Tips for Analysis of Clinical Images

Steve Pieper, PhD
Isomics, Inc.

Goal

What to do with image data pulled from the PACS

- The nature of medical images
- Some words about DICOM
- Software suggestions

Research != Clinical

Research	Clinical
Common acquisition protocol	Customized per-patient
Controlled context and data curation	Wide variability
Highest reasonable image resolution	Best clinically justified images
Coverage of target structures	Only what is needed to answer the question at hand
Acquisition often custom for research software	Acquisition paired with commercial workstation features
Often normal subjects	Almost never normal (pathology and other variability)
Quantitative	Qualitative

Keep in Mind

- Know the clinical acquisition scenario
  - If data isn't needed for a defined clinical purpose, it's not likely to exist
  - Some critical data isn’t explicitly recorded
  - If it’s needed for billing, it’s likely to be recorded
- Patient care can be hectic and imperfectly organized

DICOM

- The ubiquitous standard for **di**gital **i**mage **co**mmunications in **m**edicine
  - http://dicom.nema.org
- Can be thought of as a log file recording what the scanner was doing during an acquisition
- Highly complex even when done right
- Even more complex because it is almost always done wrong
- Better than any known alternative

(Some of) What you need to know about DICOM

- Each “DICOM file” is really an **instance** of a **class**, very much like a C++ or Java class

- Each **class*** has a **unique identifier** (SOP Class UID)

- Each instance has a **unique identifier** (SOP Instance UID)

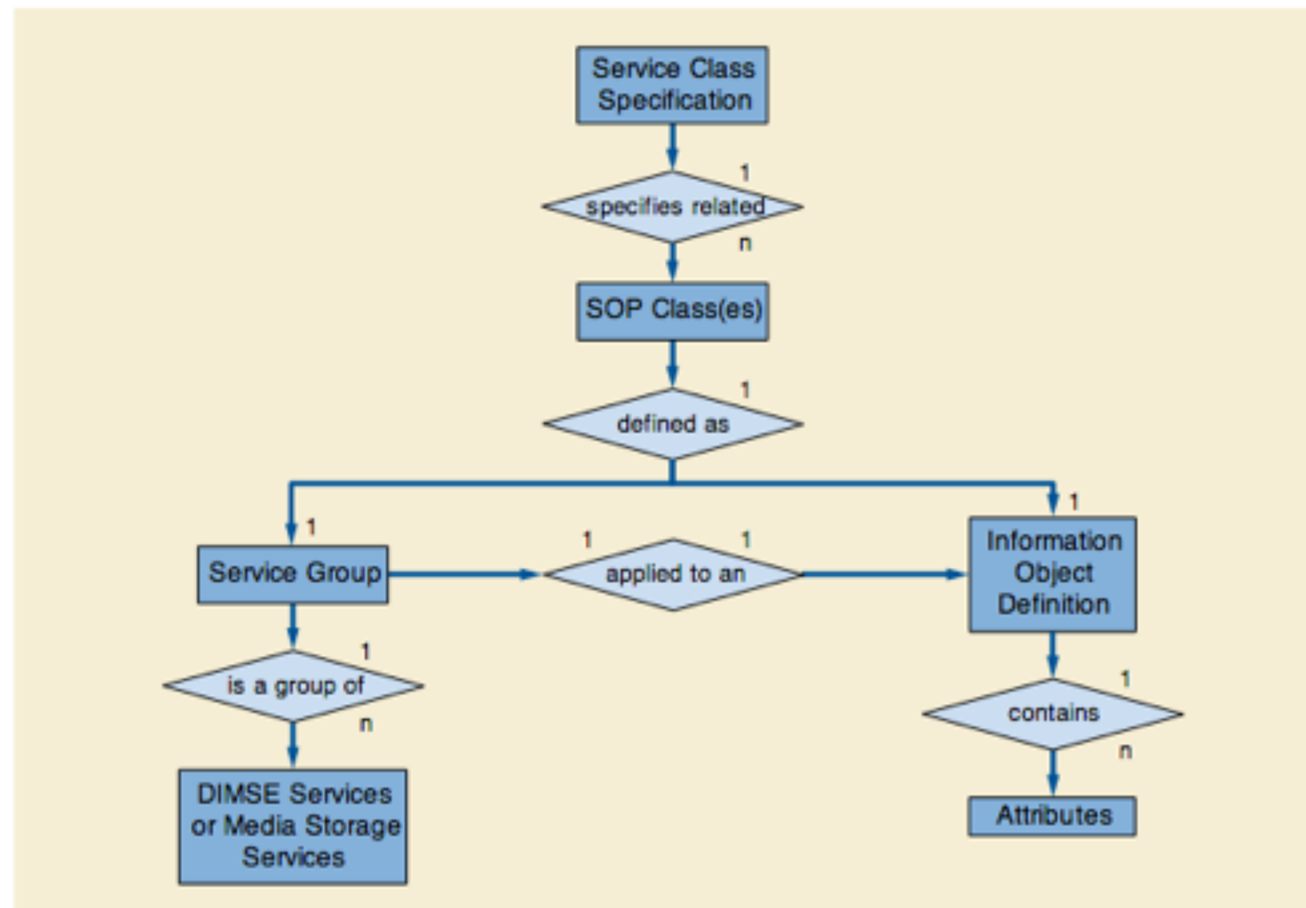
*called a “service-object pair class” or “SOP Class” in DICOM

DICOM MR Example

- File name of data exported from mi2b2:
  - **<MRN>/<YYYY-MM-DD>_<Age>Y_<Study Description>/pre__Ax_T1_2213/<InstanceUID>.dcm**
- Contains an instance of 1.2.840.10008.5.1.4.1.1.4, which is MRImageStorage SOPClassUID
- You can look at the standard and search the web to find information.

Service-Object Pairs

- An MRImageStorage Service-Object Pair (SOP) Class has two parts
  - The **Service**, in this case storing to media
  - The **Information Object Definition**, which is like the class declaration in programming



What you find in the standard

Table A.4-1. MR Image IOD Modules

IE	Module	Reference	Usage
Patient	Patient	C.7.1.1	M
	Clinical Trial Subject	C.7.1.3	U
Study	General Study	C.7.2.1	M
	Patient Study	C.7.2.2	U
	Clinical Trial Study	C.7.2.3	U
Series	General Series	C.7.3.1	M
	Clinical Trial Series	C.7.3.2	U
Frame of Reference	Frame of Reference	C.7.4.1	M
Equipment	General Equipment	C.7.5.1	M
Image	General Image	C.7.6.1	M
	Image Plane	C.7.6.2	M
	Image Pixel	C.7.6.3	M
	Contrast/Bolus	C.7.6.4	C - Required if contrast media was used in this image
	Device	C.7.6.12	U
	Specimen	C.7.6.22	U
	MR Image	C.8.3.1	M
	Overlay Plane	C.9.2	U
	VOI LUT	C.11.2	U
	SOP Common	C.12.1	M
	Common Instance Reference	C.12.2	U

- http://medical.nema.org/medical/dicom/2014a/output/html/part03.html#sect_A.4

What you find in the standard

Table C.7-10. Image Plane Module Attributes

Attribute Name	Tag	Type	Attribute Description
Pixel Spacing	(0028,0030)	1	
Image Orientation (Patient)	(0020,0037)	1	
Image Position (Patient)	(0020,0032)	1	
Slice Thickness	(0018,0050)	2	
Slice Location	(0020,1041)	3	

Equation C.7.6.2.1-1.

$$\begin{bmatrix} P_x \\ P_y \\ P_x \\ 1 \end{bmatrix} = \begin{bmatrix} X_x \Delta_i & Y_x \Delta_j & 0 & S_x \\ X_y \Delta_i & Y_y \Delta_j & 0 & S_y \\ X_z \Delta_i & Y_z \Delta_j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 0 \\ 1 \end{bmatrix} = M \begin{bmatrix} i \\ j \\ 0 \\ 1 \end{bmatrix}$$

Where:

P_{xyz} The coordinates of the voxel (i,j) in the frame's image plane in units of mm.

S_{xyz} The three values of Image Position (Patient) (0020,0032). It is the location in mm from the origin of the RCS.

X_{xyz} The values from the row (X) direction cosine of Image Orientation (Patient) (0020,0037).

Y_{xyz} The values from the column (Y) direction cosine of Image Orientation (Patient) (0020,0037).

i Column index to the image plane. The first column is index zero.

Δ_i Column pixel resolution of Pixel Spacing (0028,0030) in units of mm.

j Row index to the image plane. The first row index is zero.

Δ_j Row pixel resolution of Pixel Spacing (0028,0030) in units of mm.

Additional constraints apply:

1. The row and column direction cosine vectors shall be orthogonal, i.e., their dot product shall be zero.
2. The row and column direction cosine vectors shall be normal, i.e., the dot product of each direction cosine vector with itself shall be unity.

- http://medical.nema.org/medical/dicom/2014a/output/html/part03.html#sect_C.7.6.2

DICOM Data Elements

- Each Data Element has a **Unique Tag**
    - Tag is two bytes defining: **Group** and **Element**
    - Odd numbered groups are for **Private Elements**, defined by the vendor and not the standard (approach with caution!)
- Data Elements have a defined **Value Representation (VR)** and **Value Multiplicity (VM)**
- E.g. (0020,0037) is “Image Orientation (Patient)”, VR of DS (Decimal String) and VM of 6 (two three-dimensional vectors)
- Standard Data Elements are defined in Part 6
    - http://medical.nema.org/medical/dicom/2014a/output/html/part06.html#chapter_6

DICOM Information Model

- Captures the way the real world is represented in DICOM
- Each class instance (DICOM file) can be interpreted in context

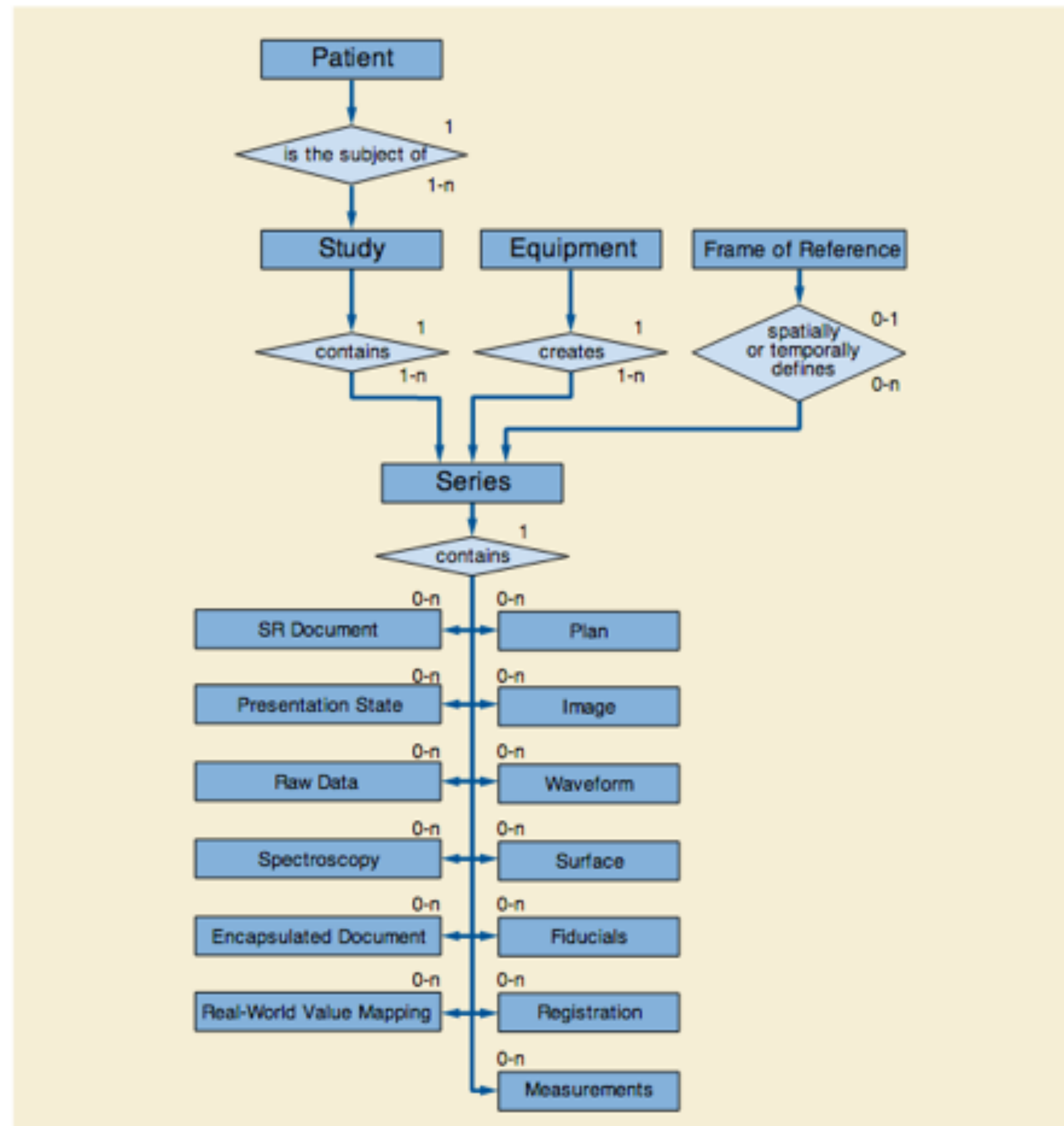


Figure A.1-1. DICOM Composite Instance IOD Information Model

How it works in the real world

- Often, understanding DICOM **is** the key to interpreting clinical images, but…
- Key data may be in undocumented private elements
- Vendors may only half-implement the standard
- The format can change without warning due to software upgrades

Other Confounds

- Images are not always “volumes” (irregular slice spacing; gantry tilt; oblique acquisitions…)
- Patient motion
- Machine failure or miscalibration
- Special case clinical acquisitions
- Variability from scanner to scanner, even in the same institution
- No doubt many more you will discover…

DICOM Goodies you won't find in the PACS (for a few years anyway)

- Newer "21st Century" DICOM Objects
  - Multi-frame images (one or more volumes in a single instance / file)
  - Enhanced MR, CT, PET… (better descriptions and more required elements)
- Spatial Registration
  - http://medical.nema.org/medical/dicom/2014a/output/html/part03.html#sect_A.39
- Segmentation
  - http://medical.nema.org/medical/dicom/2014a/output/html/part03.html#sect_A.51
- Surface Segmentation (Mesh)
  - http://medical.nema.org/medical/dicom/2014a/output/html/part03.html#sect_A.57
- Structured Reports
  - http://medical.nema.org/medical/dicom/2014a/output/html/part03.html#sect_A.35

What **can** you do?

- Visually review scans for findings that weren't made as part of clinical workup (easiest, tedious)
- Interactively measure structures (harder, more tedious)
- Run automated analysis (often not possible, still somewhat tedious due to manual QA requirements)

Low-Level DICOM Manipulation

- DCMTK (http://dicom.offis.de/)
  - BSD-style C++
  - Handles most of DICOM, including networking
  - Robust and widely used in research and industry
- dcm4che2 (http://www.dcm4che.org/)
  - Similar to DCMTK, but written in Java
  - Part of a more comprehensive suite (dcm4chee)
- PixelMed Java DICOM Toolkit (http://www.pixelmed.com/)
- pydicom (https://code.google.com/p/pydicom/)
- MATLAB (http://www.mathworks.com/company/newsletters/articles/accessing-data-in-dicom-files.html)

Processing Images

- Remember that Pixels are not Physical
    - Need to be mapped to patient space
    - Need to be converted into real-world units
    - Different acquisitions may be on different sampling grids
    - Some libraries handles this for you, others do not
- Some starting points for building software
    - The Insight Toolkit (http://itk.org/, C++, Image segmentation and registration)
    - scipy: numpy, ndimage… (http://scipy.org/, general image and numerical processing in python)
    - The Visualization Toolkit (http://vtk.org/, C++ or python, visualization and some general processing)
    - Many others…

Analysis Applications: Clinical Workstations

- First choice if you have well defined standard operation to perform
  - FDA clearance implies validated and accurate
  - Documented and supported
  - Often available after hours
- Caveats
  - Will often only perform clinical functions (not flexible)
  - May be hard to separate research work from patient record
  - May contain “Black Box” analysis tools that are not compatible with reproducible science
  - Can be expensive and unavailable
  - Can still be buggy in spite of FDA clearance

Analysis Applications: Free / Open Source

- Osirix (http://www.osirix-viewer.com/)
    - Capable and User Friendly Radiology Workstation
    - Mac only, optional paid upgrade for clinical version
- MIPAV (http://mipav.cit.nih.gov/)
    - Image analysis platform
    - Java (cross platform)
- ImageJ (http://imagej.nih.gov/ij/)
    - 2D Workhorse
    - Java (cross platform)
- Medical Imaging Interaction Workbench and Toolkit (MITK, http://www.mitk.org/MITK)
    - German Cancer Research Center
    - C++ (cross platform)

Analysis Applications: 3D Slicer (http://slicer.org/)

- Developed for over a decade by BWH and collaborators
- Completely free and open source (BSD-style license)
- C++ core based on VTK, ITK, and Qt available for Linux, Mac, and Windows
- Interactive DICOM, Segmentation, Registration, and Quantification
- Fully scriptable and extendable in python
- Dozens of core features and application-specific extensions
- End-to-end tutorials and documentation available

In Conclusion…

- Realize that clinical image data is complex and must be understood in context
- DICOM is unavoidable, and in the end that's a good thing
- Many software tools (free and paid) are available to make it possible to analyze these images, but all require significant time investment