Extension of an Open Source DICOM Toolkit to Support SCP-ECG Waveforms

David Clunie

PixelMed Publishing

2nd ECG Workshop - Berlin April 2004

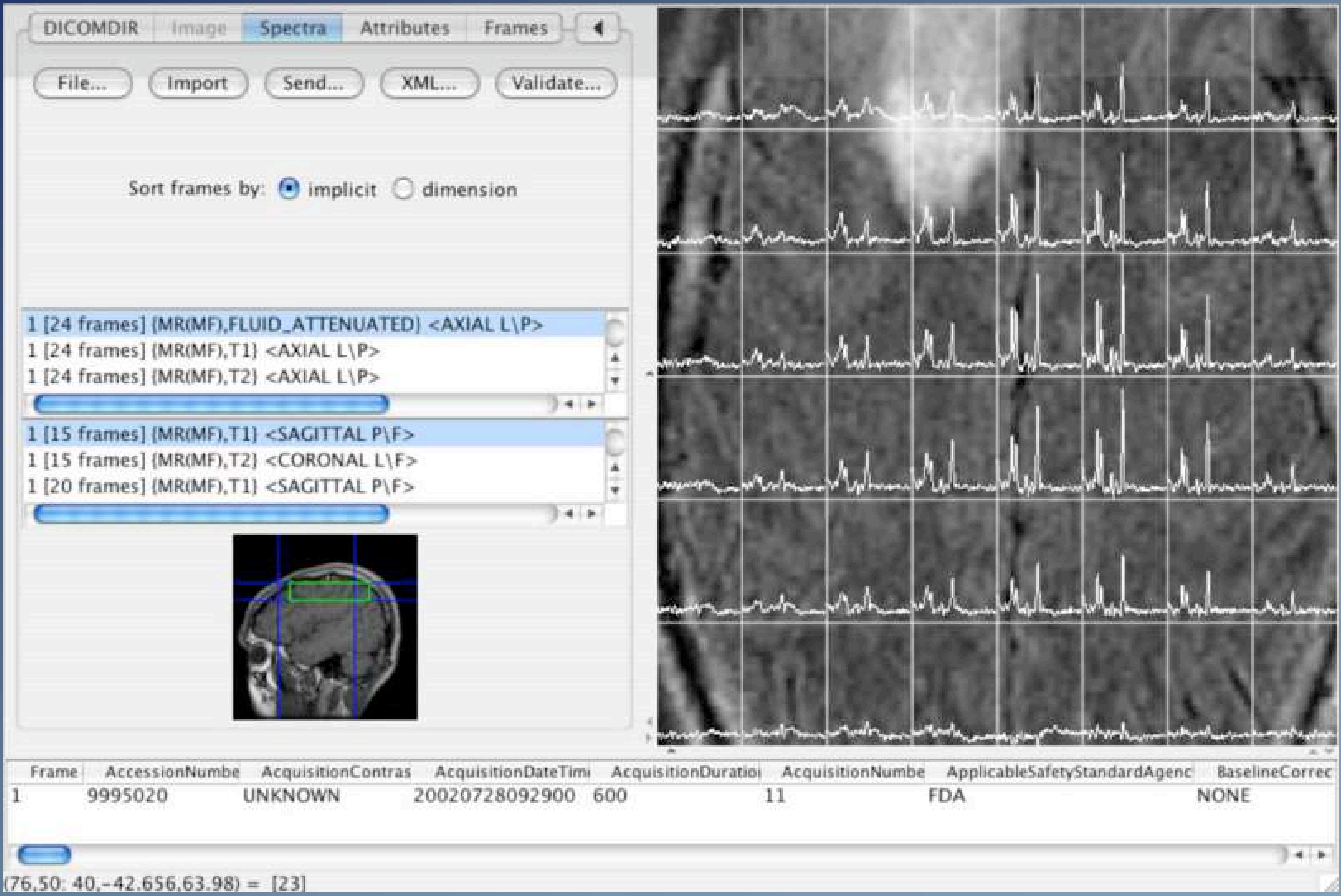
Overview

- Background
- Motivation
- Process
- Challenges
- Future

PixelMed DICOM Toolkit

- Open source
- 100% pure Java, therefore platform independent
- DICOM support
  - Parsing and creation of DICOM datasets
  - File encoding, media and network transfer
  - Data dictionary
  - Validation
- Primarily oriented towards image display & encoding
- Recent addition of MR Spectroscopy
  - Core elements present for 1D waveform display support

DICOM MR Spectroscopy



Sort frames by: [✓ implicit] dimension

1 [24 frames] {MR(MF),FLUID_ATTENUATED} <AXIAL L\P>
1 [24 frames] {MR(MF),T1} <AXIAL L\P>
1 [24 frames] {MR(MF),T2} <AXIAL L\P>

1 [15 frames] {MR(MF),T1} <SAGITTAL P\F>
1 [15 frames] {MR(MF),T2} <CORONAL L\F>
1 [20 frames] {MR(MF),T1} <SAGITTAL P\F>

Frame	AccessionNumbe	AcquisitionContras	AcquisitionDateTim	AcquisitionDuratio	AcquisitionNumbe	ApplicableSafetyStandardAgenc	BaselineCorrec
1	9995020	UNKNOWN	20020728092900	600	11	FDA	NONE

(76,50: 40,-42.656,63.98) = [23]

DICOM Waveforms

- Time-based waveforms added to DICOM in 2000
- Includes ECG, Holter, hemodynamic w/forms
- Intended for imaging devices
- E.g. cardiac catheterization lab
- Very poor support by vendors
- Despite some tools and a few samples
- Simple mechanism (if existing DICOM toolkit)
- Tag-value pairs & “array” of 16-bit data samples
- No compression (except zip)

Motivation to support SCP-ECG

- A more plausible standard ?
- Actually implemented by vendors ?
- Well-documented
- Reference data files available for testing
- SCP -> DICOM converter available
- Incentive of programming competition

Implementing SCP-ECG (I)

- Different sections with different encoding forms
- Tag-value pairs in section 1 (e.g. names)
- Fixed binary fields and variable length segments in other sections
- Tagged annotations (no samples, so not yet implemented)
- This mixed approach was very tedious to implement
- Required considerable typing of code, attention to detail, testing and experimentation

Implementing SCP-ECG (2)

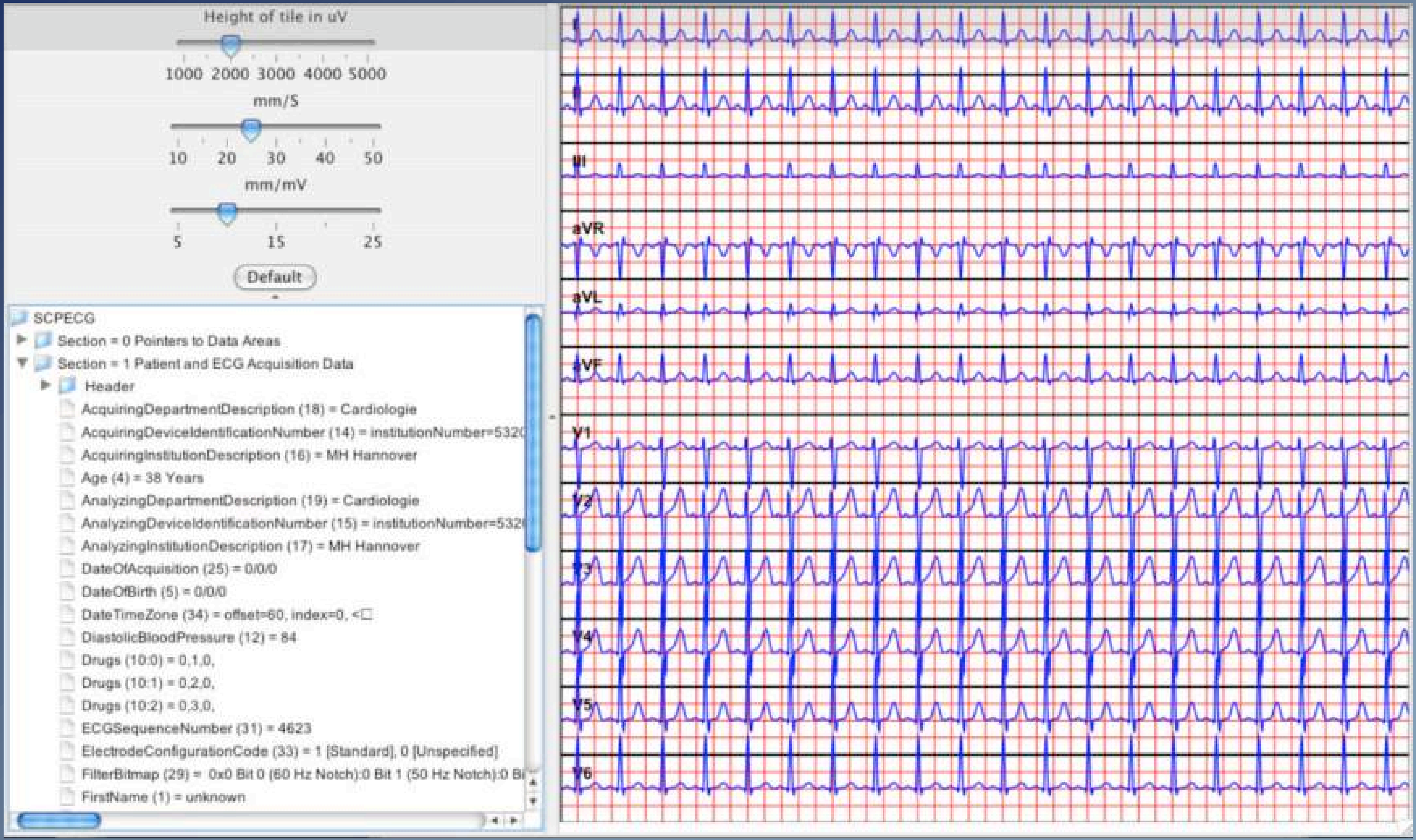
- Redundancy reduction (entropy coding) alone
- Huffman coding very straightforward to implement
- Numeric examples in standard were a useful guide

- Additional compression by subtraction of reference beats and decimation (reduced sampling rate) outside protected zones
- The most difficult part of the project
- Still not quite right, judging by quantitative evaluation
- Filtering and interpolation not yet implemented

Features Implemented

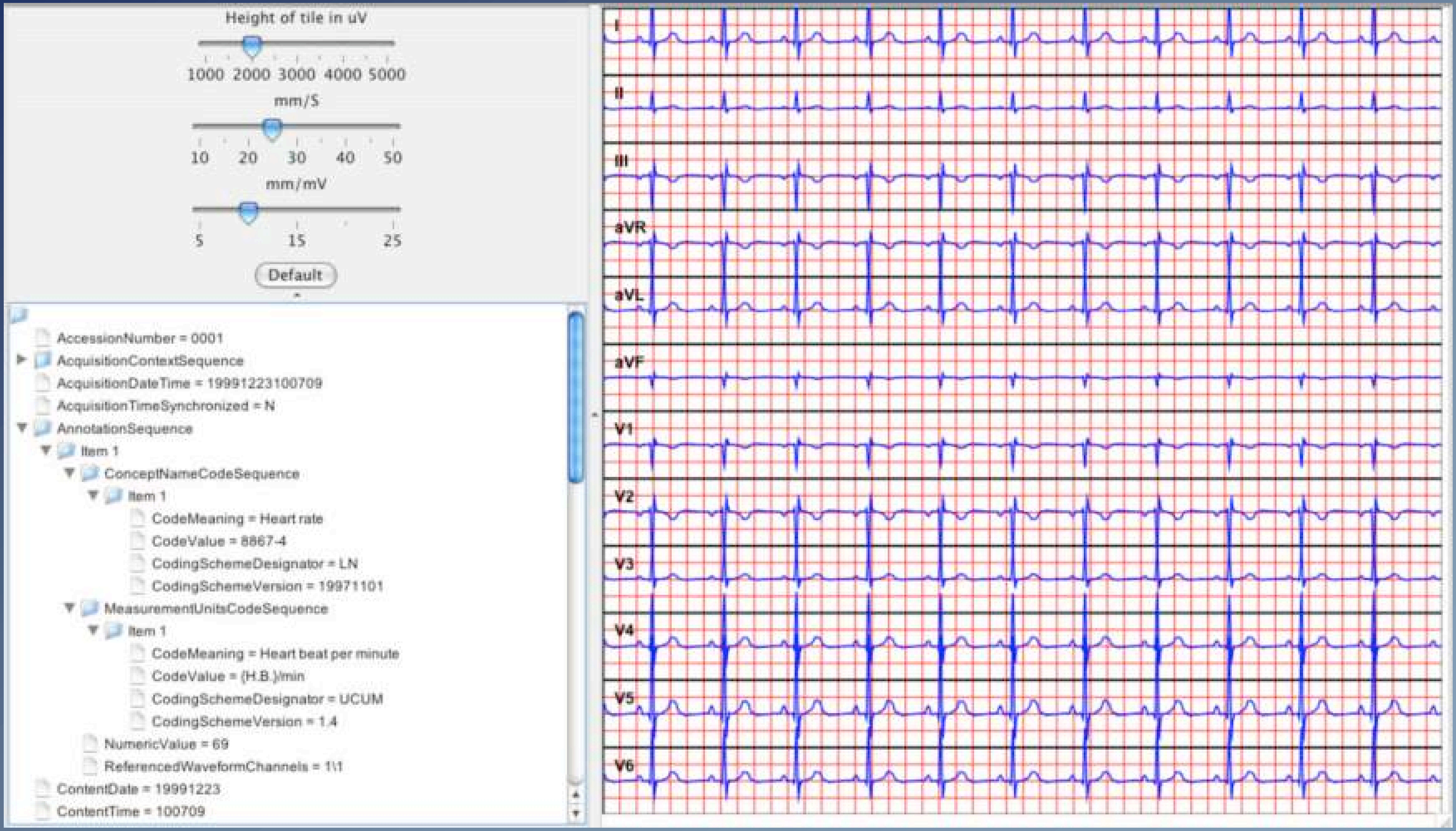
- Read & display DICOM ECG or SCP-ECG
- Validate SCP-ECG format during reading
- Derive 12 leads from 8 if necessary
- Allow user to scroll and scale
- Tree-oriented display of header
- Includes tag-value pairs of Section 1

Display of SCP-ECG



```
SCPECG
▶ Section = 0 Pointers to Data Areas
▼ Section = 1 Patient and ECG Acquisition Data
    ▶ Header
      AcquiringDepartmentDescription (18) = Cardiologie
      AcquiringDeviceIdentificationNumber (14) = institutionNumber=532
      AcquiringInstitutionDescription (16) = MH Hannover
      Age (4) = 38 Years
      AnalyzingDepartmentDescription (19) = Cardiologie
      AnalyzingDeviceIdentificationNumber (15) = institutionNumber=532
      AnalyzingInstitutionDescription (17) = MH Hannover
      DateOfAcquisition (25) = 0/0/0
      DateOfBirth (5) = 0/0/0
      DateTimeZone (34) = offset=60, index=0, <□
      DiastolicBloodPressure (12) = 84
      Drugs (10:0) = 0,1,0,
      Drugs (10:1) = 0,2,0,
      Drugs (10:2) = 0,3,0,
      ECGSequenceNumber (31) = 4623
      ElectrodeConfigurationCode (33) = 1 [Standard], 0 [Unspecified]
      FilterBitmap (29) =  0x0 Bit 0 (60 Hz Notch):0 Bit 1 (50 Hz Notch):0 Bi
      FirstName (1) = unknown
```

Display of DICOM ECG



```
AccessionNumber = 0001
▶ AcquisitionContextSequence
AcquisitionDateTime = 19991223100709
AcquisitionTimeSynchronized = N
▼ AnnotationSequence
  ▼ Item 1
    ▼ ConceptNameCodeSequence
      ▼ Item 1
          CodeMeaning = Heart rate
          CodeValue = 8867-4
          CodingSchemeDesignator = LN
          CodingSchemeVersion = 19971101
    ▼ MeasurementUnitsCodeSequence
      ▼ Item 1
          CodeMeaning = Heart beat per minute
          CodeValue = {H.B.}/min
          CodingSchemeDesignator = UCUM
          CodingSchemeVersion = 1.4
      NumericValue = 69
      ReferencedWaveformChannels = 1\1
ContentDate = 19991223
ContentTime = 100709
```

Observations - SCP-ECG

- SCP-ECG harder to implement than DICOM
- Especially if using existing DICOM toolkit
  - Benefit of reusing encoding for multiple purposes rather than reinventing the wheel ?
- Complexity primarily from high compression
  - Given current communication technology, is the complexity (and potential quantitative loss) really worth this complexity ?

Observations - Choice of Java

- Inherently multiplatform - runs on MacOS X, Windows, Solaris, Linux with no additional effort

- Parsing and decompression take trivial time
- User interface sluggish
- Naïve approach largely at fault - displaying path of thousands of line segments between samples with no effort at re-sampling

Future Directions

- Improve quantitative results for reference beat subtraction, reversal of decimation, filtering and interpolation
- Patient/exam browser built from set of files or DICOMDIR file
- Annotate displayed waveforms with measurements supplied in files
- Ability to make & save annotations & measurements
- Conversion (import/export) between formats
- Add support for HL7 V3 XML format that is now designated for FDA Definitive QT Study (?)