Implementation Experience

Objects, Test Tool & Validation

David Clunie
PixelMed Publishing

Implementation Experience

- Test objects
  - Test images & spectroscopy objects
  - Creation from existing single frame bulk data
  - Synthesis of multi-frame & technique attributes
- Test tool
  - Annotation, grayscale pipeline & multi-frame navigation
  - Display of spectra
- Validation
  - Challenges of verifying complex nested conditions
  - Use of XSL-T and Xpath for definition/execution

Creating Test Objects

- Requirements specify range of attributes, functional groups
  - Minimal (barely compliant)
  - Comprehensive (to exercise most complex compliance)
- Could have used
  - Purely synthetic pixel data
  - Automatically generate many possible attribute sets
- Disadvantages
  - Really boring to look at (esp. for clinicians, physicists)
  - Would not demonstrate the application advantages of the new object
- Approach chosen
  - Realistic examples whenever possible
  - Achieves not only mechanical conformance with the standard
  - Ensures plausible and internally consistent values for attributes

Creating Test Objects

- Requirements were specified as sets of images
  - Based on which object features were in use
- Various clinical, research, demonstration and vendor supplied single frame data sets were examined to assess
  - Feasibility for building the new object attributes based on the old
  - Scenarios that exercised the various object features (e.g. color, real world values, rescale attributes)
  - Scenarios that were suitable to demonstrate the new application areas that were poorly supported in the old object (functional imaging, motion imaging, cardiac imaging)
- Where no source data was available
  - Likely sources harassed (vendors, colleagues, researchers)
  - Pixel data synthesized (e.g., McGill Brain Simulator on the web)

Creating Test Objects

- Given
  - a large collection of single frame old MR objects
  - how to make new enhanced multi-frame MR objects out of them ?
- Adapted crude tools (dccp and dcmulti from dicom3tools)to:
  - Modify while copying single frame source images to
    - Cleanse demographics and identifying attributes
    - Set dates and times appropriately
    - Clean up inappropriate or buggy old MR object attributes
    - Add appropriate new MR object attributes as necessary
  - Sort input images based on specified criteria
  - Collect all source attributes to be mapped and determine whether varying on a "per-frame" basis or not
  - Map old attributes into new (or synthesize them as necessary)
  - Group into functional groups and decide whether shared or per-frame
  - Add temporal position index and dimensions as requested
  - Concatenate pixel data from multiple files into one (7FE0,0010)

Lessons Learned

- Re-using the old objects is not as easy as it sounds
  - Old attributes poorly defined as to their meaning, or the meaning of defined terms
  - No standard old attributes to correspond to new attributes
  - Even with extensive use of private attributes as the source, still many gaps
  - Some consistently buggy attributes from various vendors were worked around by hand - not obvious if there is a robust general solution to some of these
- MR equipment vendors using this approach to bring forward legacy systems will find gaps in what is currently being stored internally
- Third-party vendors wanting to retrofit installed base will have trouble generating truly compliant new objects - will have to be fairly creative

Particular Difficulties

- Describing phase encoding steps
  - In-plane
  - Out-of-plane
  - Using Acquisition Matrix vs. Number of Phase Encoding Steps
  - Had to resort to private attributes a lot
  - Often had to override by hand with “likely” values anyway
- Magnetic Field Strength and Imaging Frequency
  - Just encoded incorrectly by many scanners
- Building Acquisition and Reference date and time and duration attributes
  - Difficult to know just what is meant by what is encoded in Image (Content) and Acquisition date and time attributes in old images
  - Overrode manually in many cases
- Contrast Bolus module contents
  - Truly hopeless in most old images, since operator entered free text (including “none”)
  - Overrode manually in many cases

More Straightforward Mapping

- Anatomical codes
  - Fairly easy when standard defined terms for Body Part Examined were used - direct mapping to SNOMED codes
  - Often not filled in by scanners
  - Manually override Body Part Examined in source and let tool map to coded equivalent
- Receive and Transmit coils
  - Generated automatic mapping to coded values based on commonly encountered strings found in various vendor’s source images
- Image Type, etc.
  - Mapping the old Image Type, Scanning Sequence, Sequence Variant, and Scan Options provided a good basis for initial values for many new technique related attributes
  - Still ended up overriding many manually to get more realistic values

Creating Dimensions

- From a display perspective dimensions are great
  - Clear instructions on order in which to render
  - No understanding necessary of
    - meaning of attributes
    - their natural sort order
- Places greater burden on the object creator
  - Choice of which attributes (or entire functional groups) to use as dimensions
  - In which order to specify dimensions
  - For a particular dimension, how to sort that attribute (or functional group)
    - Easy for single-valued attributes with monotonically increasing values, like Temporal Position Index
    - String values - does DERIVED come before or after ORIGINAL ?
    - Multiple valued attributes - ignore or use which values ?
  - Probably considerably easier for an “application” generating images that “understands” what is intended, as opposed to a mechanical test or conversion tool

Implementation Experience

- Test objects
  - Test images & spectroscopy objects
  - Creation from existing single frame bulk data
  - Synthesis of multi-frame & technique attributes
- Test tool
  - Annotation, grayscale pipeline & multi-frame navigation
  - Display of spectra
- Validation
  - Challenges of verifying complex nested conditions
  - Use of XSL-T and Xpath for definition/execution

Test Tool - DicomImageViewer

- Goal is to meet requirements of evaluating and experimenting with new MR object, not to replicate complete workstation functionality
- Correspondingly simple - single image display panel with simple annotation and navigation features
- Emphasis is on highlighting new features of MR object - multi-frame characteristics and use of dimensions

Overview of Functionality

- Read images, spectroscopy objects, DICOMDIR from files
- Receive/send images/spectroscopy objects across network
- Query/retrieve images/spectroscopy objects across network
- Display multi-frame images/spectra in implicit and by dimension order
- Display values of common, shared and per-frame varying attributes
- Export objects into a readable (XML) form

Design Decisions - Platform

- Emphasize portability over performance
- 100% Pure Java
  - Portable across Windows, Linux, Solaris, Mac OS X, etc.
  - JRE 1.3.1 or greater
  - On a fast PC with adequate memory, sufficient to display multi-frame objects of 512x512 of several hundred MB in size
  - Takes advantage of Java internationalization - supports all DICOM character sets
  - Not yet ported to 1.4.1 or using Java Advanced Imaging or Java Image I/O - significant performance improvements expected - awaiting Mac OS X adoption of 1.4.1

Design Decisions - Toolkits

- Re-use of existing freely available Pure Java components:
  - Hypersonic SQL database
  - Sun XML pack
  - PixelMed Publishing DICOM toolkit (parsing, network, display code)

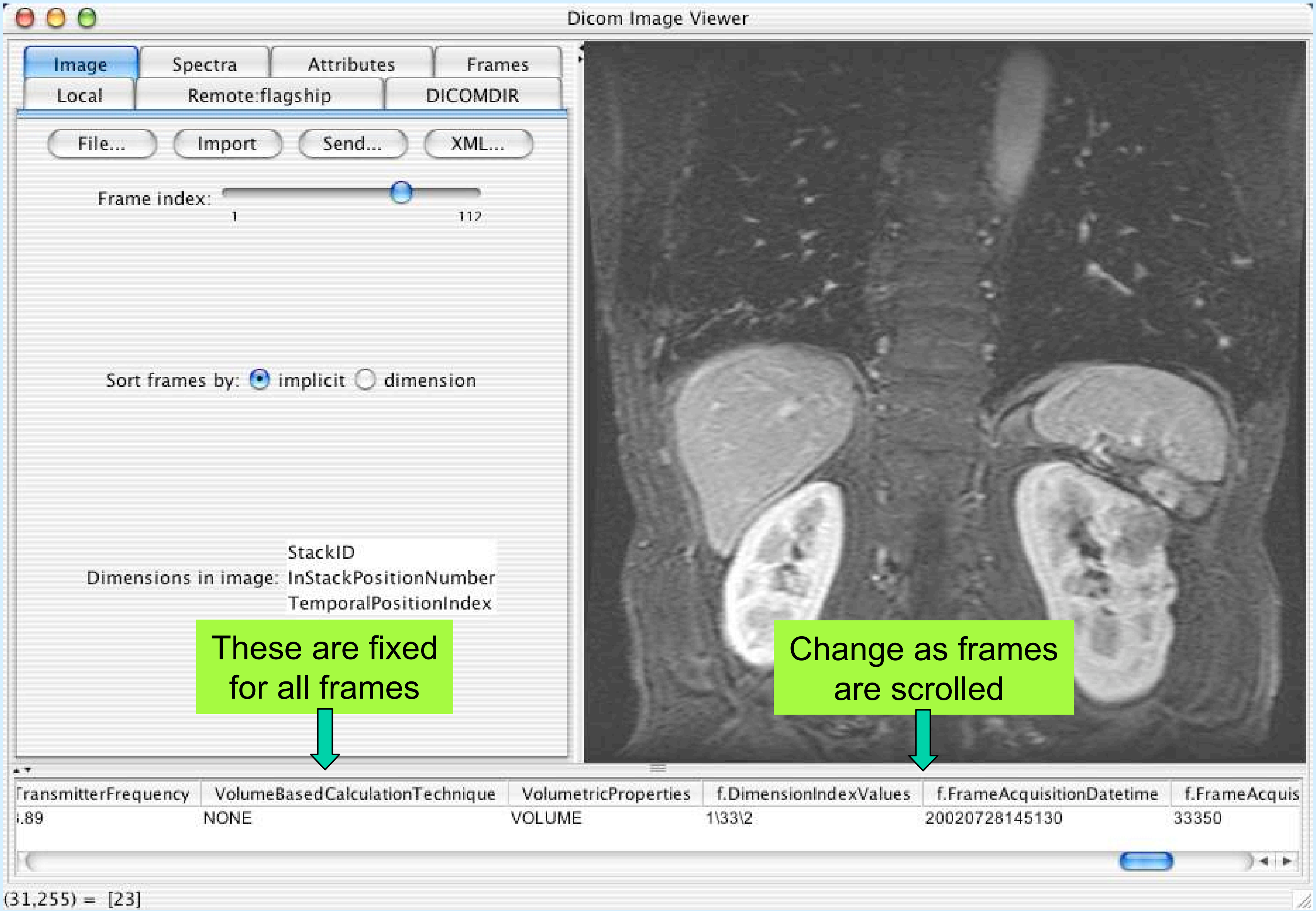
Implementation Challenges

- Annotation of attributes varying per-frame
- Convey overall “view” of what is in object
- Handle presence of “dimensions”
- New pipelines - color, real world values
- VOI LUT issues - varying per-frame
- Large size of images: hundreds of frames
- Spectroscopy

Annotation

- How to annotate hundreds of attributes that may vary per frame ?
- A real workstation would try to decorate the displayed image tile
- Tool uses separate scrolling window containing all the attributes present in the object
- Those in the “top level” dataset or the shared functional group sequence item are constant
- Those in (any of) the per-frame functional group sequence items are updated dynamically as the frames are scrolled

DICOM
Digital Imaging and Communications in Medicine



Overall View

- With hundreds of attributes vary per frame, how can one convey a sense of what is going on ?
- Tabular representation of only those attributes that vary on a per-frame basis, like a spreadsheet
- Limitations: as long as there is a meaningful “single value” to be displayed for each frame, this is easy; however, when there are lists of multiple values for a single frame the tool currently shows just the first (e.g. for list of SAR values)
- Additional features requested:
  - Sorting by columns
  - Sorting by dimension index values

Dicom Image Viewer

Local | Remote:flagship | DICOMDIR | Image | Spectra | Attributes | Frames

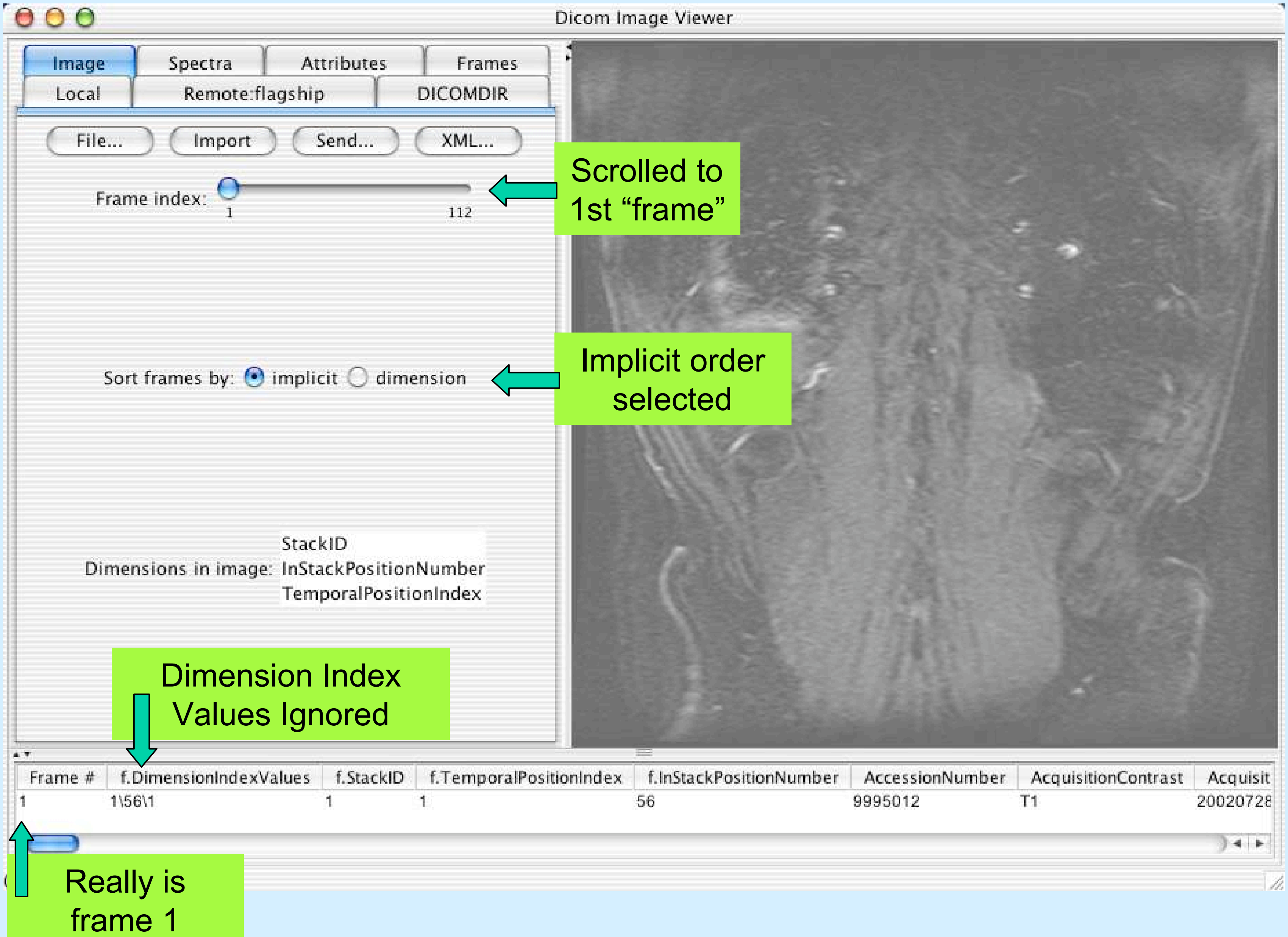
Frame #	f.Di...	f.FrameAcquisitio...	f.Fra...	f.F...	f.FrameReferenceDa...	f.ImagePositionPatient	f.I...	f...	f....	f.Windo...	f.WindowWidtl
20	1\37\1	20020728144923	33350	1	20020728144939.675	-163.200\64.8000\195.800	37	1	4	33.0000	100.000
21	1\36\1	20020728144923	33350	1	20020728144939.675	-163.200\63.3000\195.800	36	1	4	33.0000	100.000
22	1\35\1	20020728144923	33350	1	20020728144939.675	-163.200\61.8000\195.800	35	1	4	34.0000	100.000
23	1\34\1	20020728144923	33350	1	20020728144939.675	-163.200\60.3000\195.800	34	1	4	34.0000	100.000
24	1\33\1	20020728144923	33350	1	20020728144939.675	-163.200\58.8000\195.800	33	1	4	34.0000	100.000
25	1\32\1	20020728144923	33350	1	20020728144939.675	-163.200\57.3000\195.800	32	1	4	33.0000	100.000
26	1\31\1	20020728144923	33350	1	20020728144939.675	-163.200\55.8000\195.800	31	1	4	32.0000	100.000
27	1\30\1	20020728144923	33350	1	20020728144939.675	-163.200\54.3000\195.800	30	1	4	32.0000	100.000
28	1\29\1	20020728144923	33350	1	20020728144939.675	-163.200\52.8000\195.800	29	1	4	31.0000	100.000
29	1\28\1	20020728144923	33350	1	20020728144939.675	-163.200\51.3000\195.800	28	1	4	31.0000	100.000
30	1\27\1	20020728144923	33350	1	20020728144939.675	-163.200\49.8000\195.800	27	1	4	30.0000	100.000
31	1\26\1	20020728144923	33350	1	20020728144939.675	-163.200\48.3000\195.800	26	1	4	30.0000	100.000
32	1\25\1	20020728144923	33350	1	20020728144939.675	-163.200\46.8000\195.800	25	1	4	30.0000	100.000
33	1\24\1	20020728144923	33350	1	20020728144939.675	-163.200\45.3000\195.800	24	1	4	30.0000	100.000
34	1\23\1	20020728144924	33350	1	20020728144940.675	-163.200\43.8000\195.800	23	1	5	30.0000	100.000
35	1\22\1	20020728144924	33350	1	20020728144940.675	-163.200\42.3000\195.800	22	1	5	30.0000	100.000
36	1\21\1	20020728144924	33350	1	20020728144940.675	-163.200\40.8000\195.800	21	1	5	30.0000	100.000
37	1\20\1	20020728144924	33350	1	20020728144940.675	-163.200\39.3000\195.800	20	1	5	30.0000	100.000
38	1\19\1	20020728144924	33350	1	20020728144940.675	-163.200\37.8000\195.800	19	1	5	29.0000	100.000
39	1\18\1	20020728144924	33350	1	20020728144940.675	-163.200\36.3000\195.800	18	1	5	29.0000	100.000
40	1\17\1	20020728144924	33350	1	20020728144940.675	-163.200\34.8000\195.800	17	1	5	29.0000	100.000
41	1\16\1	20020728144924	33350	1	20020728144940.675	-163.200\33.3000\195.800	16	1	5	30.0000	100.000
42	1\15\1	20020728144924	33350	1	20020728144940.675	-163.200\31.8000\195.800	15	1	5	31.0000	100.000
43	1\14\1	20020728144924	33350	1	20020728144940.675	-163.200\30.3000\195.800	14	1	5	29.0000	100.000
44	1\13\1	20020728144924	33350	1	20020728144940.675	-163.200\28.8000\195.800	13	1	5	28.0000	100.000
45	1\12\1	20020728144924	33350	1	20020728144940.675	-163.200\27.3000\195.800	12	1	5	28.0000	100.000
46	1\11\1	20020728144924	33350	1	20020728144940.675	-163.200\25.8000\195.800	11	1	5	27.0000	100.000
47	1\10\1	20020728144925	33350	1	20020728144941.675	-163.200\24.3000\195.800	10	1	6	28.0000	100.000
48	1\9\1	20020728144925	33350	1	20020728144941.675	-163.200\22.8000\195.800	9	1	6	30.0000	100.000

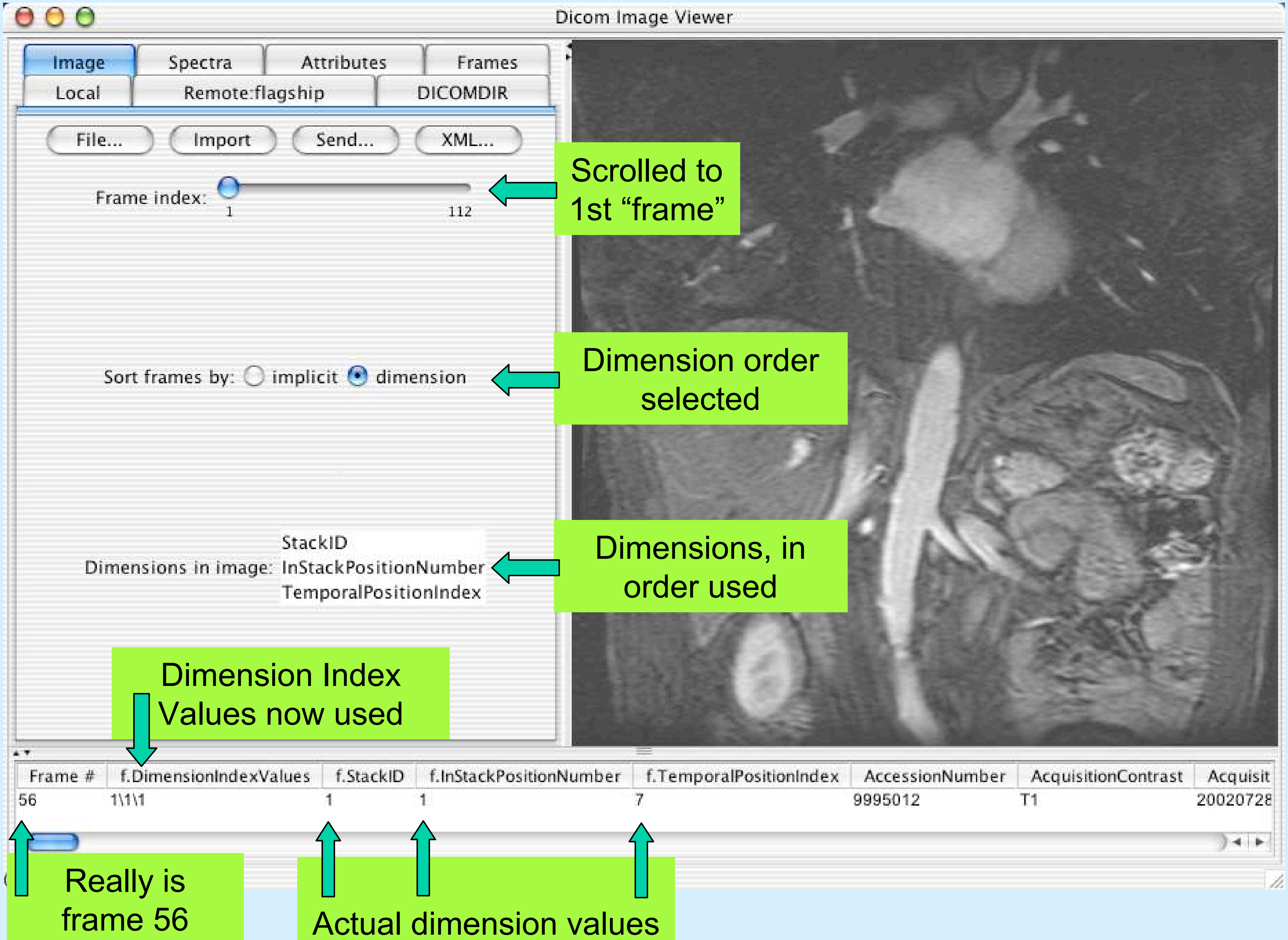
(9,259) = [2]

Dimensions

- Dimensions are used to convey from the object creator how the frames could or should be ordered for display
- Default order applied by tool is implicit order of frames as encoded in the Pixel Data attribute
- User may toggle between sorting by implicit order or dimension order
- No limit to the number of dimensions
- No need to “understand” natural sort order of attributes that are the target of the dimension indices … the indices are explicitly conveyed (i.e. the sorting is done by the object creator)
- Applies to both images and spectroscopy objects
- Trivial to implement … just de-reference the frame indices through an array with a pre-computed order based on sorted dimension indices

DICOM
Digital Imaging and Communications in Medicine

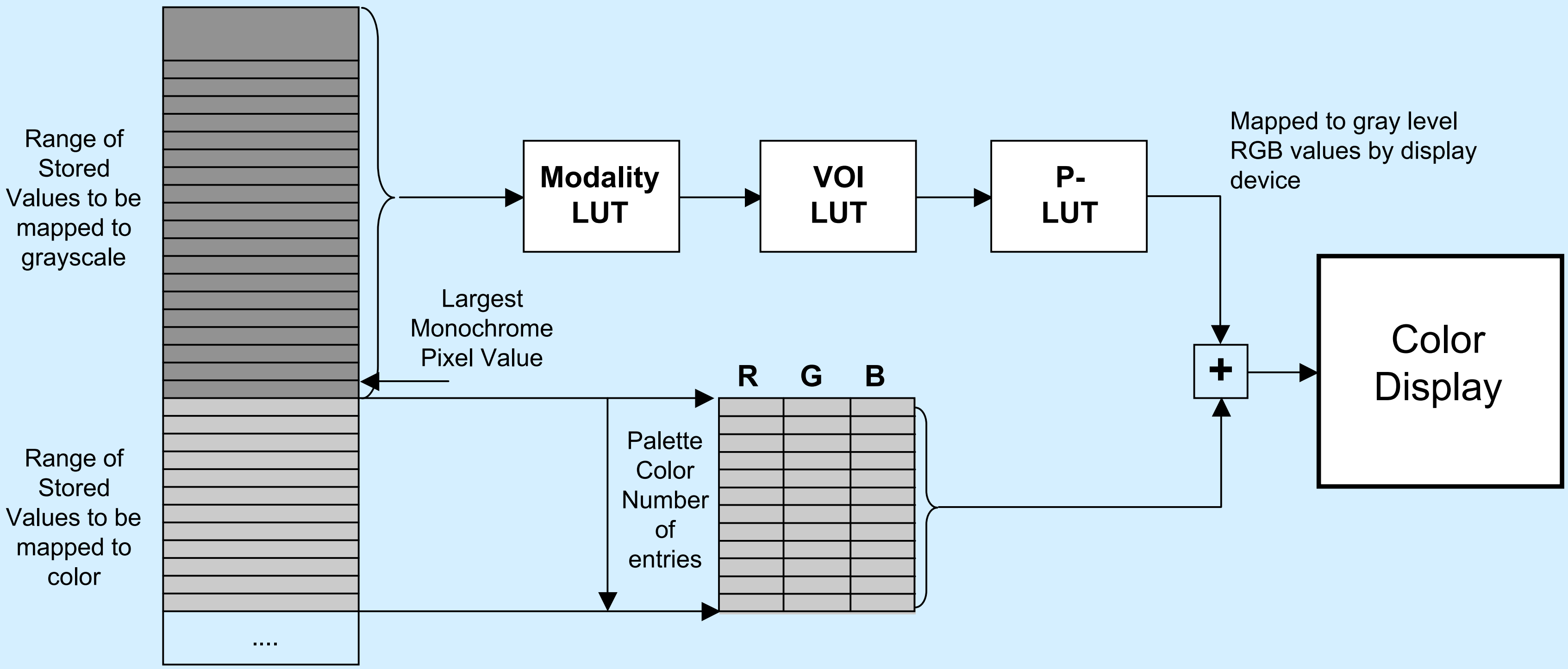



Dicom Image Viewer
Image
Spectra
Attributes
Frames
Local
Remote:flagship
DICOMDIR
File...
Import
Send...
XML...
Frame index:
1
112
Scrolled to 1st "frame"
Sort frames by: implicit dimension
Dimension order selected
StackID
Dimensions in image: InStackPositionNumber
TemporalPositionIndex
Dimensions, in order used
Dimension Index Values now used
Frame # | f.DimensionIndexValues | f.StackID | f.InStackPositionNumber | f.TemporalPositionIndex | AccessionNumber | AcquisitionContrast | Acquisit
56 | 1\1\1 | 1 | 1 | 7 | 9995012 | T1 | 20020728
Really is frame 56
Actual dimension values

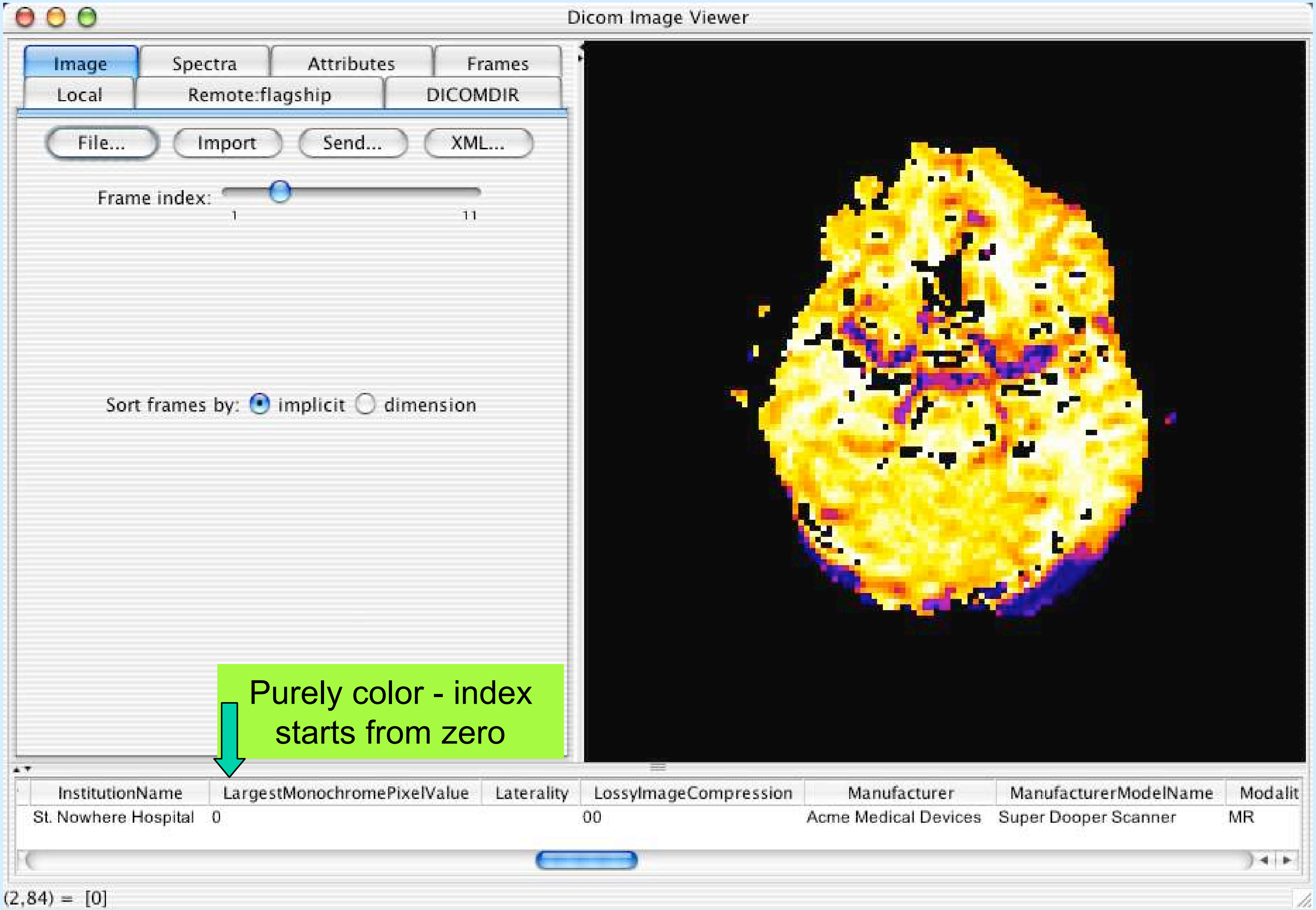
Pipelines

- Basic grayscale pipeline is same as usual
  - Modality LUT - always linear (no LUT)
  - VOI LUT - always center/width (no LUT)
- Supplemental palette color
  - Use “high” pixel values as index into supplied LUT and ignore during interactive windowing
- Real-world value mapping
  - Interactive display of pixel under cursor piped through one or more transformations as supplied, and display Code Meaning of supplied units

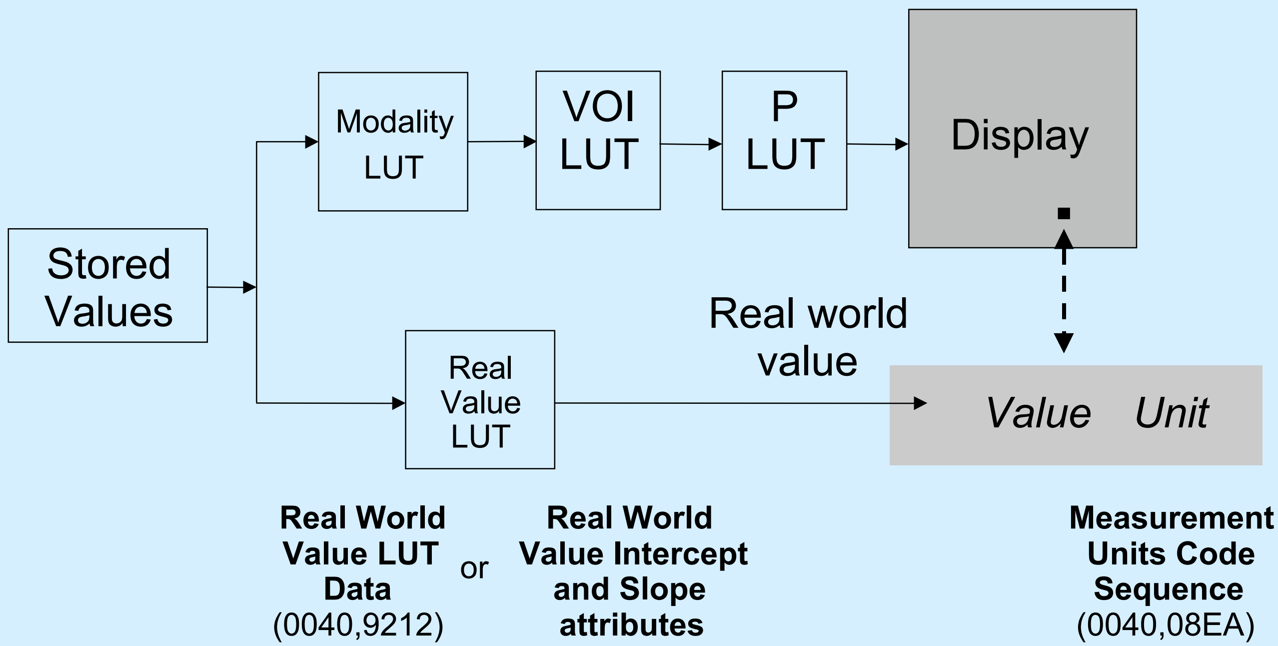
Pipelines

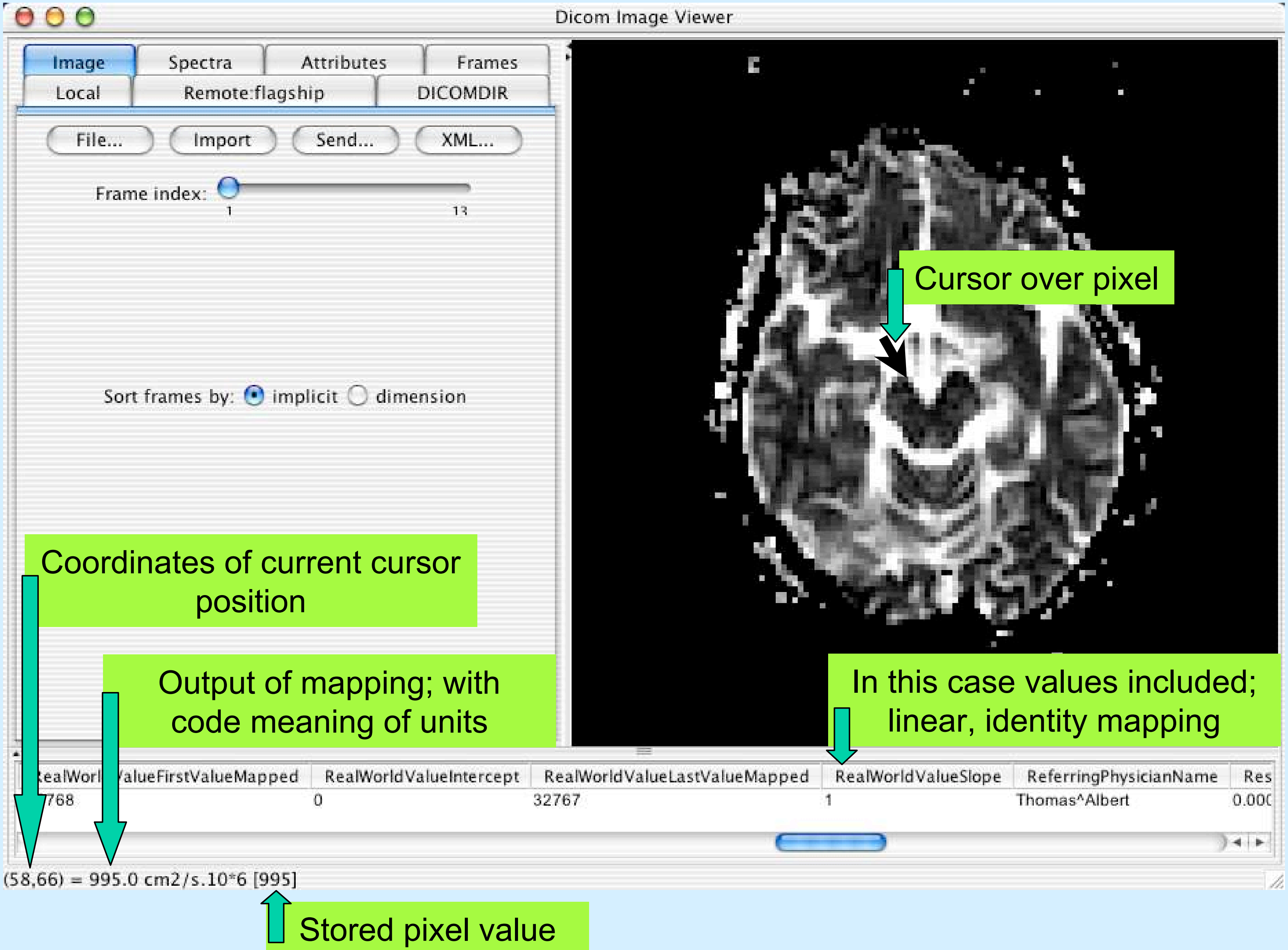

Range of Stored Values to be mapped to grayscale
Modality LUT
VOI LUT
P- LUT
Mapped to gray level RGB values by display device
Largest Monochrome Pixel Value
R
G
B
+
Color Display
Range of Stored Values to be mapped to color
Palette Color Number of entries
....

Mixed color - index greater than zero


Dicom Image Viewer
Image
Spectra
Attributes
Frames
Local
Remote:flagship
DICOMDIR
File...
Import
Send...
XML...
Frame index:
1
11
Sort frames by: implicit dimension
Purely color - index starts from zero
InstitutionName
LargestMonochromePixelValue
Laterality
LossyImageCompression
Manufacturer
ManufacturerModelName
Modalit
St. Nowhere Hospital
0
00
Acme Medical Devices
Super Dooper Scanner
MR
(2,84) = [0]

Pipelines


Modality LUT
VOI LUT
P LUT
Display
Stored Values
Real world value
Real Value LUT
Value Unit
Real World Value LUT Data (0040,9212)
or
Real World Value Intercept and Slope attributes
Measurement Units Code Sequence (0040,08EA)


Dicom Image Viewer
Image
Spectra
Attributes
Frames
Local
Remote:flagship
DICOMDIR
File...
Import
Send...
XML...
Frame index:
1
13
Sort frames by: implicit dimension
Cursor over pixel
Coordinates of current cursor position
Output of mapping; with code meaning of units
In this case values included; linear, identity mapping
RealWorldValueFirstValueMapped
RealWorldValueIntercept
RealWorldValueLastValueMapped
RealWorldValueSlope
ReferringPhysicianName
0
32767
1
Thomas^Albert
(58,66) = 995.0 cm2/s.10*6 [995]
Stored pixel value

VOI LUT Issues

- Traditionally, DICOM objects have either:
  - No VOI LUT at all included
  - One or more linear window center/width sets
  - One or more LUTs (esp. for CR and DX)
- New MR object allows (but does not require)
  - One or more linear window center/width sets only - no LUTs
  - May vary per-frame or be in shared functional group
- Implementing this:
  - Need to have meaningful (statistical) default in case absent
  - Ability to select and apply VOI values supplied for each frame
  - No need to support LUTs
- Current tool behavior
  - Use first window pair from each frame as frames are scrolled (not selectable)
  - Use mean of actual pixel values as center, actual range as width, if none
  - If user adjusts window, applies to all frames (overrides any VOI values)

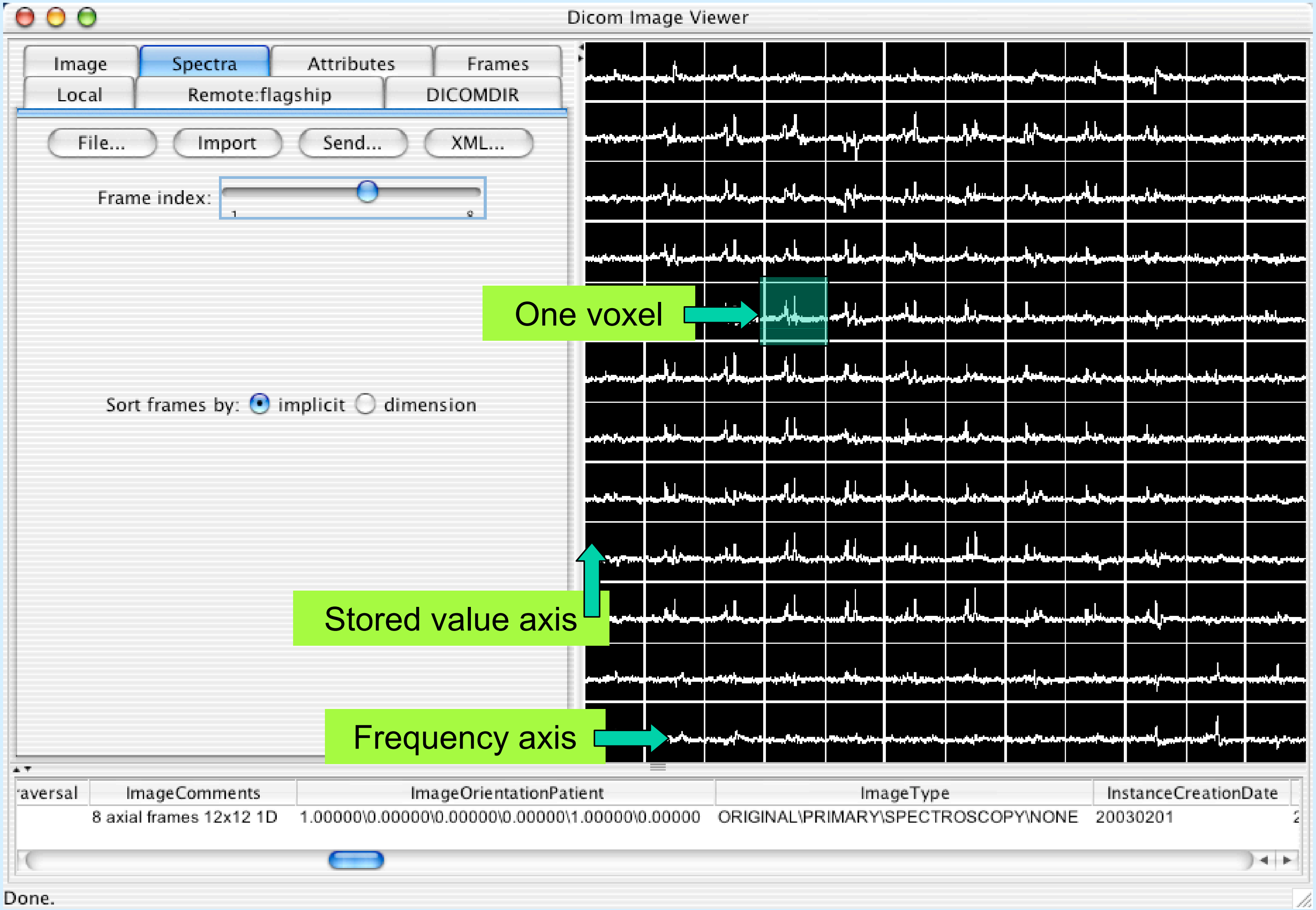
Large size of images

- Multi-frame images may get really large
  - Single huge file
  - Concatenations to be viewed as a whole
- Current viewing tool is very memory extensive
  - Maintains in memory both source pixel data and transformed pixels for current frame (rescaled, windowed, resized and platform-appropriate buffer for repainting)
  - Grabs really huge heap sizes at invocation (e.g., 512MB)
  - If heap requested is really in OS virtual memory (exceeds physical memory), will thrash horribly
- In future, use of memory-mapped file feature of Java may allow for much larger images

Spectroscopy

- What is “spectroscopy” anyway ?
  – Very new to most image and DICOM engineers
  – Floating point and/or complex values
  – Stored as arrays like rasters, but not pixels
  – Need to be represented as graphs against a frequency axis
  – May be 1D data for a single or multiple “voxels” (areas in space)
  – May theoretically be 2D with something else other than frequency as the other dimension (rare in clinical use)
- Test tool support:
  – Single/multi-voxel 1D display: graph of stored value vs. frequency
  – Need to be related back to physical location by reference to structural images - a feature to be added in Phase 2

DICOM
Digital Imaging and Communications in Medicine



Implementation Experience

- Test objects
  - Test images & spectroscopy objects
  - Creation from existing single frame bulk data
  - Synthesis of multi-frame & technique attributes
- Test tool
  - Annotation, grayscale pipeline & multi-frame navigation
  - Display of spectra
- Validation
  - Challenges of verifying complex nested conditions
  - Use of XSL-T and Xpath for definition/execution

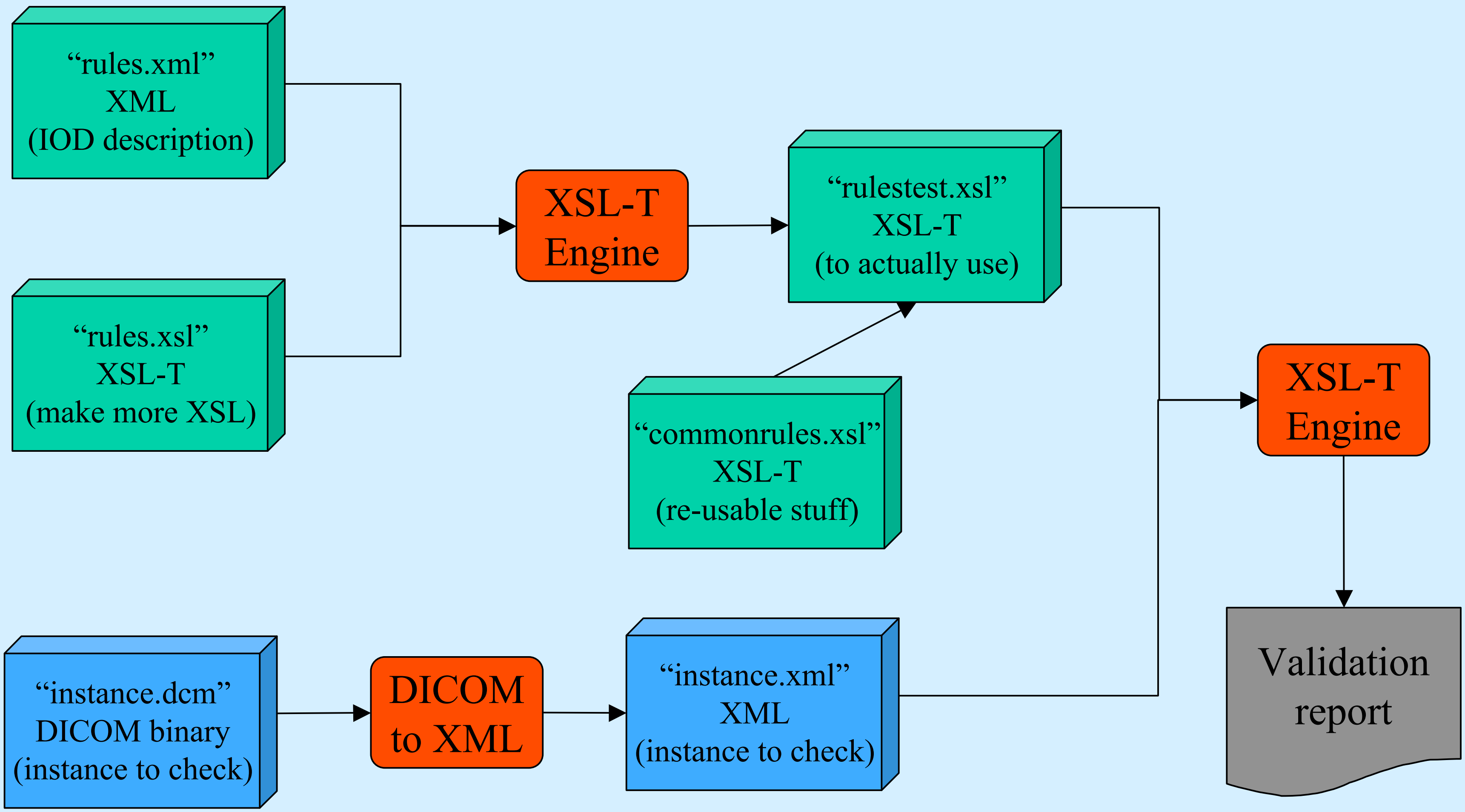
Validation

- Principles of operation
- Design decisions
- Preliminary experience
  - Actual errors encountered in sample images
  - Unexpected consequences of the details of the standard
  - Limitations of the validation tool design

Principles of Operation

- Convert XML IOD definition into XSL-T rules
  - Also performed using XSL-T
- Convert DICOM into XML attribute descriptions
- Apply XSL-T rules to XML attribute descriptions
- Output is a report listing deficiencies

Principles of Operation



Design Decisions, Consequences

- Why XSL-T ?
  - Previous experience with hard-coded (C++) validator
  - Primary difficulty - describing complex conditions
    - Presence or absence of modules, macros, attributes
    - Paths to attributes included in conditions
    - Adding "ad hoc" verification rules derived from text descriptions
    - Handling "may be present otherwise … only if" checks
  - Non-standard template input format driving the C++ validator grew more and more complex with ad hoc extensions
- New MR object has very complex conditions, especially between nested sequence attributes

Design Decisions, Consequences

- Form of XML representation of IOD
  - Ideally would be automatically generated from published XML description of DICOM (which doesn’t yet exist) - for now, typed in by hand
  - IODs, Modules, Macros, Attributes
  - Conditions
  - VR and VM of every occurrence of an attribute

Design Decisions, Consequences

- Conditions
  - Mandatory modules, macros and attributes easy
  - Conditional and user optional - not so easy
    - Verifiable conditions - e.g., ImageType value 1 = ORIGINAL
    - Real-world conditions - e.g., contrast administered
    - Is module/macro/attribute actually present anyway, and hence needs to be checked ? - e.g., if any (mandatory ?) attributes of a module are present, module is present (and hence checked)
- Strong desire to use Xpath to build conditions

Xpath rules

- XSL-T “navigates” using Xpath expressions:
  - value of a sibling attribute:

```
BitsAllocated /value[@number=1] = 16
```

  - path relative to current location:

```
../../MRImageFrameTypeSequence/Item[@number=1]
/FrameType/value[@number=1]='ORIGINAL'
```

  - absolute path from the root:

```
/DicomObject/EchoPulseSequence/value[@number=1]='GRADIENT’
```

  - occurrence at any depth:

```
count(//ReferencedImageSequence) &gt; 0
```

XML Format Decisions

- Need to represent the DICOM instance in XML prior to validating with XSL-T rules
- Fairly straightforward transliteration
  - Names of DICOM data elements as XML element names
  - Group/element tag included but not used in validation
  - VR always included (either from explicit transfer syntax or dictionary used during conversion)
  - Values (and sequence items) explicitly numbered - makes XSL-T considerably simpler (e.g. to validate defined terms applicable to a particular value only, such as with Image Type)
  - Attribute values themselves included as text of <value> element, with binary values (floats, integers) converted to text first

What to validate ?

- Value Representation
  - Matches dictionary
  - Value valid for VR (hard with XML intermediate form)
- Value Multiplicity
- Presence/ or absence of modules, attributes
- Defined terms and enumerated values
- Additional constraints specified in text in standard

Preliminary Experience

- Typical example of output:

```
Error: PatientModule/PatientID/: PatientID: Incorrect VR - got CS but expected LO

Error: PatientModule/PatientID/: PatientID: Incorrect value multiplicity - got 2 but
    expected 1-1

Error: ImagePixelModule/SamplesPerPixel/: SamplesPerPixel: Must be 1 when
    PhotometricInterpretation is MONOCHROME2

Warning: ImagePixelModule/SamplesPerPixel/: SamplesPerPixel: Unrecognized defined term 2

Error: ImagePixelModule/PlanarConfiguration/: : Missing conditional attribute
```

Output Format

- Where in the IOD is the error ?

```
Error: PatientModule/PatientID/:
```

- Where in the Instance is the error ?

```
PatientID:
```

- What is the nature of the error ?

```
Incorrect VR - got CS but expected LO
```

Output Format

- Where in the IOD is the error ?

`PlanePositionMacro/PlanePositionSequence/ImagePositionPatient/`

- Where in the Instance is the error ?

```
PerFrameFunctionalGroupsSequence/Item[32]/
PlanePositionSequence/Item[1]/ImagePositionPatient
```

- What is the nature of the error ?

`Conditional attribute present when condition not satisfied`

Unexpected Consequences

- Interaction of Shared vs. Per-frame functional groups and conditions based on top-level attributes
- When is a particular functional group required ?
- Does it vary per-frame or not ?

Functional Group problems

- Example of a legitimate problem: CARANGIO image
  - Validator complains that no MR Spatial Saturation Sequence is present in any of the per-frame functional group items, nor is it present in the shared functional group item
  - The Spatial Presaturation (top-level) attribute has a value of SLAB, which triggers the condition for the presence of the MR Spatial Saturation Functional Group Macro
- Conclusion:
  - Error in constructing the image; the MR Spatial Saturation Sequence should have been included with no Items (zero-length Type 2 sequence), rather than being omitted completely - i.e., the semantics are that though sat slabs were used, the information about their locations is “unknown”

Functional Group problems

- Example of a problem: CARANGIO image
  - Validator complains that MR FOV/Geometry Sequence is missing in first 13 per-frame functional group items (it is present in the rest)
  - The Frame Type of these “bad” items is DERIVED, whereas the rest are original
  - The standard condition is predicated on Geometry of k-Space Traversal being RECTILINEAR (a top-level attribute), and the Image Type being ORIGINAL or MIXED, all of which are satisfied
  - Yet, clearly for a MIXED Image Type it is possible to have DERIVED frames, for which the MR FOV/Geometry Sequence is not meaningful and shouldn’t be included
- Conclusion
  - Error in the standard ? Symptomatic of mixed frames in general ?
  - Arbitrary prohibition on the kind of object that can be constructed ?

Mixing “old” & “new” problems

- Example of a problem: all test images
  - Validator complains that Laterality in the General Series module has an illegal enumerated value (i.e. the empty string that is the zero length value)
  - In the new MR object, laterality is specified in the per-frame Frame Anatomy macro as Frame Laterality
  - There was no modification of the condition on the inclusion of Laterality in General Series to account for this
- Conclusion
  - Error in the standard - clearly the condition on Laterality should be modified (as it was when DX Image Laterality was added)
  - General question though - is zero length a legitimate enumerated value for a Type 2 or 2C attribute ? Yes, therefore validation rules flagged this incorrectly !

Mixing “old” & “new” problems

- Example of a problem: all test images
  - Validator complains that Pixel Aspect Ratio is missing from the Image Pixel Module
  - Pixel Aspect Ratio’s presence is conditioned by the presence of the Image Plane Module
  - In the new MR object, the equivalent of the Image Plane module is moved down into functional group macros that may vary per-frame
  - There was no modification of the condition on the inclusion of Pixel Aspect Ratio in Image Pixel to account for this
- Conclusion
  - Error in the standard - clearly the condition on Pixel Aspect Ratio should be modified and a CP is needed
  - In the interim, it would be inappropriate to start including Pixel Aspect Ratio in images just because the validator complains

Limitations of the Tool

- Potentially verifiable but complex conditions not yet implemented
- Real-world conditions are not validated
  - Some conditional modules validated only when evidence of their presence
  - Could have synthetic "pre-conditions", either manually set as preferences, or all possibilities automatically explored
- Limited potential for validation of attribute values against VR limitations … conversion to XML impacts this
  - E.g., it is easier to validate character lengths of national character sets (since already converted to Unicode), but there may be some canonicalization of spaces, etc.
  - This really should be done in the DICOM to XML translation step
- No validation of contents or length of bulk data
  - Bulk data (such as pixel data, spectroscopy data and LUTs) are not included in the XML translation

Value of the Tool

- Obviously the tool can only check mechanically against the standard
- Can not actually “understand” the contents of the image object
- I.e. does it really make “sense” in the entire context of the acquisition ?
- Not a real problem - presumably, vendors of real rather than test images will only build objects that have meaning since they reflect actual acquisitions and processing results
- Primary value of the tool - protect against trivial programming errors (e.g. defined term spelling mistakes, empty items, missing attributes, etc.) that are known to be common and do affect interoperability

Conclusion

- Test objects
  - Hardest part of building the objects was gathering enough good samples !
  - Conversion of existing single-frame objects on a routine basis is not as easy as might be expected by some
- Test tool
  - Nothing particularly challenging about the new object
  - The large size of the new images is a feature of the application requirement - it would be the same regardless of whether images are stored as single or multiple frames
  - New object contains many mandatory and otherwise useful features that make display easier, not harder
- Validation
  - Greatest difficulty is in verifying complex nested conditions
  - Use of XSL-T and Xpath proved to be a good choice because of this
  - Experience using this and earlier validators helped cleaned up the test images
  - The exercise of translating the standard into conditions and tests in the validator exposes inconsistencies or weaknesses in the standard, though most of these are relatively minor