

바이너리 심전도 파일 포맷의 효율적인 디스크립션을 위한 시각적 도구 설계

Design of Visual Tool for Efficient Describing of Binary ECG File Formats

구흥서

Heung-Seo Koo

청주대학교 컴퓨터정보공학과

요 약

본 연구에서는 BED(Binary ECG Description)기반의 바이너리 심전도(ECG) 변환시스템은 ECG 데이터의 상호연동성을 지원하기 위해 바이너리 ECG 파일을 XML기반의 HL7 aECG로 변환하는 시스템이다. 그러나, BED 시스템을 처음 사용하는 초보자는 XML을 사용하여 BED 문서를 작성하기 어렵다. 본 논문에서는 초보자들이 BED 문서를 쉽게 작성할 수 있도록 지원하는 시각적 도구인 BED 스튜디오를 구현한다. 이 시스템은 세부분, 즉 비주얼 편집기, XML트리 편집기, 그리고 포맷검증 뷰어로 구성된다. 포맷검증뷰어는 사용자가 비주얼 인터페이스를 사용하여 작성한 BED 문서를 바이너리 ECG 파일과의 포맷 매칭을 시각적으로 검사할 수 있도록 지원하여 BED 문서 작성시 발생할 수 있는 오류 가능성을 감소시켰다.

키워드 : 심전도, HL7 aECG, 바이너리 파일변환, 비주얼도구.

Abstract

In this study, A BED(Binary ECG description) based binary ECG conversion system is that supports the conversion of a binary ECG format into XML-based HL7 aECG for interoperability of ECG. HL7 aECG is a XML based standard for interoperability of ECG waveform. However, it is difficult for beginners to write the BED document of binary BED conversion system in XML. We implement a BED Studio on the Java Servlet engine that allows beginners to write BED documents more easily. Our system consists of three parts: Visual Editor, Text Editor, and Format Checking Viewer. Format Checking Viewer support users to detect the format errors in the XML files, so-called BED documents, that describe the data format of the targer binary ECG file in the BED-based binary ECG conversion system, so may reduce the format errors in BED documents.

Key Words : ECG, HL7 aECG, Binary file conversion, Visual tool.

1. 서 론

정보통신기술의 발전으로 유비쿼터스 헬스케어(ubiquitous healthcare, 이후 u-헬스케어)에 대한 연구가 최근 활발하게 진행되고 있다. u-헬스케어를 실현하기 위해서는 심전도, 혈압, 맥박, 심음 등의 주요 생체신호에 대한 측정기술과 수집된 생체신호의 효율적인 관리기술이 요구된다.

심전도(ECG) 검사는 적은 비용으로 심장질환을 조기에 식별할 수 있는 매우 중요한 비침습적(noninvasive) 진단방법 중 하나로[1], 가장 흔히 사용되는 심장진단 방법이지만 기존의 ECG 계측기들은 각자 고유한 파일포맷을 사용하여 ECG 데이터를 획득·저장·분석하고 있다. 이러한 이유로 동일한 계측기를 쓰지 않으면 ECG 데이터를 다른 의료정보시스템의 응용프로그램으로 전송하거나 재사용할 수 없다[2]. 따라서 의료정보시스템 간에 ECG 데이터의 교환을 거의 불가능하게 만드는 상호호환성(interoperability)을 저해하는 가장 큰 요인은 다양하게 존재하는 ECG 파일포맷들이다. 의료

정보시스템 관점에서는 각자의 독자적인 포맷을 지원하는 다양한 계측기들의 ECG를 통합관리하는 것이 매우 어렵고 시간의 소모되는 작업이다[3].

이러한 상호호환성 문제를 해결하기 위해 본 연구팀은 [4,5]에서 BED(Binary ECG Description) 방식으로 기존의 심전도 계측기로부터 수집된 바이너리 ECG 데이터를 북미 표준(de facto standard)인 HL7 aECG(Health Level 7 annotated ECG)[6] 포맷으로 변환하는 시스템을 제시하였다. 이 변환방식은 XML을 이용하여 목표 바이너리 ECG 파일 포맷을 미리 기술해 놓은 BED 파일을 이용한다. 그래서 추후 ECG 계측기의 개선으로 인해 ECG 파일포맷이 변경되어도 텍스트 편집기를 사용하여 BED 문서만을 간단히 수정하면 새로운 포맷의 바이너리 ECG 파일을 HL7 aECG로 변환할 수 있다는 장점을 가진다. 다시 말해서, BED기반의 변환시스템의 소스코드를 수정하지 않아도 된다.

그러나, BED 문서를 작성하려면 어느 정도 수준의 XML 관련 기반 지식을 필요로 하기 때문에 초보자들이나 XML에 대한 지식이 없는 의료정보시스템의 관리자들의 경우 BED 문서 작성시 어려움에 직면하게 된다. 또한 XML에 익숙한 사용자라고 하더라도 BED 문서를 작성하려면 두가지 지식

접수일자 : 2006년 11월 20일

완료일자 : 2007년 2월 1일

을 필요로 한다. 하나는 목표 바이너리 ECG 파일포맷에 관한 것이고, 다른 하나는 XML을 기반으로 하는 BED 언어와 BED 문서의 구조에 관한 지식이다. 그리고 바이너리 ECG 파일포맷을 기술하는 작업은 난이도가 높은 것은 아니지만 각 엘리먼트의 데이터타입과 데이터크기를 기술해야 하기 때문에 세심한 주의를 필요로 한다. 따라서 XML에 익숙한 사용자라고 하더라도 목표 바이너리 ECG 파일포맷과 BED 언어 및 문서의 구조를 고려하면서 텍스트 편집기를 사용하여 BED 문서를 작성하는 일은 시간소모적이고 오류발생 가능성이 높은 작업이다.

본 논문은 BED기반의 바이너리 ECG 변환시스템에서 사용자가 XML기반의 BED 문서를 작성할 때 쉽게 작성할 수 있도록 시각적 개발환경을 지원하는 BED 스튜디오(BED studio)의 설계에 대해 연구이다. 이 연구결과는 BED기반의 ECG 변환시스템에 익숙하지 않은 사용자들이 BED 문서 작성시 코딩과 디버깅 시간을 절감하여 생산성을 향상시키고, 시각적 개발환경을 통해 사용자 편의성을 증대시킬 수 있을 것으로 기대된다. BED 스튜디오는 BED 문서 작성시 필요한 기능과 작성 프로세스의 연관성을 고려하여 비주얼 편집 기능을 제공하는 비주얼 편집기, XML에 대한 높은 수준의 지식을 가지고 있는 사용자를 위한 XML트리 편집기, 그리고 생성된 BED 문서와 목표 바이너리 ECG 파일간의 정확한 엘리먼트 매칭에 대한 시각적인 검증환경을 제공하고 오류를 디버깅할 수 있는 포백검증 뷰어로 구성된다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2장은 본 논문을 이해하는데 필요한 XML기반의 BED 관련 배경지식을 기술하고, 3장에서는 BED 스튜디오의 설계에 대해 설명하며, 4장에서는 구현 및 사용자 인터페이스 부분을 설명한다. 그리고, 6장에서 결론을 맺는다.

2. 배경지식

BED기반의 바이너리 ECG 변환시스템은 ECG 데이터 변환요청이 발생하면, 시스템관리자가 사전에 XML로 바이너리 ECG 파일포맷을 기술한 BED 문서를 참조하여 해당 바이너리 ECG 파일을 XML기반의 HL7 aECG 포맷으로 변환하는 바이너리 파일포맷 독립적인 변환방식을 사용한다. 이 장에서는 BED기반의 ECG 변환시스템의 핵심요소인 BED관련 요소들에 대해 살펴본다.

2.1 BED의 개요

BED 언어는 바이너리 ECG 파일포맷을 기술하기 위한 XML기반의 언어이며, 이 언어를 사용하여 변환하고자 하는 목표 바이너리 ECG 파일포맷을 BED 문서라고 하는 별도의 텍스트 파일에 기술한다. 이 접근방법의 장점은 ECG 계측기가 지속적으로 개선(version-up)되면서 바이너리 ECG 파일포맷이 변경되더라도 ECG 변환 프로그램을 재작성 하지 않아도 된다는 점이다. 포맷이 변경되는 경우는 변환시스템의 소스코드를 수정함이 없이 바이너리 ECG 파일에 대한 해당 BED 문서만 수정하면 되므로 시스템의 유지보수 비용이 절감되는 효과가 있다.

2.2 BED 언어

그림 1은 BED 언어의 구성요소와 그들간의 관계를 나타낸 것이다. 그림에서 사각형박스는 BED 언어의 엘리먼트를 나타내며 사각형박스의 윗부분은 엘리먼트 이름, 아랫부분은

엘리먼트가 가질 수 있는 속성을 나타낸다. 또한 엘리먼트 간에는 1:1 또는 1:N의 관계를 가진다.

“bed” 엘리먼트는 루트 엘리먼트로서 BED 문서의 이름과 구조, 바이너리 ECG 파일의 확장자 정보가 저장된다. 그 하위의 첫 번째 엘리먼트인 “variables”에는 바이너리 ECG를 HL7 aECG로 변환할 때 필요한 값이 저장되며, 여기에 저장된 값은 “integer” 타입의 formula 속성에서 참조할 수 있다. 두 번째 엘리먼트인 “header”에는 바이너리 ECG 파일의 메타정보가 저장되며, 세 번째인 “data”는 ECG 파형정보가 저장되는 엘리먼트이다.

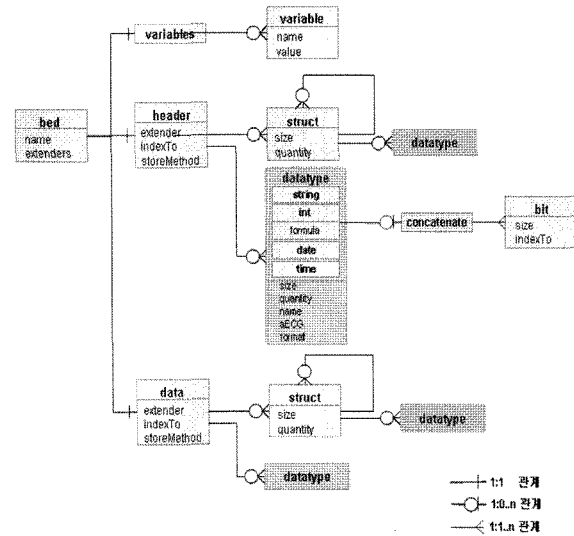


그림 1. BED 언어의 구성요소와 관계

Fig. 1. Elements of BED language and its relationships

2.3 BED문서

BED 문서는 BED 언어를 사용하여 바이너리 ECG 파일포맷을 기술한 XML 문서이다. 그러므로 BED 문서는 ECG 파일을 구성하는 데이터 구성요소에 대한 메타정보를 포함한다. BED 문서는 하나의 ECG 파일포맷에 대해 한 개가 존재하며, BED 변환시스템에서는 여러 개의 BED 문서를 포함할 수 있다. 즉, 여러 버전의 바이너리 ECG 포맷을 동시에 HL7 aECG 포맷으로 변환할 수 있다. 또한 BED 문서의 추가·수정·삭제가 용이하도록 BED 문서를 플러그엔플레이 방식으로 인식하게 설계하였다.

BED 문서는 그림 2에 나타난 것처럼 상수 선언부, 헤더 선언부, 시그널 선언부로 구성된다. 상수 선언부는 바이너리 ECG 파일에는 명시적으로 기술되지 않았지만 시그널 데이터에 관한 의미정보를 기술하는 부분이다. 헤더 선언부는 HL7 aECG 생성시 필요한 메타정보와 시그널 선언부에서 필요한 정보를 기술하는 부분이다. 바이너리 ECG의 헤더 데이터와 시그널 데이터가 하나의 물리적 파일에 존재하는 경우와 분리된 두 개의 파일에 존재하는 경우를 모두 허용한다. 그림 2는 MIT-BIH[7] 포맷같이 두개 파일로 분리된 바이너리 ECG 포맷에 대한 BED 문서의 예이다. 마지막으로 시그널 선언부에서 ECG 파형 데이터의 포맷을 기술한다. 대부분 시그널 데이터는 단순한 반복구조를 가지므로 BED 언어에서 “datatype” 엘리먼트의 “quantity” 속성을 이용하면 이러한 반복구조를 BED 문서에 간단하게 표현할 수 있다.

```

<?xml version="1.0" encoding="ascii"?>
<bed name="Example ECG Format" extenders="hea.dat">
  <!-- 선행부 -->
  <variable name="movement" value="0.002"/>
  <variable name="scale" value="5"/>
  </variable>
  <!-- 헤더 선행부 -->
  <header extender="hea" index="0" storeMethod="ascii">
    <struct size="-">
      <string name="recordName" size="-"/>
      <string name="numberOfSegments" size="-"/>
    </struct>
    <string name="numberOfSignal" size="-"/>
    <string name="ADGain" size="-"/>
    <string name="ADZero" size="-"/>
    <string name="numberOfSamplePerSignal" size="-"/>
    <string name="baseRate" format="HH:MM:SS" size="-" aECG="effectiveLine.Date"/>
    <date name="baseRate" format="DD/MM/YYYY" size="-" aECG="effectiveLine.Date"/>
  </header>
  <!-- 시그널 선행부 -->
  <data extender="dat" index="0" storeMethod="littleEndian">
    <struct size="4" quantity="{number of SamplePerSignal}">
      <int aECG="channel1" size="2" formula="{(self)-ADZero}/{ADGain}/>
      <int aECG="channel2" size="2" formula="{(self)-ADZero}/{ADGain}/>
    </struct>
  </data>
</bed>
  
```

그림 2. BED 문서의 3가지 영역
Fig. 2. Three Sections in a BED document

2.4 BED 변환시스템의 구조

BED 변환시스템은 그림 3과 같이 3개 주요 모듈로 구성되어 있다. 읽기모듈(Reading Module)의 BED 인터프리터는 BED 문서를 해석하고 바이너리 ECG 파일을 메모리로 적재하여 변환모듈(Transform Module)이 바이너리 ECG 데이터를 처리하기 용이한 형태로 전달한다. 변환모듈에 의해서 HL7 aECG 인스턴스가 생성되면 출력모듈(Serialize Module)이 파일로 저장한다.

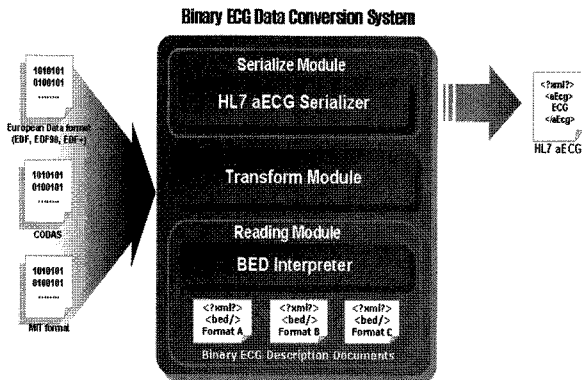


그림 3. 변환시스템의 기본 구조
Fig. 3. Basic structure of conversion system

3. BED 스튜디오 설계

3.1 시스템 구조

BED 스튜디오는 MVC(Model-View-Control) 모델을 기반으로 설계하였고, 그림 1과 같이 View, Control, Model, BED, Common의 5개의 모듈로 구성되어 있다. View 모듈은 자바 스윙과 JGraph[8]를 이용하여 비주얼 편집, XML트리 편집, 포맷검증 뷰어 기능의 GUI를 구현하는 클래스들로 구성되어 있다. Control 모듈은 GUI를 구성하고 있는 객체를 제어하고 GUI에서 받아들인 이벤트를 비즈니스 로직과 연결하는 역할을 수행하며, Model 모듈은 GUI와 비즈니스 로직 클래스들 간에 주고받는 데이터의 인터페이스 역할을 한다. BED 모듈의 클래스들은 BED 문서를 처리하는 비즈니스 로직을 구현하고 있으며, Common 모듈은 시스템에서 공통적으로 사용되는 유틸리티 클래스로 구성되어 있다.

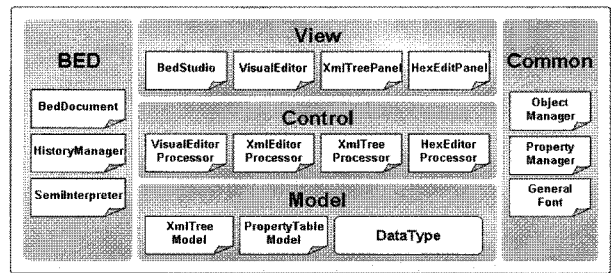


그림 4. BED 스튜디오의 주요 모듈 구성도
Fig. 4. Structure of major modules in BED Studio

3.2 주요 클래스 다이어그램

그림 5은 BED 스튜디오를 구성하는 주요 클래스들의 클래스 다이어그램을 나타낸 것이다. BED 스튜디오는 JTabbedPane를 이용하여 VisualEditor, XMLTreePanel, MappingBinaryPanel의 뷰 패널을 갖는다. VisualEditor, XMLTreePanel, MappingMainrnyPanel은 MVC 모델에서 View 모듈에 속하는 클래스들로 사용자 인터페이스를 구현하는 코드로 이루어져 있다. VisualEditor 클래스는 JGraph API를 이용하여 XML 형태의 BED 문서를 그래픽으로 표현하며 XMLTreePanel은 JTree와 JTextArea를 이용하여 BED 문서를 트리와 텍스트 형태로 표현한다. MappingMainrnyPanel은 JTextArea를 이용하여 16진수(hexa-decimal) 편집기를 포함하며 바이너리 ECG 파일을 16진수 코드로 표현한다.

DataTypeCell은 Model 모듈에 해당하는 클래스로써 BED 스튜디오에서 편집하는 BED 문서의 데이터타입에 대한 인터페이스를 제공한다. VisualEditorProcessor, XMLTreeProcessor, BedMappingProcessor는 Control 모듈에 속하는 클래스로 BED 스튜디오에서 발생하는 이벤트를 입력받아 이벤트를 비즈니스 로직에 연결해 주며, View 모듈 클래스들을 제어한다.

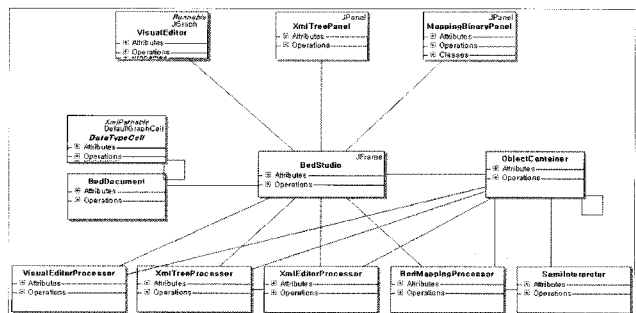


그림 5. BED 스튜디오 클래스 다이어그램
Fig. 5. Class diagram of BED Studio

ObjectContainer는 BED 스튜디오에 사용되는 Model 모듈 클래스나 View 모듈 클래스를 Control 모듈 클래스가 쉽게 사용할 수 있게 하기 위한 유틸리티 클래스로 싱글톤(singleton) 패턴을 이용하여 설계하였다.

3.3 데이터타입의 클래스 다이어그램

BED 스튜디오에서는 8개의 데이터타입 클래스를 이용하여 BED 문서를 처리하며, 그림 6은 데이터타입의 클래스 다이어그램을 UML로 나타낸 것이다. 모든 데이터타입은 추상 클래스인 DataTypeCell을 상속받으며 XML 경로를 나타낼

수 있는 XmlPathable 인터페이스를 구현한다. 8개의 데이터 타입 중 하위 데이터타입을 가질 수 있는 DataType Concatenate, DataTypeStruct, DataTypeInteger, RootCell 은 Holdable 인터페이스를 구현한다.

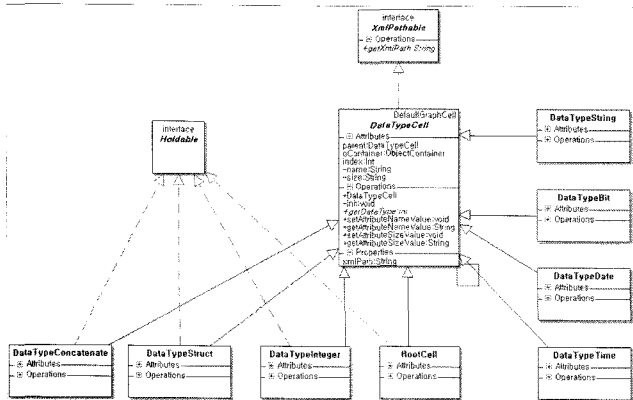


그림 6. BED 스튜디오의 데이터타입의 클래스 다이어그램
Fig. 6. Class diagram of data types in BED Studio

3.4 비주얼 편집과정의 시퀀스 다이어그램

그림 7은 비주얼 편집과정을 나타낸 시퀀스 다이어그램이다. 사용자가 BedStudio에서 데이터타입을 추가하면 BedStudio는 VisualEditor에 데이터타입을 도형으로 나타내는 이벤트를 발생시킨다. VisualEditor는 VisualEditor Processor를 호출하여 해당 데이터타입이 그려지게 될 좌표와 데이터타입에 관한 정보를 전달한다. VisualEditor Processor는 BedDocument 클래스를 이용하여 BED 문서를 갱신한다. BED 문서가 갱신되면 BedDocument는 HistoryManager를 이용하여 BED 문서의 갱신기록을 저장한다. VisualEditorProcessor는 BED 문서에 데이터타입을 갱신한 다음 AttributeTableProcessor와 DataTypePainter를 이용하여 VisualEditor의 다이어그램을 다시 그린다.

4. 시스템 구현 및 사용자 인터페이스

BED 스튜디오는 BED 문서 작성시 필요한 기능과 작성 프로세스의 연관성을 고려하여 설계된 비주얼 편집기, XML 트리 구조를 활용하여 BED 문서를 편집하는 XML트리 편집기, 그리고 생성된 BED 문서에서 바이너리 ECG 파일포

맷의 오류 유무를 시각적으로 검증할 수 있는 포맷검증 뷰어로 구성된다.

4.1 비주얼 편집기

비주얼 편집기는 XML에 대한 높은 수준의 지식이 없더라도 BED 문서를 효과적으로 작성할 수 있도록 개발된 BED 스튜디오의 초보자용 편집기이다. 비주얼 편집기가 BED 언어의 데이터타입과 문서를 그래픽으로 보면서 문서를 편집할 수 있는 시각적인 인터페이스를 제공하기 때문이다.

비주얼 편집기는 BED 문서 작성시 이용할 수 있는 데이터타입을 버튼형태로 구현한 데이터타입 패널과 데이터타입의 속성을 추가, 삭제, 수정할 수 있는 데이터타입 속성패널, 그리고 BED 문서를 2D 그래픽으로 표현하는 비주얼 편집패널로 구성된다. 비주얼 편집기는 BED 문서 작성시 오류 가능성과 작성시간을 줄이기 위해 데이터타입 패널의 데이터타입을 비주얼 편집패널 위에 위치한 데이터타입 영역으로 드래그앤드롭 하는 직관적인 인터페이스를 제공한다. 데이터타입은 비주얼 편집패널 위에 직사각형 형태로 그려지며 직관적으로 데이터타입을 판단하기 위하여 고유의 아이콘을 표시하였으며 바이너리 파일의 구조를 이해하는데 가장 중요한 정보인 데이터타입의 name과 size 속성을 표시하였다.

비주얼 편집패널 위에 위치한 데이터타입은 팝업메뉴를 통하여 추가, 삭제, 순서변경 등 기능을 지원하여 사용자의 편의성을 높였다.

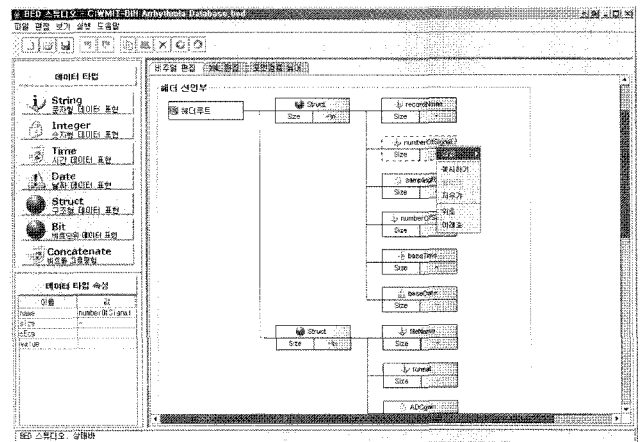


그림 8. 비주얼편집 화면
Fig. 8. Visual editing interface

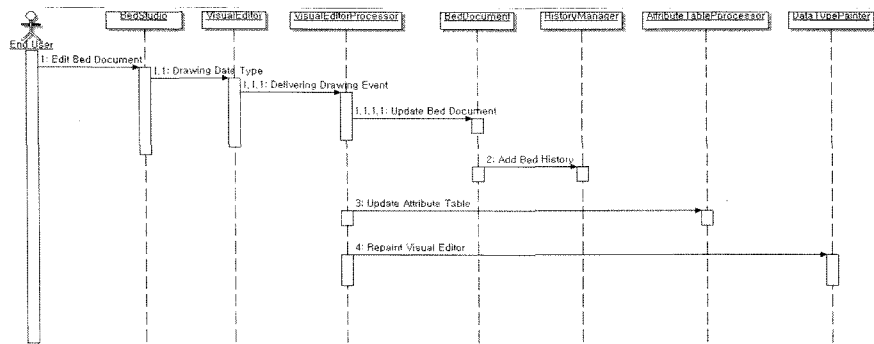


그림 7. 비주얼 편집의 과정을 나타낸 시퀀스 다이어그램
Fig. 7. Sequence diagram of visual editing process

4.2 XML트리 편집기

비주얼 편집기는 XML에 대한 높은 수준의 지식이 없더라도 BED 문서를 쉽게 작성할 수 있는 장점을 가진다면, XML트리 편집기는 XML 숙련자가 XML트리 구조를 기반으로 BED 문서를 생성해 나가는 방식으로, XML 코드 수준보다 상위단계의 트리 수준에서 문서를 편집하기 때문에 숙련자들의 편의성을 지원한다는 장점을 가진다. XML트리 편집기는 사용자가 편집작업을 수행하는 그림 9의 왼쪽에 나타난 "XML트리 편집" 패널과 "데이터타입 속성" 패널, 그리고 사용자가 편집작업을 하면서 생성되는 XML 코드를 참조하는 오른쪽에 나타난 "텍스트 뷰어" 패널로 구성되어 있다.

XML트리 편집기는 BED 문서 작성시 발생할 수 있는 XML 구문오류와 BED 문법오류를 방지하기 위해 그림 9에 나타난 것같이 BED 문서 탐색기 패널에서 팝업메뉴에 의한 편집 인터페이스를 제공한다. 트리형태로 표현된 BED 문서에서 데이터타입을 선택하여 마우스의 오른쪽 버튼을 클릭하면 팝업메뉴가 호출할 수 있다. 호출된 팝업메뉴는 선택된 데이터타입에 적용할 수 없는 기능은 비활성화 되기 때문에 사용자가 XML 구문오류나 BED 문법오류를 범하지 않고 효율적으로 문서를 편집할 수 있게 해준다. 팝업메뉴로 BED 문서를 편집하면서 텍스트뷰어 패널에 BED 문서의 소스 코드를 즉시 확인할 수 있도록 하여 사용자 편의성을 높였다.

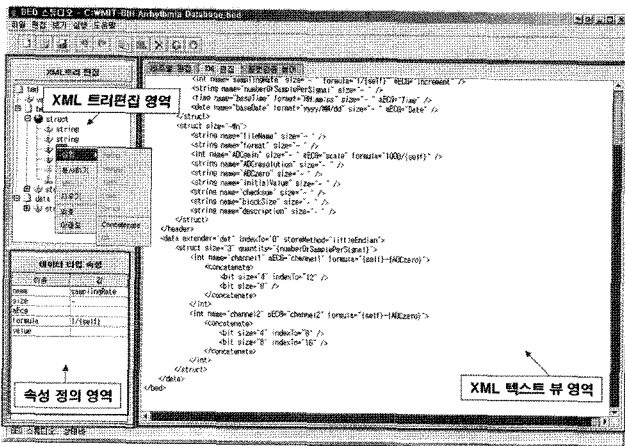


그림 9. XML트리 편집 화면
Fig. 9. XML tree-based editing interface

4.3 포맷검증 뷰어

BED 스튜디오에서 생성된 BED 문서는 사용자의 부주의로 인해 바이너리 ECG 파일포맷에 대한 데이터타입의 중복 정의나 누락, 또는 위치 및 크기로 오류를 포함할 수 있으며, 이러한 오류는 부정확한 ECG 데이터변환을 야기시킨다. 그러므로 복잡한 형태의 바이너리 ECG 파일포맷을 기술하는 경우 생성된 BED 문서가 목표 ECG 파일포맷을 정확하게 기술하고 있는지를 시각적으로 검사할 수 있는 도구가 유용한데, 포맷검증 뷰어가 이러한 기능을 지원한다.

BED 문서에 대한 포맷검증 뷰어는 그림 10에 나타난 것처럼 크게 BED 문서를 구성하는 데이터타입의 이름과 값의 쌍을 보여 주는 데이터타입 테이블 영역과 바이너리 ECG 파일에 저장된 실제 데이터값을 보여주는 16진수코드 영역으로 구성되어 있다. BED 문서 편집 후 BED 문서에 대한 포맷검증 뷰어로 이동하면 왼쪽 프레임에는 생성된 BED 문서의 상수, 헤더, 데이터 선언부에 포함된 데이터타입들이 테

이블 형태로 나타난다. 그런 다음, 오른쪽 프레임에서 이 BED 문서에 대응되는 바이너리 ECG 파일을 선택 후 확인 버튼을 누르면 16진수코드 영역에 선택된 바이너리 ECG 파일의 실제 데이터가 16진수코드 형식으로 출력되고 데이터 타입 테이블에는 해당 데이터타입의 값이 출력된다. 그러면 사용자는 데이터타입 테이블에서 데이터타입으로 선언한 바이너리 ECG 데이터의 샘플링횟수나 채널의 샘플크기를 시각적으로 확인할 수 있기 때문에 BED 문서가 정확하게 작성되었는지를 비교적 쉽게 검사할 수 있다. 또한 데이터타입 테이블에서 데이터타입을 선택을 하면 16진수코드 영역에는 해당 데이터타입에 매칭되는 바이너리 파일에서의 데이터영역과, 16진수 값, 아스키코드(ascii-code) 값을 붉은색 박스로 표시하여 쉽게 식별할 수 있도록 함으로써 오류 가능성을 감소시켰다.

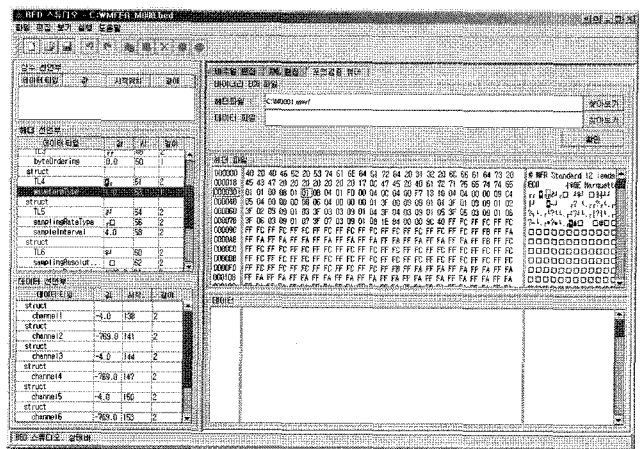


그림 10. 포맷검증 화면
Fig. 10. Format checking interface

5. 평가

BED 변환시스템에서 사용자가 BED 문서를 작성하려면 어느 정도 수준의 XML 지식을 필요로 하기 때문에 초보자들이나 XML에 대한 지식이 없는 의료정보시스템의 관리자 등의 경우 BED 문서 작성시 어려움에 직면하게 된다. 또한 BED 언어 구문에 따라 XML로 바이너리 ECG 파일포맷을 기술하는 작업이 난이도가 높은 것은 아니지만, XML 숙련자라고 하더라도 바이너리 ECG 파일포맷과 BED 언어 및 문서의 구조를 모두 준수하면서 텍스트 편집기를 사용하여 BED 문서를 작성하는 일은 시간소모적이고 오류발생 가능성이 높은 작업이다.

본 연구의 결과인 BED 스튜디오는 이러한 환경에서 사용할 수 있는 초보자를 위한 비주얼 편집기, XML 숙련자를 위한 XML트리 편집기, 그리고 생성된 BED 문서의 정확성을 시각적으로 검증할 수 있는 포맷검증 뷰어를 포함하므로, 초보자뿐만 아니라 숙련자들도 BED 문서 작성시 활용하면 높은 사용자 편의성을 제공받을 수 있다.

비주얼 편집기는 BED 문서 작성시 사용자가 정해진 프로세스에 따라 필요한 엘리먼트를 선택하면 BED 언어의 구문 오류를 포함하지 않는 코드생성이 가능하게 되며, 텍스트 편집기를 사용하는 것보다 작업 생산성이 크게 향상 될 것이다. 비주얼 편집기는 XML관련 지식이 없더라도 BED 언어 및 문서 구조만을 이해하면 BED 문서를 비교적 쉽게 작성

할 수 있지만, 바이너리 ECG 파일포맷이 복잡한 경우는 BED 문서 전체의 구조를 파악하기 어려운 단점을 갖는다. 따라서 XML관련 지식을 어느 정도 갖춘 사용자의 경우와 바이너리 ECG 파일포맷이 복잡한 경우는 숙련자를 위한 XML트리 편집기로 작업하면 XML 텍스트 형태가 아닌, 트리형태의 시각적 인터페이스를 통해 BED 문서의 구조를 쉽게 파악할 수 있어 문서작성 작업의 효율성을 증가시킬 수 있다. XML트리 편집기의 경우도 비주얼한 XML 트리 상에서 BED 문서 편집작업을 수행하며, 화면 우측에 출력되는 XML 텍스트는 단지 사용자에게 제공되는 인터페이스이다. 비주얼 편집기나 XML트리 편집기를 통해 생성된 BED 문서는 BED 언어의 구문 오류나 문서구조 오류는 포함하지 않으나, 데이터타입의 크기 오류, 즉 엘리먼트 속성값 오류나 위치 오류 등을 포함할 수 있다. 포맷검증 뷰어는 BED 문서에서 각 엘리먼트의 속성값과 이 엘리먼트와 매칭되는 바이너리 ECG 파일의 실제 데이터를 쉽게 비교할 수 있는 형식으로 출력하기 때문에 엘리먼트 속성값 오류나 위치 오류 등을 쉽게 식별할 수 있어, XML 코드 수준에서 검사하는 것보다 쉽고 빠른 디버깅 작업이 가능하다. 그림 11은 BED 스튜디오에서 지원되는 컴포넌트를 활용하여 BED 문서를 작성하는 효율적인 프로세스를 순공학과 역공학 관점에서 나타낸 것이다.

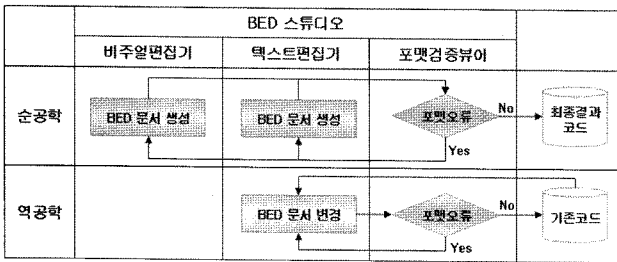


그림 11. BED 스튜디오의 기능 활용 개념도

Fig. 11. Work flow diagram of the components in the BED Studio

6. 결 론

본 논문은 계측기로부터 수집된 ECG 데이터의 상호연동성 지원을 위해 제안된 BED기반의 ECG 변환시스템에서 BED 문서 작성시 사용자의 편의성 및 생산성 향상을 제공하는 비주얼 환경의 BED 스튜디오를 개발하였다. 구현된 BED 스튜디오는 BED 문서 작성시 필요한 기능과 개발 프로세스의 연관성을 고려하여 비주얼 편집기와 XML트리 편집기, 그리고 생성된 BED 문서와 바이너리 ECG 파일간의 매칭정보를 시각적으로 검사하고 오류를 디버깅할 수 있는 포맷검증 뷰어로 구성되어 있다.

BED 스튜디오를 통해 XML 지식이 많지 않은 초보자뿐만 아니라 숙련자들도 BED 문서를 쉽게 작성할 수 있기 때문에, BED 문서 작성시 생산성을 향상시키고 BED기반의 ECG 변환시스템의 적용이 보다 용이하게 될 것이다. 또한 기존의 BED 문서도 역공학 과정을 통해 BED 문서의 구조, 엘리먼트들의 속성값, 그리고 데이터 포맷을 보다 쉽게 파악할 수 있기 때문에 바이너리 ECG 파일포맷의 개선시에도 해당 BED 문서의 수정이 용이하다는 장점을 가진다.

향후에는 이 도구를 다양한 바이너리 BED 파일포맷에 적용해 작업 프로세스를 보다 최적화시키고, 이러한 결과를

BED 스튜디오에 반영하여 사용자 편의성과 BED 문서 작성의 효율성을 높일 계획이다.

참 고 문 헌

- [1] 이병채, "12채널 심전도 해석 알고리즘 프로토타입 설계 및 성능 평가", 대한의료정보학회지, 제 5권, 3호, pp. 127-137, 1999.
- [2] CE. Chronaki, F. Chiarugi, PJ. Lees, M. Brun-Rasmussen, F. Conforti, R. Ruiz Fernandez, C. Zywiets, "Open ECG : a European project to promote the SCP-ECG standard, a further step towards interoperability in electrocardiography", Computers in Cardiology 2002, pp. 285-288, September 2002.
- [3] V. Sakkalis, F. Chiarugi, S. Kostomanolakis, CE. Chronaki, M. Tsiknakis, SC. Orphanoudakis, "A Gateway between the SCP-ECG and the DICOM Supplement 30 Waveform Standard", Computers in Cardiology 2003, pp. 25-28, September 2003.
- [4] J. Shin-Young, K. Heung-Seo. Implementation of ECG Conversion System Using XML Based Descriptor. International Technical Conference on Circuits/Systems, Computers and Communications, ITC-CSCC Proceeding, VOL. 1, pp. 437-440, 2006.
- [5] 구흥서, 정신영 "바이너리 파일 디스크립션 방식을 이용한 ECG 변환 시스템", 대한전기학회, 논문지 55D권 10호, pp. 464-470, 2006.
- [6] BD. Brown, Fabio Badilini, HL7 aECG Implementation Guide, Available at: <http://www.hl7.org>, Accessed June 2, 2006.
- [7] MIT-BIH Arrhythmia Databases, <http://www.physionet.org/physiobank/database/mitdb/>, accessed Jun. 2, 2006.
- [8] JGraph, Available at: <http://www.jgraph.com/>, Accessed Dec 2, 2006

저 자 소 개



구 흥서(Koo, Heung-Seo)

1993년 8월 : 인하대학원 전산학 전공 (이학박사).

1994년~현재 : 청주대학교 컴퓨터정보공학 학과

관심분야 : 지능형 데이터베이스, 정보 모델링, u-헬스케어.
 Phone : 043) 229-8492
 Fax : 043) 229-8432
 E-mail : hskoo@cju.ac.kr