

# u-헬스 게이트웨이를 위한 동적 관리 모듈 기반의 디바이스 어댑터 모델

## Device Adapter Model based on Dynamic Management Module for u-Health gateway

김 중 탁\* 송 시 윤\*\* 황 희 정\*\*\*  
Kim-Jong Tak Song-Si Yun Hwang-Hee Jeong

### 요 약

u-헬스 서비스를 제공하는 애플리케이션은 기본적으로 개인용 건강관리 기기 즉, PHD와 통신이 가능하고 데이터를 주고받을 수 있어야 한다. 그러나 이러한 장비들은 대부분 자체 프로토콜 규격을 사용하고 있어 측정된 데이터를 수집하고 u-헬스 서비스 프로바이더에 전달하기 위한 통합 게이트웨이 개발이 어려운 현실이다. 본 논문에서는 이러한 문제점 해결을 위해 u-헬스 게이트웨이를 위한 동적 관리 디바이스 어댑터 모델을 제안하고 프로토타입을 구현하였다.

제안된 모델의 검증을 위해 OSGI 기반 u-헬스 게이트웨이 프레임워크 상에서 프로토타입을 구현했으며 1채널 심전도 및 혈압/혈당 기기에 대한 디바이스 어댑터 모듈을 구현하였고 측정 데이터 수집 및 전달의 정확성을 확인하였고 의료정보 표준인 HL7 시스템과의 메시지 연동을 통해 기존 시스템과의 상호운용이 가능함을 보였다. 또한 객체 직렬화 방식의 통신 모델을 적용해 애플리케이션과의 연동 및 기능 확장이 용이한 구조임을 확인할 수 있었다.

### ABSTRACT

It is essential to guarantee a smooth communication and data exchange in a PHD(Personal Healthcare Device) network for applications providing U-health services. In spite of that, most of PHDs are heterogeneous, so the heterogeneity of their protocols makes it difficult to develop an integrated gateway sending sensed healthcare data to U-health service providers. To solve this problem, we suggest the design and implementation of a device adapter model based on dynamic managed module in this paper.

Our model were implemented to work on the OSGI-based gateway middleware and to have interoperability in connection with the HL7 system that is the standard of the Healthcare Information systems. In addition, our model has an architecture supporting a communication based on the object serialization in order to provide extensibility in the functional aspect of applications. Through the experiment on a test-bed which is an implementation of the device adapter module for electrocardiogram and blood-pressure/blood-sugar device having one channel, we have confirmed the accuracy of sensing and sending data.

☞ KeyWords : u-health, OSGI, gateway, middleware, device adapter

## 1. 서 론

IT기술이 발달하고 건강관리에 대한 관심이 증 가함에 따라 u-헬스가 많은 주목을 받고 있다. 기존 u-헬스는 독립적인 헬스케어 형태로서 사용자가 오프라인으로 병원을 통해서만 헬스케어 서비스를 이용할 수 있었으며 의사의 진단과 치료 및 의료 기구에 의존적인 형태였다. 그러나 인터넷

\* 준 회 원 : 가천의과학대 u-헬스케어 연구소 선임연구원

jtkim@gachon.ac.kr

\*\* 정 회 원 : 가천의과학대 u-헬스케어 연구소 연구원

ilyam@hanmail.net

\*\*\* 정 회 원 : 가천의과학대 의료공학부 교수

hwanghj@gachon.ac.kr(교신저자)

[2009/10/12 투고 - 2009/10/22 심사(2009/12/29 2차) - 2010/02/05  
심사완료]

보급의 확산으로 원격 포털 병원과 같은 디지털화된 헬스 서비스 등의 다양한 전산 시스템이 구비되면서 환자의 병을 초기에 발견하고 진단할 수 있게 되었으며, 환자의 데이터를 전자적으로 관리하게 되었다[1][2].

또한 진료 정보의 전산화, 진료 기록 및 의료행위의 전산화를 통한 공유, 건강보험 청구의 전산 매체 활용으로 병원의 디지털화는 꾸준히 진행되어 왔다. 그리고 의료 서비스 산업의 패러다임으로 의료기기 시장이 확장되었고 출산을 감소와 평균 수명 증가로 고령 연구 및 만성 질환자의 비율이 점점 늘어남에 따라 의료 수요는 지속적으로 증가하고 있다. 이에 만성질환자 및 고령자 인구가 급격히 증가하고, 일반 소비자들의 질병 예방에 대한 인식이 증대됨에 따라 개인용 헬스케어 기기 즉, PHD(Personal Health Device)를 이용하여 재택 내 관리도 늘고 있는 형상이다[3]. 이러한 PHD를 이용하면 혈압, 맥박, 심전도 등을 측정하여 측정된 생체 정보를 PC로 전송하고 또한 웹상에서 확인, 관리하여 건강 상태에 대한 적절한 조치를 받을 수 있다. 그러나 각각의 PHD마다 프로토콜이 달라 데이터를 전송하고 확인하는데 많은 어려움이 따르고 대부분 시리얼 통신만을 지원하기 때문에 u-헬스 어플리케이션에서 이들을 모두 지원하는 것은 매우 어려운 일이 아닐 수 없다.

본 논문에서는 이러한 문제점을 해결하기 위해 u-헬스 게이트웨이를 위한 동적 관리 모듈 기반의 디바이스 어댑터 모델을 제안한다. 제안된 모델은 비표준 PHD를 객체화하고 메시지에 대한 객체 직렬화를 지원함으로써 어플리케이션 및 서버와의 유연한 통신이 가능하다. 특히 동적 관리모듈이 설계 되어 u-헬스 게이트웨이의 컴포넌트로서 다양한 서비스 컴포넌트와의 연계가 가능하고 HL7, ISO/IEEE11073 과 같은 u-헬스 및 의료정보 표준 시스템과도 상호 운용이 가능한 구조이다.

제안한 모델의 검증을 위해 OSGi 기반 u-헬스 게이트웨이 프레임워크의 프로토타입을 구현한다.

## 2. 관련 연구

u-헬스 영역에서 PHD로부터 데이터를 수집하는 에이전트인 u-헬스 게이트웨이의 일반적인 구성요소는 다음과 같다[4].

(표 1) u-헬스 게이트웨이 구성요소

Standards	가정용 디바이스(PHD) 인터페이스 (ISO/IEEE11073)
	의료정보 표준 데이터 변환 (HL7/CDA, EN13606)
Function	다중 사용자 및 디바이스 매핑
	다중 서비스 관리
	동적 모듈 관리 기능
	원격 관리 기능

u-헬스 게이트웨이는 다양한 PHD 인터페이스 지원 및 동적 관리 모듈 기능이 요구되는데 본 논문에서는 이러한 요구사항을 만족시키기 위해 동적 관리 모듈 기반의 디바이스 어댑터 모델을 제안하고, 제안된 동적 관리 모듈을 OSGi 기반의 u-헬스 게이트웨이 프레임워크에서 프로토타입을 구현한다.

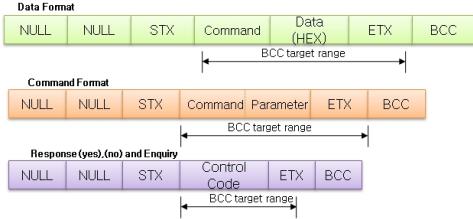
PC와 PHD간의 통신을 위한 HL7(Health Level 7) 메시지 통신 프로토콜인 MLLP(Minimal Lower Layer Message Transport protocol)는 그림1과 같은 구조를 가지고 있다. SB(0x0B)는 스타트 블록, EB(0x1C)는 스톱 블록이다. PC는 전송된 데이터 프레임을 처리하고, 정확한 데이터의 전송여부를 체크하는데 올바른 데이터일 경우에는 ACK(0x06)를, 그렇지 않을 경우는 NAK(0x15)를 보낸다[5]. 그림2는 MLLP 프로토콜을 적용한 메시지이다.



(그림 1) MLLP 프로토콜 구조



의 메시지 포맷이다. STX(0x02)는 Text의 시작을, ETX(0x03)는 Text의 끝을 나타낸다.



(그림 4) EP202 메시지 포맷

또한 EP202에 명령을 주고 데이터를 받기 위해서는 Command Data를 필요로 하며 두 종류로 구분할 수 있다. PC와 같은 모듈에서 EP202로 전송하는 경우와 EP202에서 모듈로 전송하는 경우이다. Command Data는 표1과 같다.

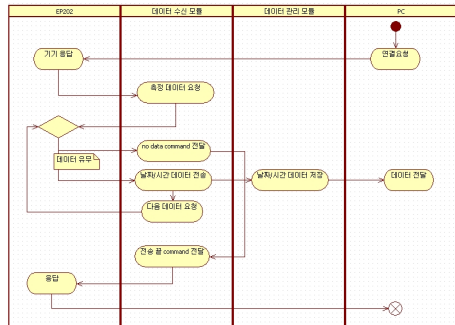
(표 2) EP202 Command Data

Command	Descriptions	Direction	In Use
0x10	Request date/time	Module →EP	First request date/time
0x11	Reserve		
0x14	Send date/time	EP→Module	Send date/time
0x16	No data	EP→Module	No data in memory
0x19	NEXT	Module →EP	RequestNext date/time
0x20~0x7F	Request Data(0~96)	Module →EP	Request block1data~block96data
0x80~0x8F	Send Data(0~96)	EP→Module	Send block 1data~block 96data
0xFF	End of transmission	Module →EP	End of transmission

### 3.2 동적관리를 위한 프로토타입 구현

본 논문에서는 IEP202 인터페이스를 정의하고 이를 구현하는 EP202Base 클래스와 Serializable

인터페이스를 상속받는 EP202Data 클래스를 구현하였다. 또한 다양한 데이터 포맷 변환을 위한 Util 클래스와 원격지로의 전송 테스트를 위한 ISender 인터페이스와 Sender 클래스도 함께 구현하였다. 그림5는 구현한 프로토타입의 액티비티 다이어그램이다. PC에서 EP202에 연결 요청을 하면 EP202는 응답을 보내준다. PC에서 응답을 받으면 데이터 수신 모듈에서는 EP202에 측정된 데이터를 요청하고 EP202는 측정된 데이터와 날짜 정보를 전송한다. 수신 모듈에서는 데이터를 받고 EP202에서 데이터가 없다는 메시지를 보낼 때까지 다음 데이터를 요청한다. 수신모듈에서는 데이터를 받아 데이터 관리 모듈로 전달하고 이 모듈에서는 받은 데이터를 객체 직렬화를 이용하여 저장하고 PC에 제공한다.



(그림 5) 시스템 액티비티 다이어그램

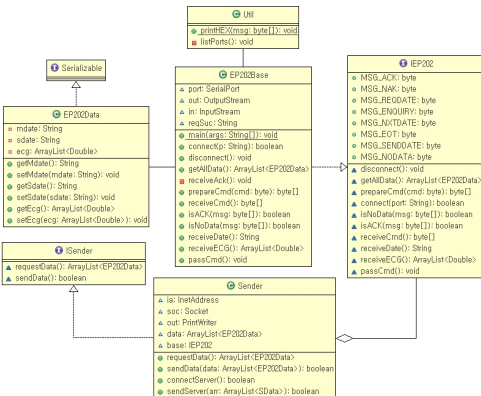
#### 3.2.1 EP202Data 정의

측정된 데이터를 DB나 파일구조가 갖는 문제점을 해결하기 위해 EP202Data객체를 이용하여 객체 직렬화된 형태로 저장을 한다. EP202Data 클래스는 객체 직렬화된 클래스로 측정 일시, 전송 일시, 심전도 측정 데이터를 변수로 갖고, 각각의 변수에 대한 getter, setter 메소드를 갖는다.

#### 3.2.2 인터페이스 정의 및 구현

클래스 간에 종속성을 줄이고 메소드명을 동일하게 구현하기 위해 인터페이스를 IEP202로 정의

하고 구현하였다. 인터페이스에서는 메시지를 보냈을 때 기기의 반응 유무를 알 수 있도록 하는 바이트 타입의 제어 코드 리스트 변수를 정의한다. 이 변수들은 EP202 API 문서에 정의된 구조이다. PC에서 EP202로 전송하고, EP202에서 PC로 전송하는 메시지를 바이트 타입의 변수로 정의한다. 그림6에서 시리얼 통신을 하기 위한 connect()와 disconnect() 메소드, EP202로부터 심전도를 측정할 날짜와 심전도 데이터를 받기 위한 receiveData()와 receiveECG() 메소드, 이 메소드로부터 받은 데이터를 객체 직렬화 된 EP202Data 타입의 ArrayList로 만들어 리턴해 주기 위한 getAllData() 메소드 등을 정의한다.



(그림 6) 프로토타입 클래스 다이어그램

시리얼 통신을 하기 위한 connect() 메소드 파라미터는 String 변수를 전달받고 CommPortIdentifier 클래스의 getPortIdentifier() 메소드는 포트를 파라미터로 전달받아 그 값을 확인한 후 포트를 열고 포트의 OutputStream과 InputStream의 객체를 이용하여 데이터를 송수신한다. disconnect() 메소드는 OutputStream과 InputStream 객체 버퍼에 있는 데이터를 실제 스트림으로 전송하고 메모리를 flush하고 포트를 닫아준다. 디바이스와 연결이 된 후 getAllData() 메소드를 호출한다. getAllData() 메소드는 PC와 EP202간 요청을 보내고 응답을 받은 PC에서 날짜와 시간 데이터를 요청한다. 그리

고 MSG\_NXTDATE 명령어를 전송하여 데이터를 요구하고 데이터가 있다면 계속 반복문을 실행하며 EP202Data 객체를 생성해 심전도 데이터를 ArrayList에 저장하고 데이터가 없다면 종료 명령어를 전송한다. 애플리케이션에서는 이 getAllData() 메소드를 호출함으로써 부가적인 구현 없이 측정된 심전도 데이터를 얻을 수 있다. 그림7은 구현한 프로토타입의 소스코드의 일부분이다.

```

public ArrayList<EP202Data> getAllData() throws IOException {
    ArrayList<EP202Data> epData = new ArrayList<EP202Data>();
    byte[] rmsg;

    // PC <-> EP : enquiry
    out.write(prepareCmd(MSG_ENQUIRY));
    receiveAck();

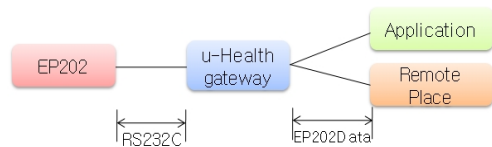
    // PC <-> EP : request date and time
    out.write(prepareCmd(MSG_REQDATE));
    rmsg = receiveCmd();

    // MSG_NXTDATE 를 보내 MSG_NODATA 가 아닌동안 계속 루프를 돌며
    // EP202Data 객체 생성해 ECG데이터 저장
    while (true) {
        if (!isNoData(rmsg)) {
            EP202Data data = new EP202Data();
            data.setMdate(receiveDate());
            data.setEcg(receiveECG());
            epData.add(data);
            out.write(prepareCmd(MSG_NXTDATE));
            rmsg = receiveCmd();
        } else {
            System.out.println("No Data !!");
            out.write(prepareCmd(MSG_EOT));
            receiveAck();
            break;
        }
    }
    return epData;
}
    
```

(그림 7) getAllData() 메소드 구현

#### 4. 실험

제안한 모델의 검증을 위한 동적 관리 모듈의 프로토타입은 윈도우 XP 운영체제와 자바 1.6 버전 환경 위에서 통합개발 플랫폼인 이클립스와 자바 언어를 사용하여 구현하였다. 그림8은 구현된 프로토타입을 실행한 시스템 구성도이다.



(그림 8) 프로토타입 시스템 구성도



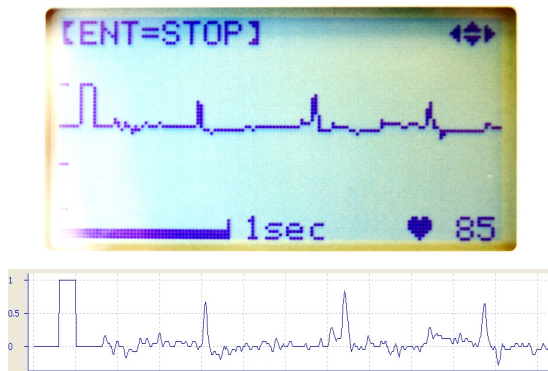
기기는 일본의 Parama-Tech사에서 제조한 심전도 측정기기인 EP202와 엘바이오의 Webdoc plus를 사용했고 디바이스 인터페이스로는 RS232C(USB to Serial)를 사용하였다. EP202의 전송 프로토콜은 비동기 전송방식을 사용하고 38400bps의 속도를 가지며 데이터비트는 8비트이다. connect() 메소드를 이용하여 시리얼 통신 시작한 후 getAllData() 메소드를 이용하여 객체 직렬화 된 데이터를 받았다. 받은 데이터는 String타입의 낱짜와 그 낱짜 별로 측정된 각 심전도 데이터가 double 타입으로 저장되어 있다. 받은 데이터 중 하나의 심전도 데이터 6144개 중 50개를 추출하여 그림9와 같이 나타내었다.

```

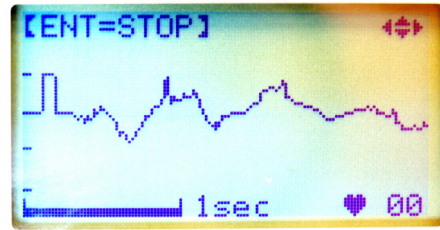
Problems Tasks Properties Servers Snippets Console
<terminated> EP202TestApp [Java Application] C:\Program Files\Java\jre1.6.0_04\bin\javaw.exe
200909211632065003180
.....
-0.12, -0.12, -0.12, -0.08, -0.08, -0.08, -0.08, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.04, 0.04, 0.08, 0.08, 0.08, 0.04, 0.04,
0.04, 0.04, 0.08, 0.08, 0.08, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12,
0.12, 0.08, 0.04, 0.04, 0.08, 0.04, 0.04, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -0.04, -0.04, -0.08, -0.08, -0.12, ...
    
```

(그림 9) 직렬화된 데이터 출력

그림10과 그림11은 동적 관리를 모듈로 구현한 프로토타입을 실행하여 그림 9에서 얻은 심전도 데이터를 애플리케이션으로 전송하여 작성한 심전도 그래프와 EP202 측정 화면이다.



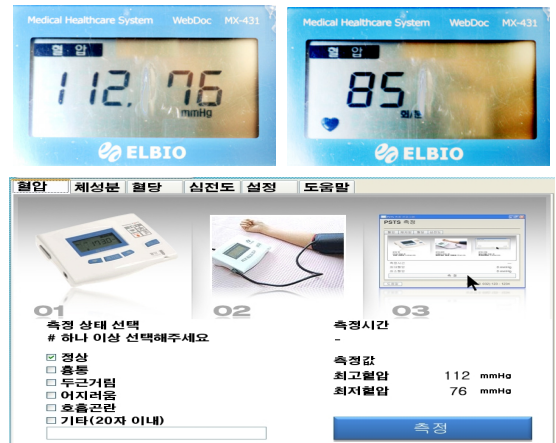
(그림 10) 정상 심전도 EP202 화면과 애플리케이션 그래프



(그림 11) 비정상 심전도 EP202 화면과 애플리케이션 그래프

첫 번째 그래프는 심전도가 정상인 사람의 그래프이고 두 번째 그래프는 파형이 작아 리듬이 확정되지 않는 사람의 그래프이다. EP202 기기에서 출력된 그래프와 제안된 모델을 통해 직렬화된 객체를 애플리케이션으로 전송하였을 때도 동일한 데이터를 얻음을 확인할 수 있었다.

또 하나의 PHD 장비인 혈압, 혈당, 체성분을 측정하는 가정용 통합기기인 WebDoc Plus 장비를 본 논문에서 제안한 모델을 통해 실험한 결과 그림 12 와 같이 측정된 결과를 확인 할 수 있었다.



(그림 12) 정상인의 혈압 측정 화면과 애플리케이션 측정결과 화면

그림 12는 정상인의 혈압을 측정한 결과화면과 애플리케이션으로 전송된 측정 결과를 나타낸다.

## 5. 결 론

본 연구에서는 PHD와 게이트웨이 어플리케이션 간에 보다 효율적인 통신 방법을 제공하고 기능의 확장이 용이한 동적 관리 모듈 기반의 디바이스 어댑터 모델을 제안하였다. 제안된 모델은 복잡한 데이터 구조를 갖는 객체를 송수신 하는데 유용한 객체 직렬화 방식의 메시지 구조를 채택 하였고 OSGi 기반 u-헬스 게이트웨이 프레임워크 상에서 다른 모듈과의 상호 운용이 가능함을 보였다. 실험을 위해 1채널 심전도 기기(EP-202) 및 Webdoc(혈압,혈당)에 대한 디바이스 어댑터 모듈을 구현하였으며 측정된 데이터가 게이트웨이로 전송되어 디바이스 및 메시지가 객체화 된 것을 확인할 수 있었다. 또한 이를 필요로 하는 어플리케이션 및 서비스 컴포넌트와의 연동과 표준 HL7 서비스와의 연계가 가능함을 보였다. ISO/IEEE 11073 규격에 해당하는 PHD 기기에 서 이를 지원하는 매핑 모듈을 구현 한다면 ISO/IEEE 11073 을 따르는 게이트웨이 시스템과 도 연동할 수 있다.

## 참 고 문 헌

- [1] M. Shepherd, D. Zitner, and C. Watters, "Medical Portals: Web-Based Access to Medical Information," Proceedings of the 33rd Hawaii International Conference on System Sciences 2000, Vol. 5, 2000
- [2] MindBranch Asia Pacific Co. Ltd, u-Health 시장 현황 및 전망, ITA 수출정보 데이터베이스, 2005
- [3] 고은지, "전자기업들의 신성장 동력'Healthcare'", HNFOCUS, 23호, 한국홈네트워크산업협회, 2008년 12월
- [4] 송시윤,황희정, "의료 게이트웨이를 위한 u-Healthcare 애플리케이션 프레임워크", 09 춘계 학술 발표 대회, pp.349~353, 한국 인터넷 정보 학회, 2009
- [5] <http://www.hl7.org/v3ballot/html/infrastructure/transport/transport-mlp.htm>
- [6] Cay S.Horstmann. Gray Cornell, "Core Java 2", Sun Microsystems, 1999
- [7] 구태완, 정연진, 엄상용, 이광모, "RMI-IIOP를 이용한 COBRA 환경에서의 XML 객체 모델링", 01 가을 학술발표논문집, Vol.28, No.2, 한국정보과학회, 2001
- [8] 서동민, 윤종현, 여명호, 유재수, 조기형, "에이전트 플랫폼에서의 효율적인 결합-허용을 제공하는 이주 기법", 07 한국콘텐츠학회논문지, Vol.7, No.9, 2007

● 저 자 소 개 ●



**김 종 탁**

1996년 인천대학교 컴퓨터공학과 졸업(학사)  
2001년 인천대학교 컴퓨터공학과 졸업(석사)  
2008년 인천대학교 컴퓨터공학과 졸업(박사)  
2009년 ~ 현재 u-헬스케어 연구소 선임 연구원  
관심분야 : 유헬스케어, OSGi, 상황인식, 미들웨어,  
E-mail : jtkim@gachon.ac.kr



**송 시 윤**

2006년 가천의과학대학교 IT학과 입학  
2008년~ 현재 u-헬스케어 연구소 연구원  
관심분야 : 유헬스케어, OSGi, ISO/IEEE 11073, HL7  
E-mail : ilyam@hanmail.net



**황 희 정**

2000년 인하대학교 컴퓨터공학과 석사  
2007년 인천대학교 컴퓨터공학과 박사  
2009년 - 현재 가천의과학대학교 의료공학부 IT학과 교수  
관심분야 : 소프트웨어공학, u-헬스, OSGi, 분산객체 등  
E-mail : hwanghj@gachon.ac.kr