

컴파일 방식을 이용한 바이너리 ECG 변환 시스템의 성능 개선

論 文
56-6-25

Improvement of a Binary ECG Conversion System Utilizing Compilation Technique

具 興 書[†]
(Heung-Seo Koo)

Abstract - In this paper, we develop a new conversion method for implementing binary ECG(Electrocardiogram) conversion scheme to improve our previous research works that supported the conversion of binary ECG files into HL7 aECG for enhancing interoperability of ECG data. HL7 aECG is a XML-based standard for interoperability of ECG waveform. To improve the performance of ECG data conversion, we utilize a compilation-based ECG conversion method on binary ECG files. Our new method supports both flexibility of BED-based ECG conversion mechanism and the performance of direct conversion mechanism.

Key Words : ECG, XML, HL7 aECG, Compilation Technique, ECG Conversion

1. 서 론

최근 정보통신기술을 이용한 유비쿼터스 헬스케어(ubiquitous healthcare, 이후 u-헬스케어)에 대한 연구가 활발하게 진행되고 있다. u-헬스케어는 지금까지 의료기관을 중심으로 제공되었던 건강진료 서비스를 가정과 개인으로 확대하려는 것으로, 특히 질병발생 전의 예방과 질병발생 직후의 즉각적이며 지능적인 대응을 중요시 한다. 또한 일상적이고 자연스러우며 개인에 따라 차별화된 맞춤형 건강관리 서비스를 특징으로 한다[1]. 이러한 u-헬스케어의 실현은 심전도, 혈압, 맥박, 심음 등 사람의 주요 생체신호를 측정해 낼 수 있는 생체신호 계측기술, 생체신호 계측기의 소형화·경량화, 그리고 생체신호 데이터의 상호호환성(interoperability) 지원을 필요로 한다.

심전도(ECG)는 가장 흔한 심장진단 방법이지만, 기존의 모든 ECG 계측기들이 각기 고유한 파일포맷을 사용하여 ECG 데이터를 획득, 저장, 분석하고 있다. 이러한 이유로 동일한 계측기를 사용하지 않는 이상 ECG 데이터를 다른 의료정보시스템의 응용프로그램에서 재사용할 수 없다[2]. 따라서 의료정보시스템 간에 ECG 데이터의 공유를 거의 불가능하게 만드는 상호호환성을 저해하는 가장 큰 요인은 다양하게 존재하는 ECG 파일포맷들이다. 현재 제안되고 있는 ECG 포맷들이 이전 포맷의 장점을 포함하면서 향상된 여러 특징들을 제공하지만, 의료정보시스템 관점에서는 각각의 독자적인 포맷을 지원하는 다양한 계측기들의 ECG 데이터를 통합관리하거나 의료정보시스템 간에 교환하는 것은 매우

어렵고 시간이 소요되는 작업이다[3].

의료정보시스템 간에 ECG 데이터의 상호연동성을 지원하기 위해, 본 연구팀이 연구[4][5]에서 ECG 계측기로부터 수집된 바이너리 ECG 데이터를 HL7 aECG로 변환하는 시스템을 개발하였다. 이 시스템은 시스템관리자가 사전에 목표 바이너리 ECG 파일포맷을 XML기반의 BED(Binary ECG Description) 문서에 기술하면, 변환요청이 발생할 때마다 변환시스템이 BED 문서를 해석(interpretation)하여 해당 바이너리 ECG 데이터를 HL7 aECG 데이터로 변환하는 메커니즘을 사용한다.

이렇게 BED 문서를 이용한 변환기법의 장점은 추후 ECG 계측기의 업그레이드로 인해 바이너리 ECG 파일포맷이 변경되더라도 텍스트 편집기로 BED 문서만을 간단히 수정하면 새로운 포맷의 바이너리 ECG 파일을 HL7 aECG 파일로 변환할 수 있다는 점이다. 그러나, 이 변환기법은 BED기반의 해석 변환방식을 사용하기 때문에 변환요청이 발생할 때마다 실행시간(runtime)에 BED 문서를 해석하여 바이너리 ECG 파일포맷을 인식(recognition)한 다음, 바이너리 ECG 데이터를 읽어서 HL7 aECG로 변환하는 과정을 반복하기 때문에 높은 성능을 요구하는 시스템에 적용하기는 어렵다는 단점이 있다.

본 논문에서는 바이너리 ECG 데이터의 변환시 반복적으로 수행되는 비효율적인 부분을 제거하여 바이너리 ECG 데이터의 변환성능을 향상시킨 BED기반의 컴파일(compilation) 변환방식을 제시한다. 컴파일 변환방식의 요지는 바이너리 ECG 파일포맷의 해석을 시스템관리자에 의해 실행되는 컴파일 단계에서 사전에 수행하고 실행시간에는 사전 단계에서 미리 해석된 데이터 포맷에 따라 바이너리 ECG 파일을 액세스하여 ECG 데이터 변환성능을 개선시킨 기법이다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 XML기반의 표준 포맷인 HL7 aECG에 대하여 기술한다. 3

† 교신저자, 正會員 : 淸州大 理工大 컴퓨터情報工學科 正教授

E-mail : hskoo@cju.ac.kr

接受日字 : 2007年 4月 26日

最終完了 : 2007年 5月 5日

장에서는 본 논문에서 제안하는 BED기반의 컴파일 변환방식에 대해 설명하고, 4장에서는 이번에 제시한 컴파일 변환방식과 연구[4][5]의 해석 변환방식을 변환 메커니즘 측면과 성능측면에서 비교하고, 5장에서 결론을 맺고자 한다.

2. 관련연구

2.1 HL7 aECG

HL7 aECG 표준은 XML을 기반으로 하며 HL7의 RCRIM(Regulated Clinical Research Information Management)이 FDA의 디지털 ECG의 필요성에 의해 만들었으며, 최종적으로 2004년 1월에 HL7 버전3의 표준으로 수락되었고 2004년 5월에는 ANSI의 승인을 받았다[6].

HL7 aECG 표준은 HL7 버전3 메시지 표준에 기반하고 모든 XML 엘리먼트들은 HL7 R-MIM(Refined Message Information Model)로부터 유도되어 정의된다[7]. R-MIM에 기반한 HL7 aECG 문서구조는 ECG의 계측정보 표현부분, 검사대상자에 대한 정보표현 부분, 계측된 ECG 신호의 파형정보 표현부분과 주석 표현부분 등으로 구성된다.

HL7 aECG는 XML에 기반을 두었기 때문에 XML의 장점인 유니코드(unicode) 지원, XML 지원도구 활용으로 인한 손쉬운 파일의 핸들링, XSL 표준기술 사용 가능 등의 장점을 그대로 가지고 있다. 특히 XSL 표준기능의 사용하여 HL7 aECG는 상당히 유연하게 정보를 제공할 수 있게 되었다. 또한 주석 정보는 표준 어휘(vocabulary)를 사용하여 주석정보의 상호호환성을 증대시켰다. 그림 1은 HL7 aECG 파일의 일부를 나타낸 것이다.

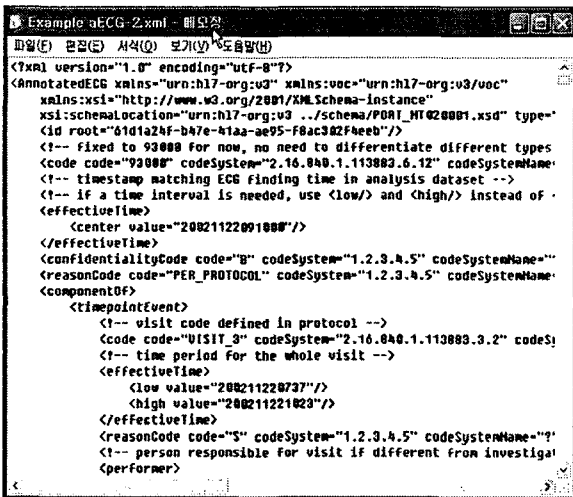


그림 1 HL7 aECG의 예제 파일
Fig. 1 An example of HL7 aECG

3. 컴파일 방식의 ECG 변환시스템 설계

이번 장에서는 성능향상을 위해 새로 제시한 BED기반 컴파일 방식의 바이너리 ECG 변환시스템의 구조와 변환 메커니즘에 대해 설명한다.

3.1 시스템 구조

그림 2는 본 논문에서 제안한 컴파일 변환방식의 시스템 구조를 나타낸 것이다. 이 시스템은 BED 컴파일러(BED Compiler), 변환기(Transformer), 출력기(Serializer)의 3개 주요 모듈로 구성되어 있다. 이들은 사전에 시스템관리자의 명시적인 요청에 의해 실행되는 BED 컴파일러 부분과, 바이너리 ECG 파일의 변환요청이 발생하면 실행되는 변환기 및 출력기 부분으로 구분된다.

BED 컴파일러의 역할은 바이너리 ECG 파일포맷을 기해 놓은 BED 문서를 기반으로 바이너리 ECG 파일포맷을 해석하여, 바이너리 ECG 파일의 액세스와 변환기 및 출력기를 호출할 수 있는 자바 소스코드를 생성한다. 이렇게 생성된 자바 소스코드는 자바 컴파일러에 의해 자바 클래스(이후 컨버터 클래스라 함)로 컴파일된 다음, 운영체제의 파일시스템에 저장된다. 그 후에 바이너리 ECG 파일에 대한 변환요청이 발생하면 ECG 변환시스템이 해당 컨버터 클래스를 호출하게 되고, 이 클래스는 바이너리 ECG 파일을 순차적으로 액세스하면서 변환기를 호출하여 바이너리 ECG 데이터를 HL7 aECG 데이터로 변환한다. 그리고 변환된 HL7 aECG 데이터는 출력기를 통해 파일에 저장한다.

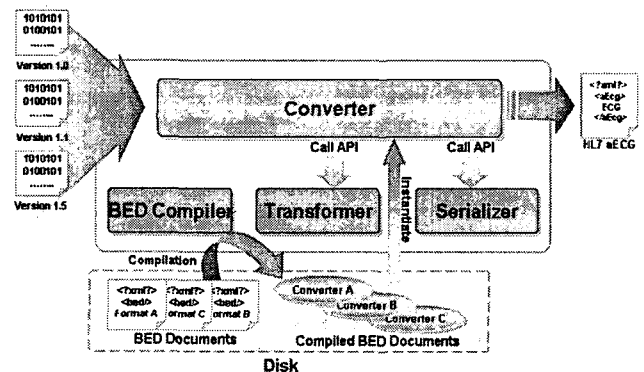


그림 2 컴파일 방식의 ECG 변환시스템의 구조
Fig. 2 Structure of compilation-based ECG Conversion System

3.2 시스템의 주요 구성요소

3.2.1 BED 컴파일러

BED 컴파일러는 BED 문서에 기술된 바이너리 ECG 파일포맷을 기반으로 자바 소스코드와 컨버터 클래스 파일을 생성하는 기능을 수행한다. BED 컴파일러는 JDOM(Java Document Object Model)[8] API를 이용하여 BED 문서를 순차적으로 해석하면서, 자바의 StringBuffer 클래스를 이용하여 바이너리 ECG 파일을 액세스할 수 있는 자바 소스코드를 주기억장치에 생성한다. BED 문서의 해석시 하위 데이터 타입을 가질 수 있는 Struct 데이터타입의 경우는 재귀함수를 이용한다.

이렇게 생성한 자바 소스코드는 OutputStream 클래스를 이용하여 파일시스템으로 출력한 다음, Apache ANT[9] 의

API를 통해 자바 컴파일러를 호출하여 컨버터 클래스를 동적으로 생성한다. 그리고 컨버터 클래스를 파일시스템에 저장한 다음, 바이너리 ECG 파일에 대한 변환요청이 발생할 때마다 ECG 변환시스템이 호출하여 실행시킨다.

3.2.2 변환기

변환기는 바이너리 ECG 데이터를 HL7 aECG 데이터로 변환하는 기능을 수행한다. 그 수행절차는 컨버터 클래스로부터 바이너리 ECG 데이터가 저장된 AEcgBaseData 객체를 전달받아 HL7 aECG 데이터를 담고 있는 JDOM의 Document 객체를 생성한다. 바이너리 ECG 파일로부터 얻어진 환자ID 값은 HL7 aECG 문서의 자유설명 공간인 text 엘리먼트의 텍스트 노드에 임시 저장한다. 그 이유는 HL7 aECG에서 환자ID 명명법으로 trialSubject 엘리먼트의 root와 extension 속성값의 조합을 사용하기 때문에 바이너리 데이터로 표현된 환자ID 값을 직접 저장할 수 없기 때문이다. 그러므로 문자열 연산을 이용하여 환자ID 값을 변환한 후에 trialSubject 엘리먼트의 root와 extension 속성에 저장한다.

3.2.3 출력기

출력기는 HL7 aECG 데이터를 파일로 저장하는 기능을 수행한다. 그 수행절차는 변환기에서 생성한 HL7 aECG 데이터를 전달받아 JDOM의 XMLOutputter 클래스를 이용하여 파일에 저장한다. 현재 버전에 구현된 파일 시스템으로 출력하는 모듈 이외에, 타 의료정보시스템으로 출력하는 모듈을 개발하면 보다 동적으로 ECG 데이터의 교환을 실현할 수 있다.

3.3 클래스 다이어그램

그림 3은 BED기반 컴파일방식의 ECG 변환시스템의 주요 클래스들에 대한 클래스 다이어그램을 나타낸 것이다. BED2ParsingCompiling 클래스는 Interpreter 클래스를 이용하여 BED 문서를 자바 소스코드로 변환하고 변환된 소스코드를 컴파일하는 클래스이다. Interpreter 클래스는 XML 형식의 BED 문서를 JDOM을 이용하여 반복적으로 해석하면서 OutputStream 클래스를 이용하여 자바 소스코드를 파일 시스템에 출력한다. 그리고 Compiler 클래스가 ANT의 API를 통해 호출되면 자바 소스코드를 컴파일한다.

바이너리 ECG 파일에 대한 변환요청이 발생하면 BED2Runner 클래스는 파일시스템에 저장된 컴파일된 컨버터 클래스를 호출한다. 컨버터 클래스는 InterfaceConvertor의 execute 메소드를 포함하므로 BED2Runner 클래스에서 동적 클래스 로딩기술을 이용하여 컨버터 클래스를 실행할 수 있다. Transformer 클래스는 InputStream 클래스를 이용하여 바이너리 ECG 파일의 데이터를 전달받아 HL7 aECG 데이터로 변환한다. AbstractSerializer 클래스는 추상클래스로 serialize 메소드를 포함하며, 이 클래스를 상속받아 여러 형태의 HL7 aECG Serializer 클래스를 만들 수 있다.

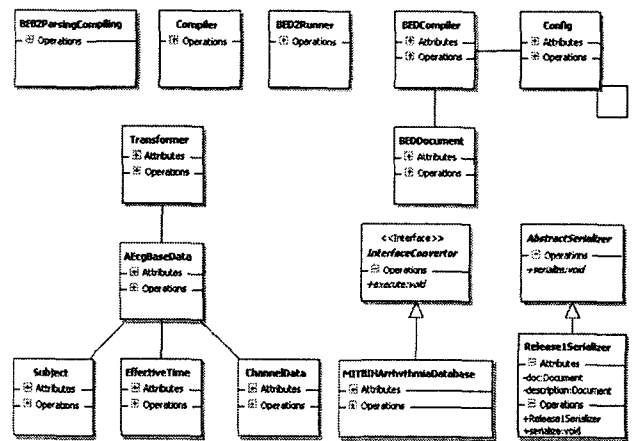


그림 3 ECG 변환시스템의 클래스 다이어그램
Fig. 3 Class diagram of compilation-based ECG Conversion System

3.4 시퀀스 다이어그램

그림 4는 컴파일 방식의 ECG 변환시스템에서 주요 작업의 수행절차를 UML로 나타낸 것이다. 즉, BED 문서를 자바 클래스로 번역하는 컴파일 과정과 컴파일된 컨버터 클래스가 바이너리 ECG 파일을 HL7 aECG 파일로 변환하는 과정을 나타낸 시퀀스 다이어그램이다. 시스템관리자가 Compile and Runner 객체에 BED 문서의 컴파일을 요청하면 Compile and Runner 객체는 Interpreter 객체를 생성하여 BED 문서를 해석한다. Interpreter 객체는 XML 형식의 BED 문서를 해석하여 BED 문서를 자바 소스코드로 번역한다. 생성된 자바 소스코드는 Compiler 객체에 의하여 컴파일된다.

Compile and Runner 객체는 Compiler 객체에 의하여 컴파일된 컨버터 객체를 동적 클래스 로딩기술을 이용하여 로딩 후 execute 메소드를 실행한다. 주기억장치에 로딩된 컨버터 객체는 EcgFileLoader 객체를 이용하여 변환할 바이너리 ECG 파일을 로딩한 후 변환에 필요한 데이터를 HL7 aECG AEcgBaseData 객체에 저장한다. AEcgBaseData 객체는 Transformer 객체에 전달되며 Transformer 객체는 전달받은 데이터에 해당하는 HL7 aECG 객체를 생성한다. Serializer 객체는 생성된 HL7 aECG 객체를 전달받아 파일에 출력한다.

3.5 ECG 변환결과

이번 장에서는 본 논문에서 제시한 BED기반 컴파일 변환방식으로 생성된 결과물, 즉 변환된 HL7 aECG 파일의 유효성(validation)을 검증하고자 한다. 이를 위해 실험 데이터로 MIT-BIH 부정맥 데이터베이스[10]에서 레코드 100번 파일(크기 1,950KB)을 사용하고, 유효성 검증을 위하여 HL7 aECG의 XML Schema를 사용하고 도구로는 Altova사의 XMLSpy를 이용하였다. 그림 5는 XML Schema를 이용하여 HL7 aECG의 유효성을 검증한 내용의 일부를 나타낸 것이다.

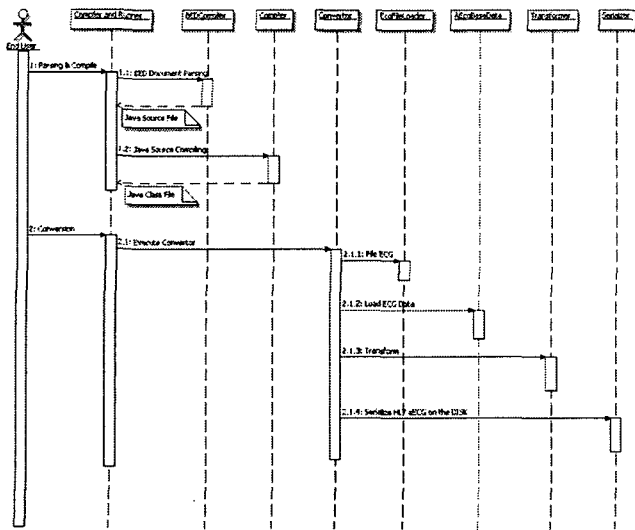


그림 4 ECG 변환시스템의 컴파일과 변환과정을 나타낸 시퀀스 다이어그램

Fig. 4 Sequence diagram of compilation-based ECG Conversion System

변환된 HL7 aECG 파일의 크기는 5,718 KB로 바이너리 ECG 데이터에 비해 약 3배 증가하여, 파일의 크기 측면에서는 해석 변환방식과 동일한 결과가 나타났다. 이것은 컴파일 변환방식과 해석 변환방식에서 바이너리 데이터에 대한 문자열 데이터의 표현을 위해 동일한 방식을 사용하였기 때문이다. 또한, 변환된 파일의 크기 증가는 바이너리 데이터 표현에 비해 문자열 데이터 표현이 더 많은 저장공간을 요구하기 때문에 필연적으로 나타나는 현상이다.

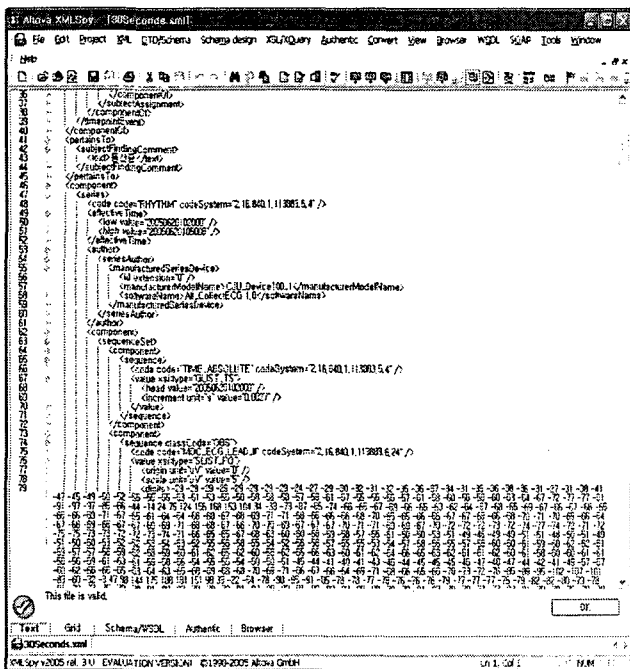


그림 5 XML Schema를 이용한 HL7 aECG의 유효성 검증
Fig. 5 Validating a HL7 aECG document using XML schema

4. 해석 변환방식과의 비교

이번 장에서는 변환 메커니즘과 성능 관점에서 본 논문에서 제시한 컴파일 변환방식과 해석 변환방식을 비교한다

4.1 변환 메커니즘 측면

4.1.1 해석 변환방식

해석방식의 변환시스템은 그림 6과 같이 3개 모듈로 구성된다. 변환요청이 발생하면, 읽기모듈(Reading Module)이 BED 문서를 순차적으로 액세스하면서 바이너리 ECG 파일 포맷을 해석한 다음, 바이너리 ECG 파일을 메모리로 적재하여 변환모듈(Transform Module)로 전달한다. 변환모듈은 읽기모듈로부터 전달된 AEcgBaseData 객체를 이용하여 HL7 aECG 데이터가 저장된 DOM(Document Object Model) 객체를 생성한다. 출력모듈은 변환모듈로부터 전달된 DOM 객체를 JDOM의 XMLOutputter 클래스를 이용하여 파일로 저장한다. 해석 변환방식은 변환요청이 발생하면 바이너리 ECG 파일의 마지막 데이터를 처리할 때까지 '해석 - 변환 - 저장' 주기(cycle)를 반복한다.

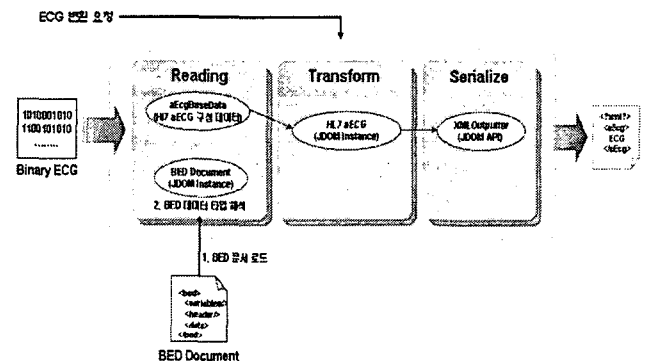


그림 6 해석 변환방식의 ECG 변환 메커니즘
Fig. 6 Mechanism of ECG conversion in interpretation-based conversion method

4.1.2 컴파일 변환방식

본 논문에서 제안한 컴파일 방식의 변환시스템은 그림 7과 같이 BED 컴파일러, 변환기, 출력기의 3개의 모듈로 구성된다. 이 방식이 해석 변환방식과의 차이점은 변환요청이 발생하기 전에 시스템관리자에 의해 실행된 BED 컴파일러가 BED 문서를 해석하여 ECG 변환시스템이 호출할 수 있는 컨버터 클래스 파일을 생성한다는 점이다. 그런 다음, 변환요청이 발생하면 ECG 변환시스템이 컨버터 클래스를 호출하고, 컨버터 클래스는 변환기를 호출하여 바이너리 ECG 데이터를 HL7 aECG 데이터로 변환한다. 그리고 변환된 데이터는 출력기를 호출하여 파일에 저장한다. 컴파일 변환방식은 변환요청이 들어오면 사전에 해석된 파일포맷에 따라 바이너리 ECG 파일의 끝을 만날 때까지 '변환 - 저장' 주기(cycle)를 반복한다.

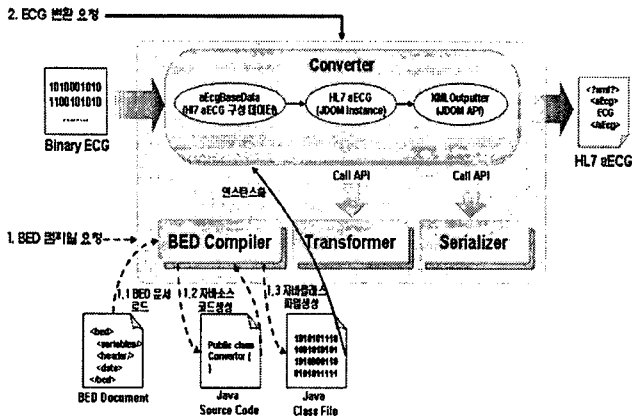


그림 7 컴파일 변환방식의 ECG 변환 메커니즘
Fig. 7 Mechanism of compilation-based ECG Conversion System

표 1은 BED 방식을 사용하지 않고 바이너리 ECG 파일을 HL7 aECG 파일로 직접 변환하는 데이터 포맷 종속적인 직접 변환방식과 해석 변환방식, 그리고 컴파일 변환방식에서 수행되는 주요 작업들을 실행시간 이전 단계와 실행시간 단계로 구분하여 비교한 것이다.

표 1 세가지 변환방식의 단계별 세부 수행작업 비교
Table 1 Comparing the details work for the three ECG conversion methods

	실행시간 이전	실행시간
직접 변환방식	(해당작업 없음)	- 바이너리 ECG 파일 액세스 - HL7 aECG 데이터로 변환 - ECG 파일 생성
해석 변환방식	(해당작업 없음)	- BED 문서 해석 - 바이너리 ECG 파일 액세스 - HL7 aECG 데이터로 변환 - ECG 파일 생성
컴파일 변환방식	- BED 문서 해석 - 자바코드 생성	- 바이너리 ECG 파일 액세스 - HL7 aECG 데이터로 변환 - ECG 파일 생성

4.2 성능 측면

이번 장에서는 본 논문에서 제시한 컴파일 변환방식의 성능을 해석 변환방식과 바이너리 ECG 데이터를 직접 변환하는 직접 변환방식과 비교한다. 실험평가 방법은 다음과 같다. MIT-BIH 부정맥 데이터베이스의 48개 레코드 중 100 ~ 109 까지 10개의 레코드를 한 번씩 변환하여 10개의 결과값을 구한 다음, 이 중에서 최댓값과 최솟값을 제외한 평균값을 사용한다. 실험에 사용한 샘플 데이터는 5분, 10분, 15분, 20분, 25분, 30분 동안 연속적으로 측정된 바이너리 ECG 데이터(2 채널, 360Hz)를 사용하였다. 실험은 PC급인

인텔 펜티엄 노스우드 2.4 Ghz, 7200 RPM의 HDD, 메모리 PC3200 1 GB의 시스템 환경에서 마이크로소프트 윈도우즈 XP 프로페셔널 운영체제에서 수행하였다.

실험결과는 그림 8에 나타난 것과 같이 컴파일 변환방식이 해석 변환방식 대비 약 39% 수준의 우수한 성능을 나타내었다. 직접 변환방식과 비교해도 거의 비슷한 성능인 101% 수준을 나타내었다. 컴파일 변환방식이 직접 변환방식과 거의 유사한 수준의 우수한 성능을 나타내는 이유는 변환처리 과정을 두 단계, 즉 컴파일 단계와 데이터 변환 단계로 분리하고, 표 1에서 알 수 있듯이 성능저하의 원인이 되는 BED 문서 해석을 변환요청의 발생 이전에 수행함으로써 실행시간에는 직접 변환방식과 거의 유사한 ECG 변환 작업(바이너리 ECG 파일 액세스, HL7 aECG 데이터로 변환, ECG 파일 생성)만을 수행하기 때문이다.

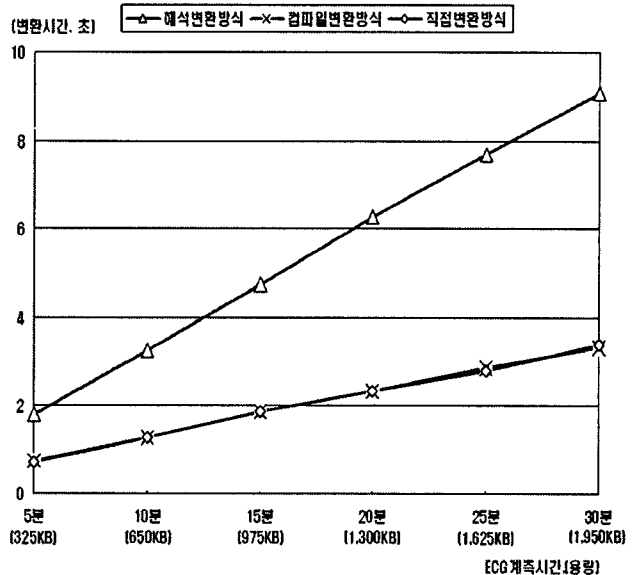


그림 8 세가지 변환방식 간의 성능비교
Fig. 8 Performance comparison for the three ECG conversion methods

5. 결론

본 연구팀은 선행연구[4][5]에서 의료정보시스템 간에 ECG 데이터의 상호호환성을 지원할 수 있는 시스템을 개발하였다. 이 시스템은 기존의 심전도 계측기로부터 수집된 바이너리 ECG 데이터를 북미지역 표준이면서 국제표준(ISO)으로 부상하고 있는 HL7 aECG로 변환한다. 이 시스템에서는 운용 시의 유지보수 비용 절감을 위해 BED라고 명명한 XML기반의 기술문서(description file)를 이용한 변환 메커니즘을 사용하였다. 이 방식은 추후 심전도 기기의 업그레이드로 인한 바이너리 ECG 파일포맷이 변경되더라도 텍스트 편집기를 사용하여 BED 문서만을 간단히 수정하면 업그레이드된 새로운 ECG 파일을 HL7 aECG 파일로 변환할 수 있다는 장점을 가진다.

그러나, 이 변환기법은 BED기반의 해석 변환방식을 사용하기 때문에 변환요청이 발생할 때마다 BED 문서를 해석하

여 바이너리 ECG 파일포맷을 인식(recognition)한 다음, 바이너리 ECG 데이터를 액세스하여 HL7 aECG로 변환하는 과정을 반복하기 때문에 높은 성능을 요구하는 시스템에 적용하기 어렵다는 단점이 있다.

본 논문에서는 해석 변환방식의 이러한 문제점을 해결하면서, 동시에 해석 변환방식의 바이너리 파일포맷 독립적인 장점을 지원하는 컴파일 변환방식을 제안하였다. 컴파일 변환방식은 변환처리 과정을 두 단계, 즉 컴파일 단계와 데이터 변환 단계로 분리하고, 성능지연의 원인이 되는 BED 문서 해석을 변환요청의 발생 이전에 시스템관리자의 요청에 의해 수행되는 컴파일 단계에서 수행하고 실행시간에는 직접 변환방식과 유사한 ECG 변환 작업만 수행함으로써 해석 변환방식에 비해 성능을 2배 이상 크게 개선시켰다. 본 논문에서 제시한 BED기반의 컴파일 변환방식을 채택한 바이너리 ECG 변환시스템은 성능이 더욱 개선되었고, 유지보수가 간편하며, 확장이 용이하여 여러 가지 포맷의 바이너리 ECG 데이터 변환에 쉽게 적용할 수 있으므로 이기종 의료정보시스템 간에 ECG 데이터 교환을 위한 게이트웨이로서 활용될 수 있을 것이다. 향후에는 BED 언어를 기술언어(description language)로서의 범용성을 보다 더 확장하는 연구를 진행할 예정이다.

참 고 문 헌

[1] 백승재, 이철희, 정동현, "유비쿼터스 헬스케어 시스템을 위한 센싱 단말기 구현", 한국정보과학회 학술발표 논문집: A, 제31권, 제1호, pp.124-126, 2004.

[2] C.E. Chronaki, F. Chiarugi, P.J. Lees, M. Brun-Rasmussen, F. Conforti, R. Ruiz Fernandez, C. Zywietz, "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, Sep. 2002.

[3] V. Sakkalis, F. Chiarugi, S. Kostomanolakis, C. E. Chronaki, M. Tsiknakis, S. C. Orphanoudakis, "A Gateway between the SCP-ECG and the DICOM Supplement 30 Waveform Standard", *Computers in Cardiology* 2003, pp. 25-28, Sep. 2003.

[4] H. S. Koo, S. Y. Jung, "Implementation of ECG Conversion System using XML-based Descriptor", *ITC-CSCC 2006, Thailand*, pp.437-440, 2006.

[5] 구흥서, 정신영, "바이너리 파일 디스크립션 방식을 이용한 ECG 변환시스템", *전기학회논문지 D*, 제55권, 제10호, pp.464-470, 2006.

[6] Barry D. Brown, Fabio Badilini, *HL7 aECG Implementation Guide*, Available at: <http://www.hl7.org>, accessed June 2006.

[7] H. Wang, F. Azuaje, G. Clifford, B. Jung, N. Black, "Methods and tools for generating and managing ecgML-based information", *Proceeding of Computers in Cardiology* 2004, pp.573-576, 2004.

[8] JDOM, <http://www.jdom.org>, accessed June 2006.

[9] Apache ANT Project, <http://ant.apache.org/>, accessed Feb. 2007.

[10] MIT-BIH Arrhythmia Database, <http://www.physionet.org/physiobank/database/mitdb/>, accessed Jun 2006.

저 자 소 개



구 흥 서 (具 興 書)

1985년 2월 인하대학교 전산학과
 1989년 2월 인하대학원 전산학전공
 1993년 8월 인하대학원 전산학전공(이학박사)
 1994년~현재 청주대학교 컴퓨터정보공학과 교수
 Tel : 043-229-8492
 Fax : 043-229-8432
 E-mail : hskoo@cju.ac.kr