

# An Open Source ECG Toolkit with DICOM

MJB van Ettinger, JA Lipton, MCJ de Wijs, N van der Putten, SP Nelwan

Erasmus MC, Rotterdam, The Netherlands

## Abstract

*Unlike the ubiquitous use of DICOM for sharing various medical image modalities, the storage and retrieval of digital 12-lead electrocardiograms across different ECG management systems is often limited to manual file-based export and import using vendor-specific solutions. In addition to SCP-ECG, a number of open standard initiatives have been introduced, such as DICOM-ECG and HL7 aECG. By extending and improving a previously developed ECG toolkit we have explored the capabilities of these open standards in terms of interoperability, filesize and transmission times. The ECG toolkit is almost entirely available under the Apache License Version 2.0.*

## 1. Introduction

When considering the future of the electronic health record, a primary goal might be the availability of the entire relevant medical history of a patient at the point of care, such as the ambulance, general practitioner and hospital. One of the most frequent examinations is the 12-lead electrocardiogram (ECG), but currently the exchange of ECGs is limited to users of the same manufacturer application at each side.

The exchange of a paper ECG recording can easily be achieved by fax. The introduction of the digitally stored ECG should have provided an easy way to exchange ECGs. Integrating the Healthcare Enterprise provides a solution with Retrieve ECG for Display, but does not provide signal exchange.

To enable us to exchange our resting ECGs to any medical center we explored the following open standards: SCP-ECG, HL7 aECG and DICOM-ECG.

SCP-ECG is part of an international standard for medical device communication (EN ISO 11073). The U.S. Food and Drug Administration have accepted the HL7 aECG format for submitting digital annotated ECGs obtained in certain drug trials [1]. DICOM supported waveform (supplement 30) in 2000 [2], and on March 9 2006 Mortara Instrument announced its adoption of the DICOM standard for diagnostic ECGs [3]. Adopting any of the open standards for ECGs could allow the independence from manufacturers, specifically for ECG carts and archiving systems. Additionally DICOM-ECG

would allow integration of ECGs into existing image exchanging systems, thereby reducing the need for custom ECG exchanging systems.

Currently the Thoraxcenter uses multiple proprietary ECG Management Systems. Three of these systems provide a resting ECG archive: Mortara E-Scribe, Draeger Infinity MegaCare and Welch Allyn CardioControl Workstation (CCW).

## 2. Methods

In 2005 a Unified ECG Framework was developed by the Thoraxcenter enabling the use of a single viewer for different resting ECG formats [4]. The main part of the Unified ECG Framework is a C# .NET ECG Toolkit [5]. The ECG toolkit can be used on multiple platforms, because it works using Mono, an open source implementation of the Microsoft .NET framework [6], except for the ECG printing application.

### 2.1. The ECG toolkit

The ECG toolkit was extended so it supports the three open standards: SCP-ECG, HL7 aECG and DICOM-ECG (based on DICOM# [7]). The ECG toolkit provides applications for conversion of ECGs, storage on a DICOM PACS, and for viewing. Printing is also possible in a Windows environment.

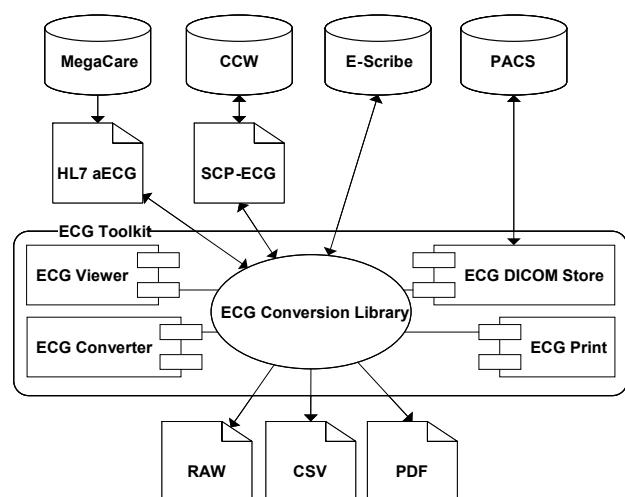


Figure 1. Hospital Application of ECG toolkit

Resting ECGs stored in a Mortara E-Scribe ECG management system can be accessed by using the UNIPRO plug-in. The Draeger Infinity MegaCare ECG management system, which stores the bedside resting ECGs, can export to the HL7 aECG format. Welch Allyn CardioControl Workstation supports SCP-ECG for resting ECGs.

An export to PDF (based on iTextSharp [8]) is also available. The capabilities of the ECG toolkit are illustrated by Figure 1.

## 2.2. Methods of testing

The ECG toolkit has previously been tested to perform batch conversions to and from certain file formats [4]. The new ECG formats added to the ECG toolkit, additional testing was required for performance evaluation, but also to assess signal quality and archive size of the exported ECGs.

The exporting of resting ECGs to a PACS using the DICOM C-STORE protocol was also tested on multiple PACS, including THOPACS used in our hospital and two open source PACS. For this test, we sent 100,000 resting ECGs sequentially archived (sampled at 500 Hz, 12-bits ADC) across a network (100 Mbps), using: Conquest DICOM software [9] and DCMTK [10].

To estimate the relative size of each file we converted the same 100,000 resting ECGs to each of the open standards, the file size provides information about sending these files using low bandwidth connections.

To confirm whether the conversion works correctly we selected two sets of 100 ECGs. The first set was selected based on technology: 25 recorded with Mortara Portrait Cart, 25 recorded with Mortara ELI250 Cart, 25 stored at Infinity MegaCare and 25 stored at CardioControl Workstation. The second set consists out of five sets of 20 ECGs selected based on the following diagnoses: bradycardia, tachycardia, premature ventricular contraction, atrial fibrillation and myocardial infarction. All the ECGs from both sets were converted to each of the three supported open standards. The converted ECGs were subsequently compared to the original ECG.

## 3. Results

To provide an overview of the effect on storage and transfer of an ECG recording we converted 100,000 resting ECGs to each of the supported open standard. For the conversion to SCP-ECG QRS subtraction was disabled. Table 1 provides an overview of each format.

Based on the values displayed in Table 1 a HL7 aECG or DICOM archive consumes at least 10 times as much space as a SCP-ECG archive. The 100,000 resting ECGs were also uploaded and stored in the Conquest and DCMTK DICOM PACS. The transmission of all ECGs

lasted 6 hours and 47 minutes for Conquest DICOM Software and 1 hour and 40 minutes for DCMTK. The total transmission size for all ECGs was 11.1 Gb.

Format	Filesize (Kb)	Transmission Time over GPRS (s)		Lines of Code
		56 kbit/s	114 kbit/s	
UNIPRO	6 ± 1	<1	< 1	6975
SCP-ECG	11 ± 1	2	< 1	6490
HL7 aECG	164 ± 8	24	12	5898
DICOM-ECG	119 ± 1	17	9	2193

Table 1. Average (± SD) filesize, minimal transmission time and lines of code for each ECG format

When sending an ECG recording using a mobile network (GPRS 114 kbit/s), the transmission time of a single resting ECG recording (8-10 seconds) would be at least 0.75 seconds for SCP-ECG versus at least 8.4 seconds for DICOM and 11.5 seconds for HL7 aECG.

All the ECGs of the two sets were thoroughly inspected for changes in each of the available viewers. To view the HL7 aECG files we used SigViewer part of BioSig [1]. We used a DICOM-ECG viewer for ECGs provided by Mortara Instrument to display our DICOM-ECG files. For SCP-ECG we used the SCP-AV [1]. Original printouts were compared with the printout of the DICOM viewer.

None of the ECGs showed any changes in the displayed information except for a change in the QTc value for 25 of the ECGs. The change in the QTc value is due to use of different algorithms by the manufacturers. To solve this issue the open standards do already allow the QTc value to be provided by manufacturers, but some manufacturers do not use all available elements of the open standard they support.

## 4. Discussion

By using the ECG toolkit all the resting ECGs recorded at the Thoraxcenter can be provided in three open standards: SCP-ECG, HL7 aECG and DICOM. To demonstrate the capabilities of the library included in the ECG toolkit we added example SCP-ECG to DICOM-ECG converter in the appendix.

The ECG toolkit is entirely available under the Apache License Version 2.0 [11]. The ECG toolkit incorporates all of the previously described elements except for the plug-in that allows us to access the Mortara Escribe management system. Figure 2 displays the ECG Viewer application that is part of the ECG toolkit.

Each of the three open standards enables a hospital to be partially or completely manufacturer independent for resting ECGs. Additionally adopting any of the three standards would allow exchange of ECGs with every hospital supporting that open standard.

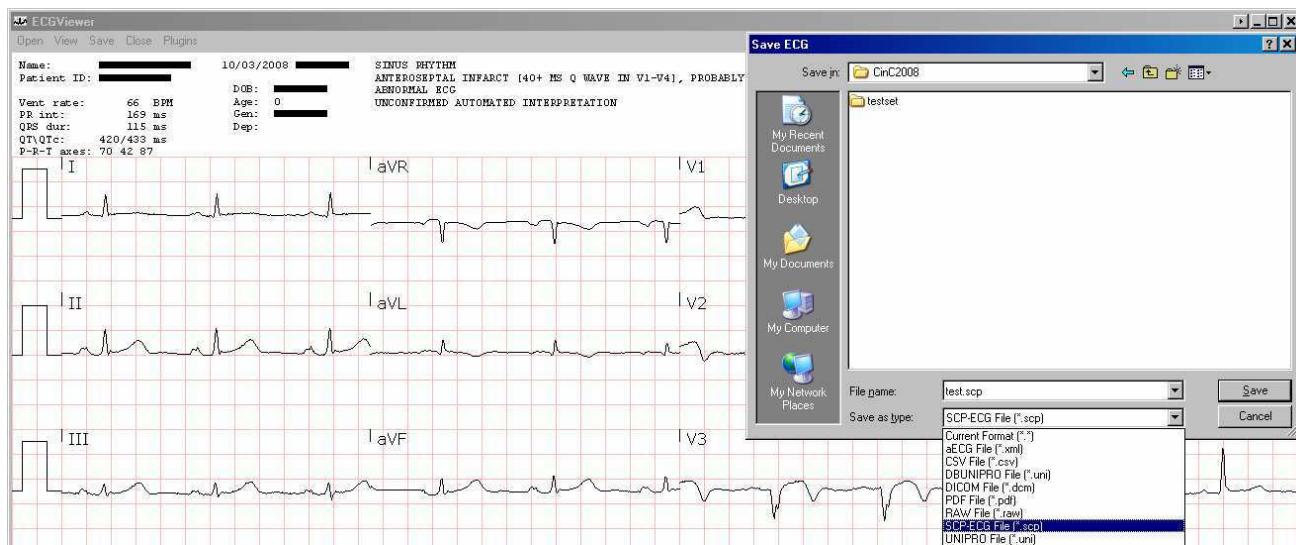


Figure 2. The ECG Viewer part of the ECG toolkit

The small size of SCP-ECG recordings also provide the smallest transmission time and is useful with a low bandwidth or high latency connection, as may be the case in rural (pre-)hospital settings. However, the SCP-ECG standard is complex and requires the most lines of code compared with HL7 aECG and DICOM-ECG.

HL7 aECG is the preferred standard for exchanging and comparing ECGs required by the FDA for drug trials. HL7 aECGs are XML textfiles and can be compressed by general purpose compression utilities, such as gzip or bzip, as well. We did not evaluate compressed HL7 aECG.

DICOM allows integration of ECGs and other modalities into the available enterprise imaging storage and retrieval systems, enabling simultaneous exchange of ECGs and medical images.

DICOM and SCP-ECG allow a manufacturer specific section, which could limit the independence of hospital when choosing new equipment. The private section could contain important information, which can only be accessed by the applications provided by the manufacturer.

Identifying a DICOM viewer able to display an ECG recording is still a challenge. Most DICOM viewers cannot display an ECG or do not display the ECG correctly. The DICOM viewer in use at the Thoraxcenter supports the resting ECG modality, but provides incorrect aspect ratio and does not provide a reference grid.

DICOM-ECG could adopt parts of the SCP-ECG standard similar to its adoption of JPEG2000. Adopting only the basic compression of SCP-ECG would be enough to reduce file size and subsequently transmission time specifically for low bandwidth connections.

Currently the three open standards each have their

advantages on specific application field. A single open standard for ECGs should still be preferable.

## Acknowledgements

The authors would like to thank J. de Bie and B. Brown for supplying us the DICOM-ECG viewer for ECGs developed by Mortara Instrument. The authors also thank Mr. McFadden from Draeger Medical.

## References

- [1] Schloegl A, Chiarugi F, Cervesato E, Apostolopoulos E, Chronaki CE. Two-Way Converter between the HL7 aECG and SCP-ECG Data Formats Using BioSig. Computers in Cardiology 2007;34:253-256.
- [2] 12-Lead Electrocardiogram Information Object Definition. In: DIGITAL IMAGING AND COMMUNICATIONS IN MEDICINE (DICOM) PART 3: INFORMATION OBJECT DEFINITIONS. National Electrical Manufacturers Association; 2007. p. 170-172.
- [3] Hilbel T, Brown BD, de Bie J, Lux RL, Katus HA. Innovation and Advantage of the DICOM ECG Standard for Viewing, Interchange and Permanent Archiving of the Diagnostic Electrocardiogram. Computers in Cardiology 2007;34:633-636.
- [4] de Wijs MCJ, van Ettinger M, Meij SH, Nelwan SP. Integration of multiple ECG databases into a unified framework, Computers in Cardiology 2005;32:447-450.
- [5] van Ettinger MJB. C# ECG Toolkit. Available from <http://ecgtoolkit-cs.sourceforge.net/>
- [6] Mono open source project: Mono. Available from: <http://www.mono-project.com/>
- [7] Yang F. DICOM#. SourceForge.net. Available from: <http://sourceforge.net/projects/dicom-cs/>
- [8] Lowagie B, Soares P. iTextSharp. SourceForge.net. Available from: <http://sourceforge.net/projects/itextsharp/>
- [9] van Herk M, Zijp L. Conquest DICOM software. Available from: <http://www.xs4all.nl/~ingenium/dicom.html>.

- |  |  |
|--|--|
| [10] OFFIS: DCMTK - DICOM Toolkit. Available from:<br>http://dicom.offis.de/dcmtk.php.en                               | Address for correspondence   |
| [11] The Apache Software Foundation: Apache License<br>Version 2.0. Available from:<br>http://www.apache.org/licenses/ | Maarten van Ettinger Jr.<br>Office BA569<br>Thoraxcentrum, Erasmus MC<br>Postbus 2040<br>3000 CA Rotterdam<br>m.vanettinger@erasmusmc.nl |

## Appendix: C# example code

The following example application that will convert a SCP-ECG file to an anonymous DICOM-ECG file

```

using System;
using ECGConversion;

namespace ECGExample {
    class ECGExample {
        [STAThread]
        static void Main(string[] args) {
            if (args.Length != 2) {
                Console.WriteLine("Usage: ECGExample scp-file output-file [-a]");
                return;
            }

            IECGReader reader = ECGConverter.Instance.getReader("SCP-ECG"); // get SCP-ECG reader
            if (reader == null) {
                Console.Error.WriteLine("Error: reading of SCP-ECG not supported!");
                return;
            }

            IECGFormat in_file = reader.Read(args[0]); // read SCP-ECG file.
            IECGFormat out_file = null;
            if (reader.getError() != 0) {
                Console.Error.WriteLine("Reading Error: {0}!", reader.getError());
                return;
            }

            // perform conversion to DICOM-ECG.
            if ((ECGConverter.Instance.Convert(in_file, "DICOM", out out_file) != 0)
                || (out_file == null)) {
                Console.Error.WriteLine("Error: Converting failed!");
                return;
            }

            out_file.Anonymous(); // make the ECG file anonymous

            ECGWriter.Write(out_file, args[1], true); // write DICOM-ECG file.
            if (ECGWriter.getLastErr() != 0) {
                Console.Error.WriteLine("Writing Error: {0}!", ECGWriter.getLastErr());
                return;
            }

            Console.WriteLine("Converting successful!");
        }
    }
}

```