Two-Way Converter between the HL7 aECG and SCP-ECG Data Formats Using BioSig

A Schloegl[1,2], F Chiarugi[3], E Cervesato[4], E Apostolopoulos[3,5], CE Chronaki[3]

[1]Fraunhofer FIRST, Berlin, Germany
[2]Institute for HCI, Graz University of Technology, Graz, Austria
[3]Institute of Computer Science, FORTH, Heraklion, Crete, Greece
[4]ARC-Cardiologia, Pordenone, Italy
[5]Computer Science Department, University of Crete, Heraklion, Crete, Greece

Abstract

This paper presents an effort launched in 2006 by the OpenECG network, led by the Graz University of Technology and supported by IEEE 1073, ISO 11073 and CEN TC251 to create a two-way converter in C++ between the SCP-ECG and the HL7 aECG standards. In the conversion, GDF, the BioSig internal data format, was used as an intermediate structure. This design approach allowed people with different expertise to be involved in the implementation of the converter. ECG data sets from the OpenECG portal were used to test the converter. However, some data mapping problems were identified. In fact, the SCP-ECG standard includes clinical data of the patient such as blood pressure, weight, and height, etc, which are not part of the HL7 aECG standard. Moreover, the annotations of HL7 aECG can be translated to GDF events, but, currently the only way SCP-ECG might support HL7 aECG annotations or GDF events is by using custom tags or sections.

The first version of the converter has been released in open source to be tested by the BioSig and OpenECG communities. Some data mapping issues remain open in this first release. However, it is the expectation of the OpenECG community that they will be addressed in the collaboration among the relevant Standard Developing Organizations as a critical step towards the improvement of interoperability in electrocardiography.

1. Introduction

BioSig [1] is an open source library for biomedical signal processing. Its creation was motivated by the need to develop new analysis methods for a variety of physiological signals including the electroencephalogram (EEG), the electrocardiogram (ECG), the electrooculogram (EOG), the electrocorticogram (ECoG), the electromyogram (EMG), and so on. One of the BioSig modules entitled "data formats and storage" was specifically designed to provide a common interface for accessing various data formats supporting automated format detection.

OpenECG [2] is a world-wide network supporting interoperability in electrocardiography through the consistent implementation of standards. In 2007 it has more than 850 members from 63 countries. OpenECG supports its members with information, news, converters and other services including an online interoperability certification service for the SCP-ECG standard, the European data standard for resting ECGs [3]. The development of open source converters among ECG data formats is supported and encouraged by OpenECG as the means to promote interoperability in electrocardiography.

HL7 [4] is an international organization developing healthcare standards for clinical and administrative data. The HL7 annotated ECG standard (HL7 aECG) [5] addressed the call of FDA for full disclosure in clinical trials with a flexible XML schema and a rich set of annotations. The HL7 aECG standard is of rising popularity. In the most recent version of SCP-ECG (EN1064:2005+A1:2007) the nomenclature of SCP-ECG has been harmonized with HL7 aECG.

The world-wide diffusion of different standards for ECG data storing can hamper the creation of full interoperable health information system able to integrate devices from different vendors. The correct implementation of a standard has often a significant cost in terms of required expertise and man-power, thus for health information providers the implementation of multiple standards can be unaffordable. The diffusion of open source converters among different data formats can facilitate interoperability. The availability of an open source two-way converter between the two most widely used standards for ECG data storage is a significant step towards the realization of multi-vendor, interoperable health information systems.

2. Methods

At the end of 2005 there was an open discussion in the digital ECG community on which interoperable standard for digital ECG an implementer had to support and to work with. The discussion was mainly conducted through the OpenECG community and the main available standards were the SCP-ECG and the HL7 aECG.

The result of the discussion was that the creation of an open source two-way converter between the two main standards would have provided an effective answer to the question and allowed each implementer, in principle, to work with a single standard and to integrate devices or information systems supporting the other standard using the open source converter.

Thus, at the beginning of 2006, an international working group formed by people with different expertise was created by the OpenECG network, with the support of IEEE 1073, ISO 11073 and CEN TC251 and the coordination of Graz University of Technology and BioSig, for the development of an open source two-way converter in C++ between the SCP-ECG and the HL7 aECG standards.

2.1. BioSig

The BioSig project [1] consists of several subprojects. The first and largest part is implemented in "BioSig for Octave and Matlab" (BioSig4OctMat). It includes filters for over 40 data formats, methods for artifact processing and quality control, methods for feature extraction, classification of EEG, QRS detectors and methods for HRV analysis, statistical analysis and visualization. Unfortunately, it lacks a sufficient XML support, as needed for HL7 aECG management.

In order to improve performance and to provide a "low-level" interface "BioSig for C/C++" (BioSig4C++) was established. The present converter has been implemented within this subproject.

2.2. Design of the converter

Based on the experience of 40 different data formats, the "*General Data Format for biomedical signals*" (GDF) [6], a common internal data structure (Table 1), was established, and reading and writing functions were provided in order to convert the internal data structure to and from the various data formats. This has the advantage that for M data formats only 2*M instead of M*(M-1) converters need to be implemented.

The internal data structure provides a data buffer for the raw data (as specified by the respective format specification), the internal data matrix (each column contains the samples of one channel), fields for the event table (using predefined event codes), internal fields supporting the conversion between raw data and the data matrix, and some control flags (e.g. for scaling and automated overflow detection).

Table 1: Fields of the internal data structure of "BioSig for C/C++".

HDR variable	*Description*
HDR-Patient.{Name, Sex, Id, Birthday, Weight, Height}	Demographic information (like gender, age, weight, height, etc.)
HDR.T0	Starting time of recording
HDR.NS	Number of channels
HDR.SampleRate	Common sampling rate (least common multiple of the sampling rates from each channel)
HDR.NRec	Number of data blocks
HDR-SPR	Samples per block (least common multiple of CHANNEL[k.].SPR)
HDR.data.block	Data matrix (common sampling rate and word length)
HDR.data.size[2]	Rows (HDR.NRec*HDR.SPR) and columns (HDR.NS) of data.block
HDR.CHANNEL[K].{Notch, LowPass, HighPass}	Filter settings of recording
HDR.CHANNEL[K].LowPass	Series control variable
HDR.CHANNEL[K].HighPass	Series control variable
HDR.CHANNEL[K].SPR	Samples per block of each channel
HDR.CHANNEL[K].Cal	Scaling (calibration) factor
HDR.CHANNEL[K].Off	Offset
HDR.CHANNEL[K].DigMin	Digital minimum
HDR.CHANNEL[K].DigMax	Digital maximum
HDR.CHANNEL[K].PhysMin	Physical minimum
HDR.CHANNEL[K].PhysMax	Physical maximum
HDR.CHANNEL[K].{PhysDim, PhysDimCode}	Physical units and its encoding according to ISO11073
HDR.CHANNEL[K].{Label, LeadIdCode}	Channel label and lead identification code according to SCP-ECG and HL7 aECG
HDR.AS.rawdata	Raw data representation (sampling rate and word length can differ between channels)
HDR.CHANNEL[k].GDFTYP	Data type used (e.g., 1: int8, 2: uint8, 3: int16, 4: unit16, 5: int32, 6: uint16, 16: float32, 17: double64)
HDR.AS.{bi, bpb, spb}	Support the conversion between raw data and data block
HDR.FLAG.UCAL	0: Calibrated (scaled), 1: Not calibrated
HDR.FLAG.OVERFLOWDETECTION	0: Off, 1: ON; samples exceeding PhysMin/Max are replaced by not-a-number (NaN)
HDR.EVENT	Event table
HDR.EVENT.N	Length of event table
HDR.EVENT.POS	Position of event (in samples)
HDR.EVENT.TYP	Type of event (according to the predefined table of event codes)
HDR.EVENT.DUR	Duration of event
HDR.EVENT.CHN	Event applies to this channel; 0 indicates all channels

Besides several utility functions, the most important functions, provided in the "data formats and storage" module of BioSig, are SOPEN and SCLOSE for reading and writing of the header information, and SREAD and SWRITE for data conversion between raw data and the data matrix. These functions provide a common interface for all supported data formats, and call internally the functions for the actual conversion of each data format. Special attention was paid to supporting 32 bit and 64 bit, big and little endian platforms. Moreover, the zlib library [7] is used for on-the-fly compression/decompression [8] of several data formats.

2.3. The SCP-ECG reader/writer

The SCP-ECG reader module was derived from another open source project [9,10]. Both high compression and redundancy reduction modalities are supported. Root mean squares and maximal absolute differences, between the reconstructed and the original signal, are below the limits of the SCP-ECG standard. Special effort was put into implementing the decimated sample reconstruction and the signal low-pass filtering, which are considered the highest critical aspect of SCP-ECG. The decoding algorithm is able to reconstruct different types of short-term ECGs, although intensive testing was only performed on 12 leads / 10 seconds resting ECG.

The SCP-ECG writer takes the information stored in the internal data format and produces an SCP-ECG file. In the present release none of the SCP-ECG compression methods (redundancy reduction or high compression) is supported but the samples are stored as 16-bit signed integer. This is the simplest signal storage allowed by the flexible SCP-ECG standard and does not require too much implementation effort. Optionally, on-the-fly compression using zlib can be used. The advantage of using compression methods mainly consists in the size of the produced SCP-ECG file. In our case, the size was around 80 kBytes while with high compression it is possible to have SCP-ECG files with a size of a few kBytes. The file size of the zlib-compression is usually in between.

2.4. The HL7 aECG reader/writer

For reading and writing XML data, the TinyXML toolbox [11] is used. The HL7 aECG reader module starts reading the ID and effectiveTime tags, which hold information for the recording. Then, it proceeds with the patient's data, which includes name, birthday and sex.

Afterwards, it reads from the series tag the values of the HighPass, LowPass and Notch filters. In order to allocate dynamically the necessary memory, the reader calculates beforehand the number of channels used in the recording. Then, after getting the sample rate, it reads for every channel the number of samples, the digital values of the signal, the calibration and offset factors for the scaling, and the corresponding physical dimension code. After evaluating the minimum and maximum digital values, the module calculates the physical minimum and maximum value using the corresponding digital values and the scaling factors. All information is used to fill in the internal data structure.

The HL7 aECG writer takes the information stored in the GDF internal data format and produces an HL7 aECG file in output. The structure of the output file is compliant to the XML schema for the HL7 aECG files.

3. Results

3.1. Testing

The converter was tested to work with the "OpenECG certified" testing data set available in the OpenECG site. Starting from these files, the sequence of conversion was SCP-ECG > GDF > HL7 aECG > GDF > SCP-ECG > GDF > HL7 aECG. Once obtained the converted HL7 aECG files, the second conversion (testing HL7 aECG files) used the sequence HL7 aECG > GDF > SCP-ECG > GDF > HL7 aECG. The converter was tested to work both in Windows (Cygwin) and Linux OS.

To check qualitatively if the result files were converted correctly two available viewers were used: the "SCPViewer" to visualize the SCP-ECG files and the "XMLFDA" to visualize the HL7 aECG files.

Furthermore, 48 SCP-ECG files (with different compression methods) and 35 HL7 aECG files were converted directly (SCP-ECG > GDF, HL7 aECG > GDF) and indirectly (SCP-ECG > HL7 aECG > GDF, HL7 aECG > SCP-ECG > GDF) to GDF. The resulting GDF files were loaded into Octave using "BioSig for Octave and Matlab". The relative differences in the root-mean squares between direct and indirect conversion was in the range of 1 to 3 times of the floating point accuracy, indicating practically no difference. This quantitative test was performed on three different platforms (i386, amd64 and ppc64) using Debian Linux; thus, big and little endian, 32 bit and 64 bit platforms were tested.

3.2. Open issues

There are some open issues in the conversion between HL7 aECG and SCP-ECG that need to be resolved in the future:

a) Multiple ECG recordings that may be included in one single HL7 aECG file are not supported. Only the first recording is converted.
b) HL7 aECG annotations are not supported yet. The support for annotations can be implemented using the event table as defined in [6], but, currently, the only way SCP-ECG might support HL7 aECG

annotations or GDF events is by using custom tags or sections.

c) Some SCP-ECG information (some patient and ecg acquisition data in section 1, the global measurements in section 7 and the interpretive statements in section 8-11) are not present in the HL7 aECG file structure.

The latter two points are beyond the scope of this project, but have to be addressed by the corresponding standardization organizations.

4. Discussion

There are several important aspects in this work like the use of open source design, the improvement in interoperability and standard harmonization and the extension of freely available software like the BioSig library and the OpenECG tools.

In fact, in the development of this converter the working group had necessarily to provide a lot of useful insights into each of the standards. Based on this work, one can see the strength and weaknesses of each standard and these considerations are available to each user that will use this open source software. Other data formats without an open source implementation often lack such a detailed public analysis. Moreover, this work not only improves the current status in interoperability in electrocardiography, but also provides the necessary information to the standardization organization in order to better harmonize the standards in a future perspective.

Finally, the design decision to use a common data representation has shown its advantages in two ways. First, the use of a common data structure allows an easy comparison of different data formats. Second, besides HL7 aECG, SCP-ECG and GDF, also other data formats are fully (EDF, BDF, CFWB) or partially (ACQ, BKR, CNT) supported. Support of other data formats (e.g. MFER, DICOM waveform supplement 3.0) can be also included with great advantage for the scientific community and implementers.

The current solution uses open source tools only, and is therefore a truly free and open source solution. The authors hope this will facilitate further contributions from domain experts towards a more complete version to support the vision of interoperable Electronic Health Records able to include all ECGs of a person in a single standard format.

5. Conclusions

Bidirectional converters between the SCP-ECG, HL7 aECG, and GDF data formats have been implemented and tested. The software is free/libre open source software licensed with the GPL [12]. The software is available from the BioSig [1] and the OpenECG [2] sites.

Acknowledgements

This research work was supported by EU Grant BrainCom FP6-024259 and by the EU FP6 IST-ICT for Health, within the STREP project "HEARTFAID: a Knowledge based Platform of Services for supporting Medical-Clinical Management of the Heart Failure within the Elderly Population" (IST-2005-027107), 2006-2009.

References

[1] The open source software library for biomedical signal processing. http://biosig.sourceforge.net/.

[2] The OpenECG portal. http://www.openecg.net/.

[3] Health informatics – Standard communication protocol – Computer-assisted electrocardiography. ICS: 35.240.80 IT applications in health care technology; reference number EN 1064:2005+A1:2007.

[4] Health Level 7. http://www.hl7.org/.

[5] HL7 Annotated ECG. http://www.hl7.org/V3AnnECG/sectioncontent/ZI/HMPOZI_SB_RM.htm#POZI_RM020001-rmi.

[6] GDF 2.0: A General data format for biomedical signals. http://arxiv.org/abs/cs.DB/0608052.

[7] Zlib: A Massively Spiffy Yet Delicately Unobtrusive Compression Library. http://zlib.net/.

[8] RFC 1950-1952, IETF, 1996.

[9] E. Cervesato, G. De Odorico, A. Accardo, P. Pascolo, F. Antonini-Canterin, G.L. Nicolosi. Is the SCP-ECG format suitable for inpatient ECG management? IEEE Computers in Cardiology 2003; 30: 375-378.

[10] E. Cervesato, G. De Odorico. Standard Communications Protocol for computer-assisted electrocardiography advanced viewer. Proceedings of the 2nd OpenECG workshop 2004, Berlin, Germany; p. 18-19.

[11] The TinyXML Toolbox. http://tinyxml.sourceforge.net/.

[12] GNU General Public License. http://www.gnu.org/copyleft/gpl.html.

Address for correspondence

Catherine E Chronaki
Institute of Computer Science, FORTH
PO 1385 Vassilika Vouton
Science & Technology Park of Crete
71110, Heraklion, Crete, Greece

E-mail address: chronaki@ics.forth.gr