đó hiện diện trong một ngôn ngữ lập trình hiện đại.
- C# cũng hỗ trợ giao diện interface, nó được xem như một cam kết với một lớp cho những dịch vụ mà giao diện quy định.

- Trong ngôn ngữ C#, một lớp chỉ có thể kế thừa từ duy nhất một lớp cha, tức là không cho đa kế thừa như trong ngôn ngữ C++, tuy nhiên một lớp có thể thực thi nhiều giao diện.
- Khi một lớp thực thi một giao diện thì nó sẽ hứa là nó sẽ cung cấp chức năng thực thi giao diện.
- Ngôn ngữ C# cũng hỗ trợ việc truy cập bộ nhớ trực tiếp sử dụng kiểu con trỏ của C++ và từ khóa cho dấu ngoặc [] trong toán tử.
- Các mã nguồn này là không an toàn (unsafe). Và bộ giải phóng bộ nhớ tự động của CLR sẽ không thực hiện việc giải phóng những đối tượng được tham chiếu bằng sử dụng con trỏ cho đến khi chúng được giải phóng.

### 3.3. Bảng từ khóa C#

<code>abstract</code>	<code>default</code>	<code>foreach</code>	<code>object</code>	<code>sizeof</code>	<code>unsafe</code>
<code>as</code>	<code>delegate</code>	<code>goto</code>	<code>operator</code>	<code>stackalloc</code>	<code>ushort</code>
<code>base</code>	<code>do</code>	<code>if</code>	<code>out</code>	<code>static</code>	<code>using</code>
<code>bool</code>	<code>double</code>	<code>implicit</code>	<code>override</code>	<code>string</code>	<code>virtual</code>
<code>break</code>	<code>else</code>	<code>in</code>	<code>params</code>	<code>struct</code>	<code>volatile</code>
<code>byte</code>	<code>enum</code>	<code>int</code>	<code>private</code>	<code>switch</code>	<code>void</code>
<code>case</code>	<code>event</code>	<code>interface</code>	<code>protected</code>	<code>this</code>	<code>while</code>
<code>catch</code>	<code>explicit</code>	<code>internal</code>	<code>public</code>	<code>throw</code>	
<code>char</code>	<code>extern</code>	<code>is</code>	<code>readonly</code>	<code>true</code>	
<code>checked</code>	<code>false</code>	<code>lock</code>	<code>ref</code>	<code>try</code>	
<code>class</code>	<code>finally</code>	<code>long</code>	<code>return</code>	<code>typeof</code>	
<code>const</code>	<code>fixed</code>	<code>namespace</code>	<code>sbyte</code>	<code>uint</code>	
<code>continue</code>	<code>float</code>	<code>new</code>	<code>sealed</code>	<code>ulong</code>	
<code>decimal</code>	<code>for</code>	<code>null</code>	<code>short</code>	<code>unchecked</code>	

### 3.4. Công nghệ .NET

#### 3.4.1. Giới thiệu

- Đầu năm 1998, sau khi hoàn tất phiên bản 4 của Internet Information Server (IIS), một đội lập trình ở Microsoft nhận thấy họ còn rất nhiều sáng kiến để kiện toàn IIS. Họ bắt đầu thiết kế một architecture mới dựa trên những ý đó và project được đặt tên là Next Generation Windows Services (NGWS).

- Sau khi Visual Basic 6 được trình làng vào cuối năm 1998, dự án kế tiếp mang tên Visual Studio 7 được sát nhập vào NGWS. Đội ngũ COM+/MTS góp vào một môi trường thống nhất cho tất cả các ngôn ngữ lập trình trong Visual Studio, họ có ý định cho ngay cả các ngôn ngữ lập trình của công ty khác dùng.

- Công tác này được giữ bí mật mãi đến hội nghị Professional Developers' Conference ở Orlando vào tháng 7/2000. Đến tháng 11/2000 thì Microsoft cho phát hành Beta 1 của .NET gồm ba CD. Tính đến lúc ấy thì Microsoft đã làm việc trên dự án ấy gần ba năm rồi. Điều ấy cắt nghĩa tại sao Beta 1 version tương đối rất vững chãi.

- .NET mang dấu tích những sáng kiến đã được áp dụng trước đây như **p-code** trong UCSD Pascal cho đến **Java Virtual Machine**. Có điều Microsoft góp nhặt những sáng kiến của người khác, kết hợp với những sáng kiến của chính mình để làm nên một sản phẩm ăn rơ từ trong ra ngoài. Có lẽ cuối năm 2001 hay đầu năm 2002 Microsoft mới phát hành .NET. Có người hỏi Microsoft xem .NET quan trọng như thế nào. Các lãnh đạo của Microsoft cho biết 80% từ khóa Research & Development (Nghiên cứu và Triển khai) của Microsoft trong năm 2001 được dành cho .NET, tất cả sản phẩm của Microsoft đều sẽ được dọn nhà qua .NET platform.

- .NET là vũ khí chiến lược của Microsoft để trong tương lai thâu chiém lĩnh môi trường Desktop, Distributed cho đến Internet và Mobile (Phone, Pocket PC). Visual Studio.NET cho ta một IDE (**I**ntergrated

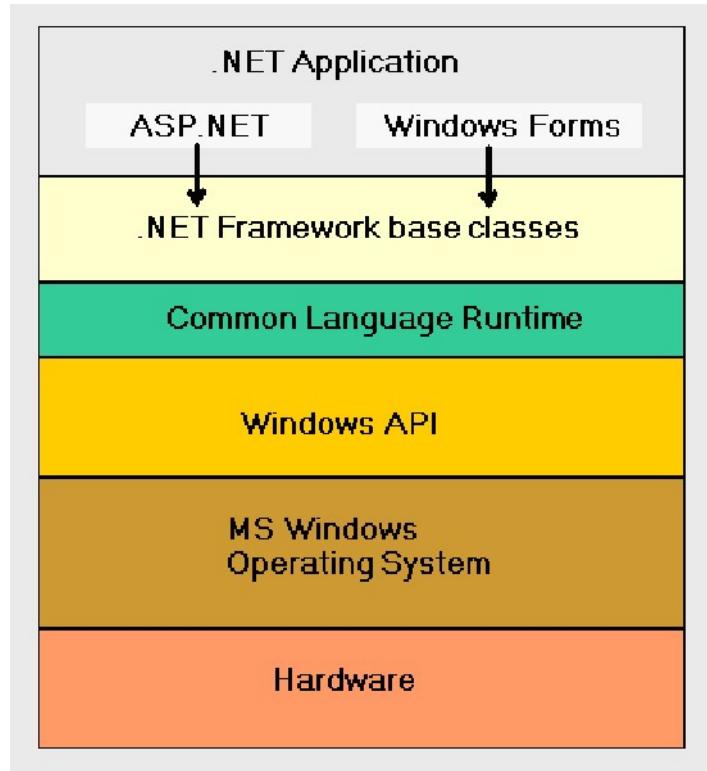
**Development Environment**) tuyệt diệu, đầy đủ để triển khai mọi loại dự án. Cốt lõi của .NET là .NET Framework, hỗ trợ lập trình theo hướng đối tượng (Object Oriented) cho VB.NET (Visual Basic 7) và C#. Hai ngôn ngữ lập trình này khá đơn giản (chỉ có chừng 80 reserved words), tương đương nhau về chức năng và code của hai bên có thể thura kế lẫn nhau. .NET Framework cung cấp khoảng 5000 classes để hỗ trợ mọi nhu cầu lập trình như Streaming, Threading, Collections, Delegate và EventHandling, Interface, Remoting, Reflection, Unicode, XML, Disconnected database ADO.NET, Encryption.

.v.v..

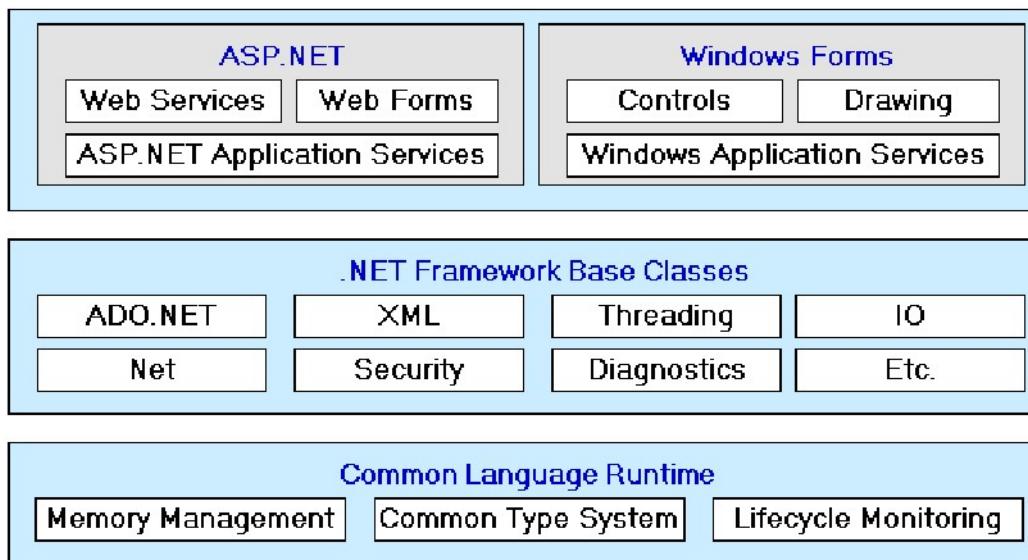
### **3.4.2. .NET FRAMEWORK**

- .NET được phát triển từ đầu năm 1998, lúc đầu có tên là Next Generation Windows Services (NGWS). Nó được thiết kế hoàn toàn từ con số không để dùng cho Internet. Viễn tưởng của Microsoft là xây dựng một hệ thống phân tán toàn cầu, dùng **XML** (chứa những databases tí hon) làm chất keo để kết hợp chức năng của những computers khác nhau trong cùng một tổ chức hay trên khắp thế giới.

- Những computers này có thể là Servers, Desktop, Notebook hay Pocket Computers, đều có thể chạy cùng một software dựa trên một platform duy nhất, độc lập với hardware và ngôn ngữ lập trình. Đó là .NET Framework. Nó sẽ trở thành một phần của MS Windows và sẽ được port qua các platform khác, có thể ngay cả Unix.



Hình 6: Mô hình một ứng dụng .NET



Hình 7: Các thành phần chính của .NET Framework

- .NET application được chia ra làm hai loại: cho Internet gọi là **ASP.NET**, gồm có **Web Forms**, **Web Services** và cho desktop gọi là **Windows Forms**.
  - Windows Forms giống như Forms của VB6. Nó hỗ trợ **Unicode** hoàn toàn, rất tiện cho chữ Việt và thật sự mang tính hướng đối tượng.
  - Web Forms có những **Server Controls** làm việc giống như các Controls trong Windows Forms, nhất là có thể dùng codes để xử lý Events như của Windows Forms.
    - Điểm khác biệt chính giữa ASP (Active Server Pages) và ASP.NET là trong ASP.NET, phần đại diện visual components và code nằm riêng nhau. Ngoài ra ASP.NET hoàn toàn có tính hướng đối tượng.

## **Chương 4: CÀI ĐẶT VÀ THỬ NGHIỆM CHƯƠNG TRÌNH**

### **4.1. Môi trường cài đặt**

Hệ điều hành Window XP/ Vista 7 có cài đặt phần mềm hỗ trợ Microsoft .Net Framework 3.5.

Ngôn ngữ lập trình C#.

Máy tính pentum IV, Ram 512MB.

### **4.2. Tập ảnh thử nghiệm**

Ảnh được tải về từ trang web: <http://www.barre.nom.fr/medical/samples/>



anh1.png



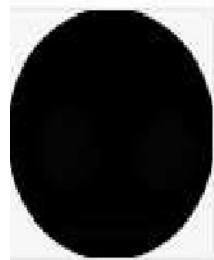
anh4.png



anh5.png



anh2.png



anh3.png



anh6.png



anh7.png

## 4.3. Giao diện của chương trình

### 4.3.1. Giao diện chính của chương trình

Cho phép người dùng thao tác với nút lệnh của chương trình gồm có các chức năng chính sau:

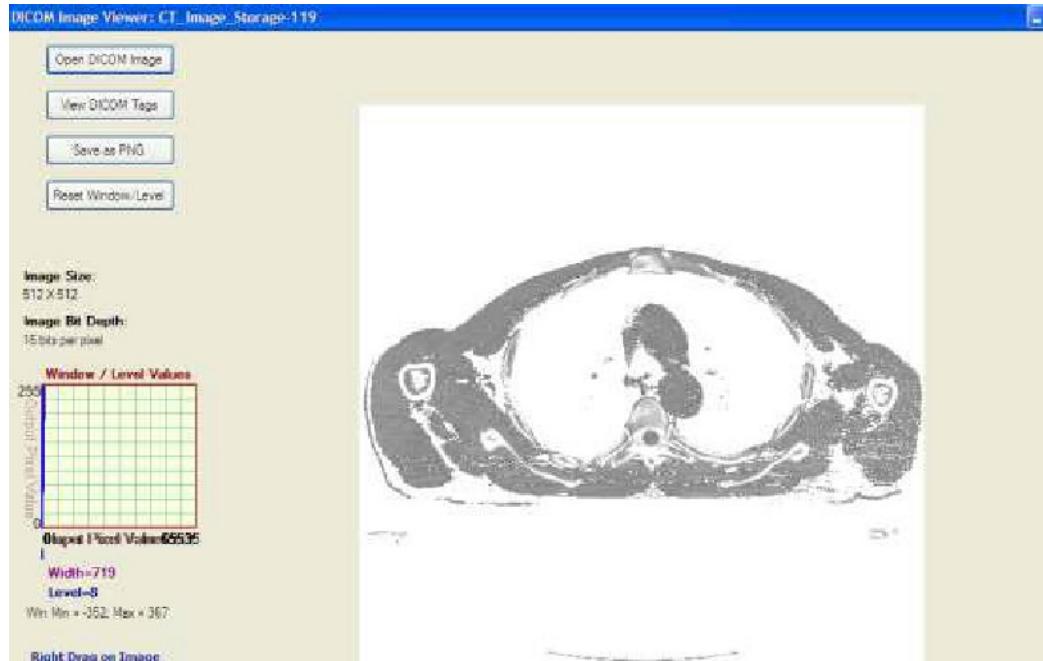
- **Open DICOM Image:** mở thư mục chứa ảnh DICOM, lựa chọn ảnh cần xem.
- **View DICOM Tags:** hiển thị các thông tin có trong ảnh DICOM.
- **Save as PNG:** lưu ảnh DICOM sang định dạng PNG.
- **Reset Window / Level:** thiết lập lại trạng thái cửa sổ.



Hình 8: Giao diện chính của chương trình

#### 4.3.2. Giao diện hiển thị ảnh DICOM

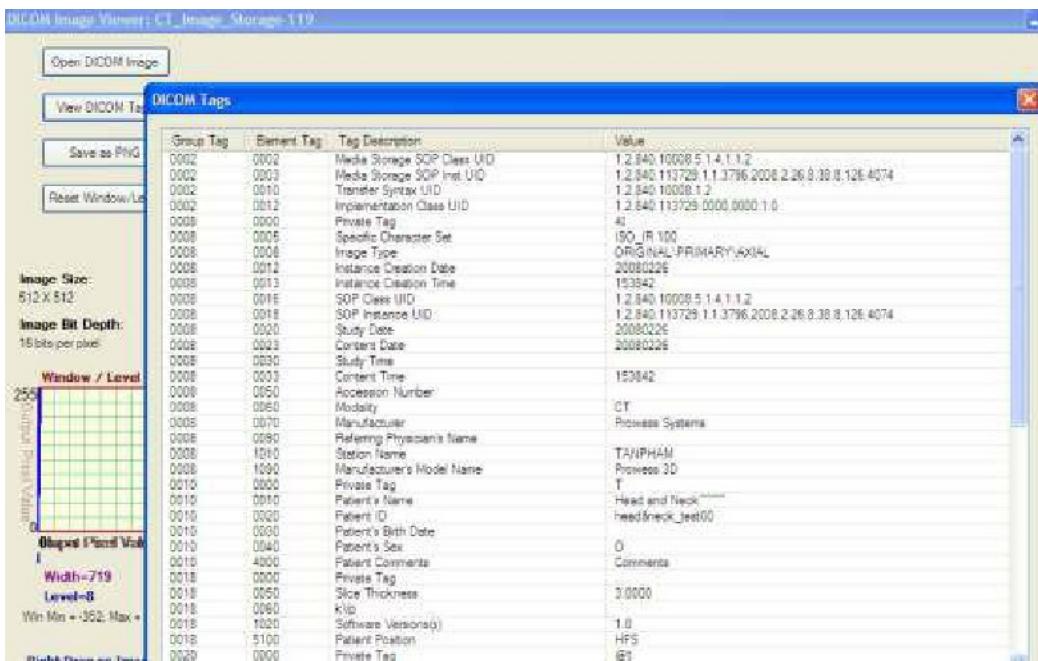
Khi click vào nút lệnh **Open DICOM Image** sẽ làm xuất hiện thư mục chứa ảnh DICOM, lựa chọn ảnh cần hiển thị như ví dụ bên dưới.



Hình 9: Giao diện hiển thị ảnh DICOM

### 4.3.3. Giao diện hiển thị thông số của ảnh DICOM

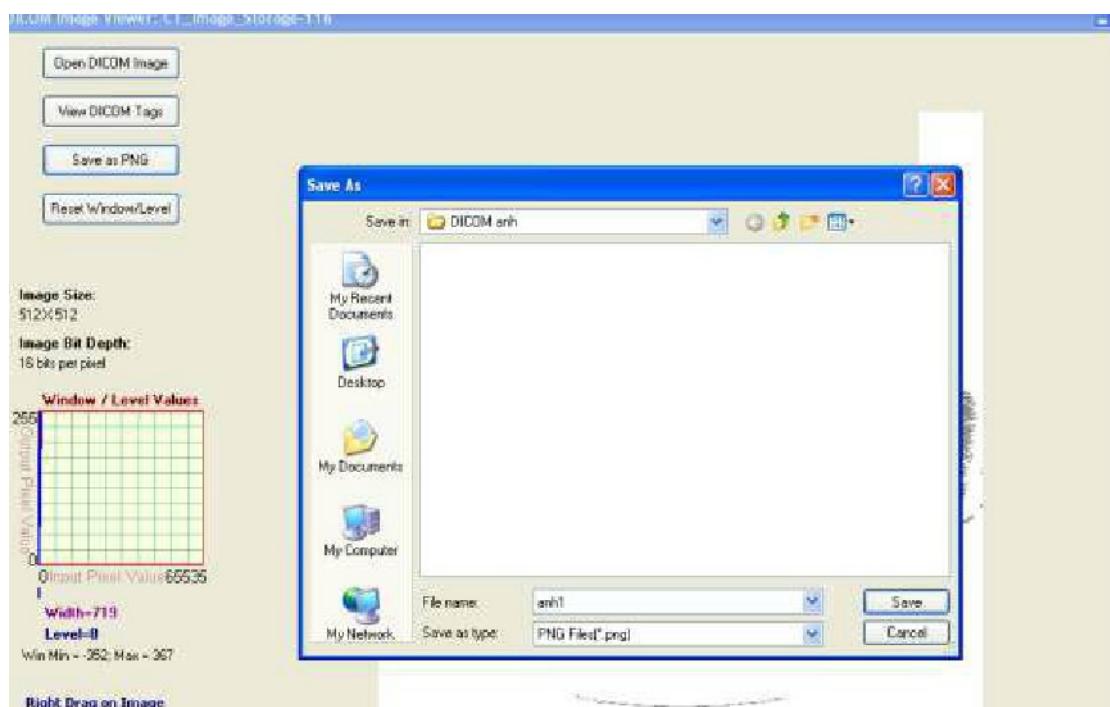
Sau khi ảnh DICOM được hiển thị, click vào nút lệnh **View DICOM Tags** ta thấy một bảng DICOM Tags như ở hình 10 cho phép người dùng biết được các thông số có trong ảnh như tên của bệnh nhân, mã ID của bệnh nhân, giới tính.....



Hình 10: Giao diện hiển thị thông số của ảnh DICOM

#### 4.3.4. Giao diện lưu ảnh sang định dạng PNG

Click vào nút lệnh **Save as PNG** làm xuất hiện một cửa sổ như ở hình 11, đặt tên cho ảnh cần lưu, click vào nút Save để lưu ảnh DICOM sang định dạng PNG. Với định dạng ảnh PNG sẽ giúp cho các phần mềm xử lý ảnh như photoshop có thể đọc ảnh và chỉnh sửa ảnh dễ dàng.



Hình 11: Giao diện lưu ảnh sang định dạng PNG

## KẾT LUẬN

Ngày nay DICOM là định dạng ảnh được dùng phổ biến nhất trong lĩnh vực y tế. Do nhu cầu khám chữa bệnh của người dân ngày càng tăng việc quản lý bằng giấy tờ, sổ sách là không thể đáp ứng được đòi hỏi phải ứng dụng công nghệ vào quá trình quản lý cũng như nhiều khâu khác của quá trình chuẩn đoán bệnh. Với việc ứng dụng ảnh DICOM trong y khoa đã giải quyết một cách hiệu quả các vấn đề trên.

Trong bài đồ án này em đã nêu những nét đặc trưng nhất về chuẩn lưu trữ ảnh DICOM như lịch sử ra đời, các khái niệm cơ bản, cấu trúc file, giao thức DICOM... Đồ án cũng đã xây dựng được chương trình đọc ảnh DICOM. Việc cài đặt thử nghiệm chương trình trên bằng ngôn ngữ C# đã được thực hiện, bước đầu cho kết quả tốt.

Tuy nhiên do hạn chế về điều kiện và thời gian và kiến thức, đồ án sẽ không thể tránh khỏi những thiếu sót. Kính mong được sự đóng góp ý kiến của thầy cô và các bạn, để em có thể hoàn thiện tốt hơn để tài nghiên cứu của mình.

## TÀI LIỆU THAM KHẢO

1. " *Digital Imaging and Communications in Medicine (DICOM)*" , In National Electrical Manufacturers Association, USA, 2007.
2. Nguyễn Thị Ngọc, " *Phân cụm dữ liệu mật đố*" , Báo cáo thực tập tốt nghiệp ngành Công nghệ Thông tin, Trường ĐHDL Hải Phòng, 2008.
3. DICOM trên Wikipedia tại  
[http://en.wikipedia.org/wiki/Digital\\_Imaging\\_and\\_Communications\\_in\\_Medicine](http://en.wikipedia.org/wiki/Digital_Imaging_and_Communications_in_Medicine)
4. <http://tailieu.vn/>
5. Tài liệu về chuẩn DICOM (phiên bản 3.0) tại  
<http://medical.nema.org/dicom/2006/>