Methods and Tools for Generating and Managing ecgML-Based Information

H Wang[1], F Azuaje[1], G Clifford[2], B Jung[3], N Black[1]

[1]University of Ulster , Newtownabbey, N Ireland, UK
[2]Massachusetts Institute of Technology, Cambridge, MA, USA
[3]University of Victoria, Victoria, BC, Canada

Abstract

There is a need to develop methods to harmonize electrocardiogram data representation and management. An extensible markup language vocabulary for electrocardiogram data encoding and analysis, ecgML, illustrates an alternative to support intelligent and system-independent electrocardiogram information management. In order to facilitate the development of ecgML-based applications, this paper presents tools for managing this type of information. Emphasis has been placed on the design of the ecgMLgenerator and ecgMLbrowser. An information visualization application, which automatically transforms data from the MIT-BIH Arrhythmia Database into ecgML format, is discussed. Recommendations on future research directions to support XML-based ECG information integration are provided. The tools are freely available on request from the authors.

1. Introduction

The prevalence of multiple electrocardiogram (ECG) data formats and devices, together with the variety of architectures and platforms for storing, reading and viewing ECG data that are currently in operation motivates the need for a unified, platform- and device-independent approach to such tasks. Attempts to address this problem include the *Standard Communications Protocol for Computer-Assisted Electrocardiography* (SCP-ECG) [1], proposed by a CEN/TC251 (Comité Européen de Normalisation Technical Committee 251) project team in 1993. This standard allows the representation of the ECG waveforms, patient demographics, as well as measurement and interpretation results. Although this protocol is well-documented and supported by a professional community (the OpenECG group), the utilization of SCP-ECG has demonstrated some disadvantages. For example, it concentrates on 12-lead ECG and only has limited provision for annotation. There seems to be little evidence to indicate that this protocol has been widely incorporated into commercial and academic products [2]

Since 1999 PhysioNet [3] has provided an on-line forum for dissemination and exchange of biomedical signals including ECG data, stored in the WFDB (waveform database) format. It currently contains over 1000 recordings organized in more than 30 databases, in which almost all records include a binary signal file. This may represent a highly limited solution for advanced ECG management, as it lacks referencing and vocabulary control as well as an easily accessible and extensible data format.

Digital Imaging and Communications in Medicine (DICOM) 3.0 Supplement 30 is another example of ECG data interchange formats [4]. Due to popular demand from the ECG community, the *DICOM cardiovascular working group* added the ability to encode ECG, electrophysiological and haemodynamic curve data to the DICOM standard in 2000. It provides "*Waveform Object Definitions"* for general ECG, ambulatory ECG, and 12-lead ECG. This data representation format aims to achieve a robust exchange of waveform and related data within the DICOM environment.

More recently, the *Food and Drug Administration agency* (FDA), together with a number of other organizations such as *Heath Level 7* (HL7), initiated extensive discussions in connection to an *eXtensible Markup Language* (XML)-based ECG representation. In December 2003, the *FDA XML* format, which is based on the HL7 version 3 messaging standard, was finalized and the Annotated ECG (aECG) format is now part of the HL7 family of standards [5]. The relevant XML Schema can be derived from the HL7 *Refined Message Information Model* (R-MIM).

In order to promote these approaches and improve ECG interoperability, it is crucial to develop open, user-friendly tools to manage data representation formats and protocols. The lack of publicly-available tools is one of the main constraints to the deployment and assessment of potential standards. To cope with this problem and promote the SCP-ECG protocol, the OpenECG

0276−6547/04 $20.00 © 2004 IEEE

Consortium announced in 2003 a programming contest for electrocardiography applications and tools using the SCP-ECG Standard [6]. Similar efforts have been supported by other initiatives. An ECG viewer for the XML FDA format, for example, is freely available [7]. The WFDB package, which includes a collection of software tools for viewing, analyzing, and creating of WFDB format compatible data, is frequently updated and freely available.

ecgML, a markup language for ECG data acquisition and analysis, was proposed as an inexpensive alternative to existing data formats for achieving structured and meaningful ECG data representation [8]. In order to assist users in exploiting ecgML-based applications, this paper presents methods and tools for generating and managing ecgML-based information. By way of an illustration, Section III discusses an information visualisation application whereby data from the MIT-BIH Arrhythmia Database [3] is automatically converted into the ecgML format. Advantages and future development of this data format are included in the last section.

2. Methods

2.1. The ecgML Model

ecgML [8] is a flexible approach to representing, exchanging, and mining ECG data. The specification is encoded as an XML-vocabulary. Its hierarchical structure for the representation and storage of ECG data has been synthesized from existing recommendations and formats such as SCP-ECG. A full description of ecgML can be found at http://ijsr32.infj.ulst.ac.uk/~e10110731/ecgML/.

Based on XML technology, ecgML offers several advantages over existing systems [8], [9]. Unlike SCP-ECG, which was developed exclusively for 12-lead ECGs, ecgML supports the full spectrum of ECG data and allows multiple time-related ECGs to be kept in one single record file. It is self-explanatory. An ecgML-based record does not only incorporate the relevant waveform data but also their description, leading to a both human- and machine- readable representation. Moreover, it is extensible, i.e. the record structure can be adapted to future requirements and additional information added. Unlike ecgML, both DICOM 3.0 supplement 30 and FDA XML format are designed to meet the needs of their specific users and application environments (DICOM and HL7). XML elements of the FDA XML format must be derived from the HL7 R-MIM vocabulary and codes, resulting in a less readable format due to proprietary encoding schemes. It is evident that the application of such a format requires at least an adequate understanding of the HL7 philosophy.

2.2. Fundamental ecgML tools

A series of tools have been developed to assist (technical and medical) personnel in the development of ecgML-based applications. ecgMLgenerator and ecgMLbrowser are Java-based, platform-independent tools, which are introduced as follows.

2.2.1. ecgMLgenerator

The ecgMLgenerator provides a user-friendly interface for automatically creating ecgML-based records. Working in conjunction with the Oracle XML Parser, Oracle Class Generator generates a set of Java classes (one class per XML element) based on either the ecgML DTD or XML Schema. The generated classes are then used to dynamically construct an ecgML-based representation, which is compliant to the DTD/XML Schema specified.

Given the vast amount of ECG data available in different formats, it is an important requirement that the ecgMLgenerator automatically creates ecgML-based records using data from existing ECG databases. The current version of the ecgMLgenerator includes a first module to create ecgML files from the MIT-BIH Arrhythmia Database. Based on a series of Java classes dedicated to parse header, annotation and signal files for a selected record, the tool automatically extracts relevant features from the selected record and pipes the results directly into the ecgMLgenerator. An overview of how the ecgMLgenerator constructs an ecgML document from the MIT-BIH database is illustrated in Figure 1.

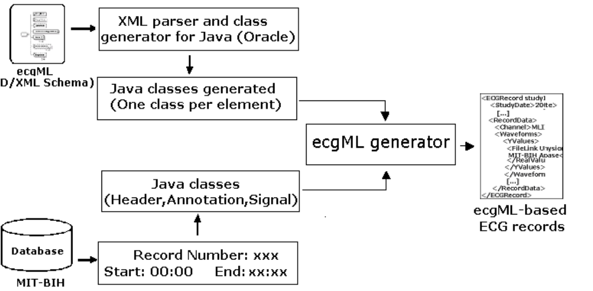


Figure 1. An overview of how the ecgMLgenerator constructs an ecgML record from MIT-BIH database.

2.2.2. ecgMLbrowser

The ecgMLbrowser application provides an on-screen display of ecgML records, including waveform and annotation data. Using Oracle SAX parser, an ecgML document is broken up into a series of chronological parsing events. For the purpose of mirroring the hierarchical nature of the data, an in-memory tree representation of the data is constructed using a stack-based algorithm [10], which transforms linear events into a tree structure, as illustrated in Figure 2.

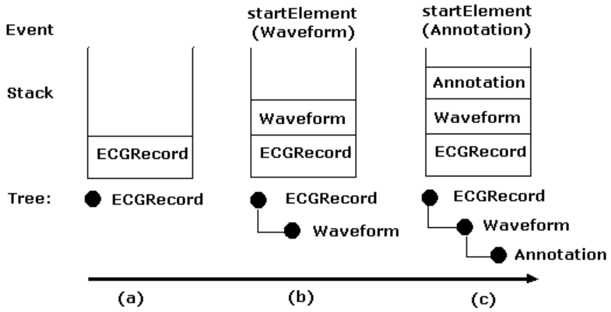


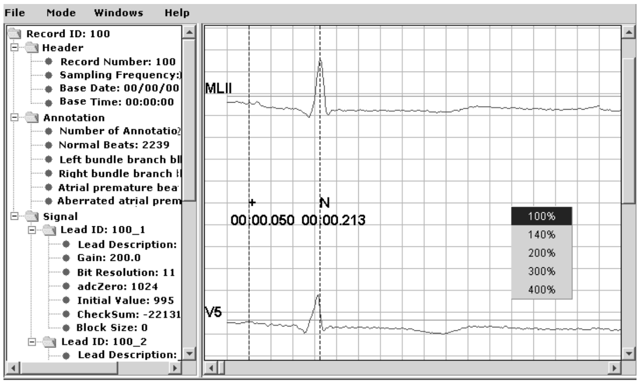
Figure 2. A stack-based algorithm: (a) pushing an ECGRecord (root node) into the stack. (b) Receiving a start element event for Waveform. (c) Receiving a start element event for Annotation.

3. Results

By way of illustration, the following results are based on the processing of the MIT-BIH Arrhythmia Database [3], which is a key reference for the biomedical research community.

3.1. ecgMLgenerator

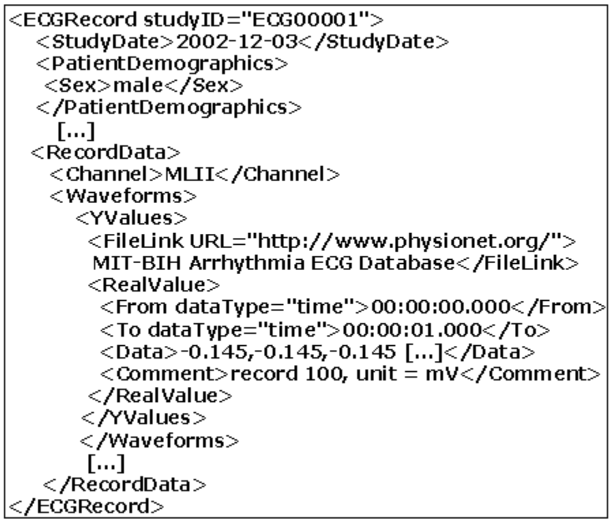
To facilitate the automatic creation of ecgML records from the MIT-BIH database, a Java-based waveform viewer was first developed as shown in Figure 3. Relevant information extracted from header and annotation files is displayed on the left hand side, while the ECG waveform, together with each beat annotation, is rendered in the right panel.



```
Record ID: 100
  Header
    Record Number: 100
    Sampling Frequency:
    Base Date: 00/00/00
    Base Time: 00:00:00
  Annotation
    Number of Annotatio
    Normal Beats: 2239
    Left bundle branch bl
    Right bundle branch
    Atrial premature bea
    Aberrated atrial prem
  Signal
    Lead ID: 100_1
      Lead Description:
      Gain: 200.0
      Bit Resolution: 11
      adcZero: 1024
      Initial Value: 995
      CheckSum: -22131
      Block Size: 0
    Lead ID: 100_2
      Lead Description:
```

Figure 3. A Java-based waveform viewer for MIT-BIH Arrhythmia Database

After visualizing the raw waveform data, users can select "*Convert to ecgML*" from the *Mode* menu to convert the selected waveform data into an ecgML-based representation. Figure 4 depicts a portion of an ecgML record generated in this application.

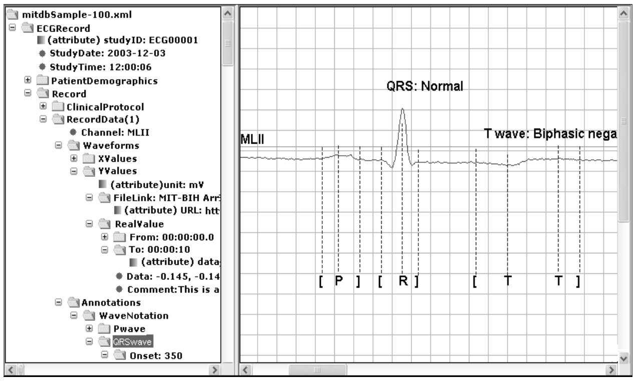


```
<ECGRecord studyID="ECG00001">
  <StudyDate>2002-12-03</StudyDate>
  <PatientDemographics>
   <Sex>male</Sex>
  </PatientDemographics>
    [...]
 <RecordData>
   <Channel>MLII</Channel>
   <Waveforms>
      <YValues>
       <FileLink URL="http://www.physionet.org/">
        MIT-BIH Arrhythmia ECG Database</FileLink>
       <RealValue>
        <From dataType="time">00:00:00.000</From>
        <To dataType="time">00:00:01.000</To>
        <Data>-0.145,-0.145,-0.145 [...]</Data>
        <Comment>record 100, unit = mV</Comment>
       </RealValue>
      </YValues>
      </Waveforms>
       [...]
   </RecordData>
</ECGRecord>
```

Figure 4. A portion of an ecgMLrecord created by ecgMLgenerator.

3.2. ecgMLbrowser

The ecgMLbrowser, as shown in Figure 5, includes a panel dedicated to a tree navigational display, which highlights the hierarchical structure of the ecgML model. It also includes a panel that provides browsing capability to display waveform data and annotations. The user has options to zoom in/out and to turn on/off annotations by type. In this version, we incorporated *ecgpuwave* software provided by PhysioNet [3] to detect the QRS complexes and locate the onset, peak, and offset of the P, QRS, and ST-T waveforms. By default, the browser uses the waveform data directly encoded in ecgML (in *RealValue* element). Alternatively, users can click on the *FileLink* element to dynamically link to the raw binary data.



```
mitdbSample-100.xml
  ECGRecord
    (attribute) studyID: ECG00001
    StudyDate: 2003-12-03
    StudyTime: 12:00:06
    PatientDemographics
    Record
      ClinicalProtocol
      RecordData(1)
        Channel: MLII
        Waveforms
          XValues
          YValues
            (attribute)unit: mV
            FileLink: MIT-BIH Arr
              (attribute) URL: htt
            RealValue
              From: 00:00:00.0
              To: 00:00:10
                (attribute) data
              Data: -0.145, -0.14
              Comment:This is a
        Annotations
          WaveNotation
            Pwave
            QRSwave
              Onset: 350
```

Figure 5. A screenshot of ecgMLbrowser. "[" represents wave onsets; "]" refers to wave offsets. Notations P, R and T stand for P, R and T waves.

4. Discussion and conclusions

The advantages of representing ECG with ecgML is the same as the advantages of XML itself; an almost endless set of ecgML DTD-compatible extensions can be added to the data without affecting legacy code. Both data and applications can be built to utilize only the parts of the data structure that are relevant to their project, and to add personal annotations which can be ignored by other users. The potential to encode parallel annotations will facilitate multiple-expert scorings and allow the aggregation of algorithmic outputs more easily. An ecgML-based data format also provides the ability to integrate alternative formatted files and algorithm outputs both locally and across networks. The ability to embed a virtually infinite number of URLs in the data format allows easy integration with other relevant clinical information such as electronic patient records and related medical data [11].

The ecgMLbrowser and ecgMLgenerator are key tools for facilitating the application of ecgML and the integration of different ECG sources. These tools are freely available on request from the authors. We aim to apply an open source approach to its development.

Given the diversity of ECG representation formats and the fact that there is not clear consensus as to whether any specific encoding format should be adopted, it is important to develop data format- as well as platform-independent tools for ECG information representation. One possible solution is to use a plugin-library structure for reading different formats, which may hide the differences of the data formats behind a common application interface. Schneider [12] has provided relevant examples, by developing freely available libraries (libRASCH) for several medical information formats, including WFDB [3] compatible formats. Integration of such a framework into the ecgML model is an important task of future research.

An important problem in relation to an XML-based representation is that the data are severely uncompressed. It contains redundant information such as opening and closing tags. Thus, large resulting files would need to be zipped/unzipped on the fly [13]. For example, inserting the complete wave amplitude data of Record 100 taken from the MIT-BIH Arrhythmia Database into one single ecgML record, the size of the file included more than 15 megabytes (the corresponding data as a binary file encoded in WFDB is about 1.9 megabytes). After compressing with Gzip, the file size is reduced to approximately 3 megabytes. However, we argue that, as discussed above, ecgML provides an efficient and flexible way to represent ECG data. In the case where large ECG records are produced, such as Holter recording, the waveform data should be kept in an external file and a link pointing to this file should be provided.

The data and meta-information encoded in ecgML, such as *Diagnosis* and *Measurements*, may be useful to improve knowledge discovery and decision support. An ecgML-based database using data originating from *PhysioBank* [3] is under development using the tools discussed above. We aim to study a framework for incorporating ecgML meta-data to support pattern discovery and visualization.

References

[1] ENV 1064 standard communications protocol for computer-assisted electrocardiography. European Committee for Standardization, Brussels, Belgium, 1996.

[2] Clunie D A. Extension of an open source DICOM toolkit to support SCP-ECG waveforms, in proceedings of 2nd OpenECG Workshop 2004, Berlin, Germany, pp. 20-21.

[3] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101(23):e215-e220 [Circulation Electronic Pages; http://circ.ahajournals.org/ cgi/ content/ full/101/23/e215]; 2000 (June 13)

[4] DICOM Suppl. 30, Waveform interchange, Nat. Elect. Manufacturers Assoc.: ARC-NEMA, Digital Imaging and Communications, NEMA, Washington D. C. , 1999.

[5] Health Level Seven Version 3 Standard Ballot Package,, [30 July 2004] Available at: http://www.hl7.org/v3annecg/.

[6] Welcome to the OpenECG Portal, [30 July 2004] Available at: http://www.openecg.net/.

[7] Digital ECG XML Viewer, [30 July 2004] Available at: http://www.ert.com/products/xml_viewer.htm.

[8] Wang H, Azuaje F, Jung B, and Black N. A markup language electrocardiogram data acquisition and analysis (ecgML). BMC Medical Informatics and Decision Support 2003, 3(4).

[9] Wang H, Jung B, Azuaje F, and Black N. “ecgML: tools and technologies for multimedia ECG representation”, in proceedings of XML Europe 2003 conference, London, England, United Kingdom, 2003.

[10] Laurent S, Cerami E. Building XML Applications. USA: McGraw-Hill, 1999.

[11] Synapses Homepage, [13 September 2004] Available at: http://www.cs.tcd.ie/synapses/public/.

[12] Schneider R, LibRASCH, [30 July 2004] Available at: http://www.librasch.org/librasch/.

[13] Neumüller M. Compression of XML data. M.S. thesis, University of Strathelyde, 2001.

Address for correspondence.

Name: Haiying Wang
Full postal address: 16J27, School of Computing and Mathematics, University of Ulster at Jordanstown, Shore road, BT37 0QB, UK
E-mail address: hy.wang@ulster.ac.uk