

## OCL을 이용한 빅데이터 기반의 생체신호 분석 모델

김태웅\*

### A Model of Vital Signs Analysis based on Big Data using OCL

Tae-Woong Kim\*

\*Assistant Professor, Department of Computer Education, Silla University, Busan, 46958 Korea

#### 요 약

오늘날 생체신호의 종류 및 크기가 방대해짐에 따라 생체신호를 빅데이터로 정의하고 이를 분석하고자 하는 연구가 활발히 진행되고 있다. 하지만 생체신호를 빅데이터로 처리함에 있어 소셜 네트워크에서 발생하는 빅데이터 처리와 비슷한 방법을 이용하고 있다. 생체신호 빅데이터는 특징값을 추출하여 이를 별도로 저장하고 다양한 분석기기로 분석할 수 있어야 한다. 즉 데이터의 상호운용성과 호환성이 보장되어야 하며 분석도구에서의 지표 표현이 간결해야 한다. 이를 위하여 본 논문에서는 생체신호를 표준화된 HL7의 표준 메타모델기반으로 표현하고 OMG의 표준 수학적 명세 언어인 OCL을 이용하여 생체신호를 분석하는 모델을 제안한다. 또한 제안한 모델을 이용하여 심전도 데이터를 이용한 칼로리 소모량을 구해봄으로써 활용 가능성을 확인한다.

#### ABSTRACT

As the type and size of vital signs become extensive at the moment lately, a research is actively progressing to define vital signs as big data and analyze it. We generally use a similar method of processing big data on social network as a way to treat vital signs as big data. Vital Sign Big Data should be extracted as feature data, stored separately, and analyzed with various analytical instruments. In other words, it should ensure interoperability and compatibility of data, and the index expression in analytical tools should be concise. For this end, I defined the vital sign as the standard meta-model base of HL7 in this dissertation, and I propose a model for analyzing vital signs using OCL, the OMG's standard mathematical specification language. In addition, the proposed model can be confirmed the applicability by figuring out the consumption of calories using ECG data.

**키워드** : 생체신호, 빅데이터, aECG, 메타모델, OCL

**Key words** : vital signs, Big Data, Annotated ECG(Electrocardiogram), meta-model, OCL(Object Constraint language)

Received 31 August 2019, Revised 6 September 2019, Accepted 9 September 2019

\* Corresponding Author Tae-Woong Kim(E-mail:twkim@silla.ac.kr, Tel:+82-55-999-5865)

Assistant Professor, Department of Computer Education, Silla University, Busan, 46958 Korea

Open Access <http://doi.org/10.6109/jkiice.2019.23.12.1485>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서론

최근 다양한 생체신호를 측정할 수 있는 웨어러블 디바이스의 증가로 인해 생체신호의 종류 및 크기가 방대해지고 있다[1]. 방대한 크기의 생체신호는 빅데이터의 특징을 가지게 되었으며, 빅데이터화된 생체신호를 활용하여 질병 예방, 관리, 진단과 치료방법을 찾고자 하는 다양한 연구가 진행되고 있다[2-4]. 빅데이터의 탄생 배경은 처음에는 평판분석, 소셜 네트워크 분석, 군집 분석등과 같은 텍스트 마이닝에서 시작하여 현재는 생체신호(본 논문에서는 심전도나 가속도신호와 같은 일정시간 계속 유입되는 선형적인 신호)를 분석하는 기술로써 빅데이터를 활용하는 사례가 증가하고 있다[5]. 일반적으로 빅데이터 기술을 이용하여 생체신호를 분석하는 방법에는 생체신호 기기로부터 수집된 raw-data를 순차적으로 저장하고 이를 신호 분석 알고리즘을 통하여 분석하고자 하는 데이터의 특징값(본 논문에서는 심전도 데이터에서 추출 가능한 R-peak, RR-Interval등과 가속도 신호에서 추출 가능한 걸음 수 등)을 추출하여 저장한다. 추출된 특징값 또한 선형적인 데이터로서 빅데이터의 성질을 가지고 있다. 생체신호의 raw-data나 특징값은 같은 빅데이터 성질을 가지나 데이터를 활용하는 측면에서 raw-data와 특징값은 다르게 저장되어 활용된다. 즉 raw-data는 최초 생체신호가 수집된 후 전처리를 통해 신호를 보정하고 특징값을 추출하고 난 후에는 활용성이 상대적으로 크지 않으나 추출한 특징값은 비교적 데이터의 활용 비율이 높게 나타난다. 따라서 대부분의 생체신호로부터 추출된 특징값을 저장하고 활용하기 위해 HIVE, Spark와 같은 데이터웨어 하우스를 포함한 데이터 분석도구를 이용한다[6-8]. 이러한 데이터 저장 및 추출도구는 자체적으로 특별한 형태의 데이터베이스 구조를 사용하거나 Map/Reduce의 형태를 빌어 내부적으로 데이터를 구조화 하여 사용자에게 쉬운 인터페이스를 통하여 액세스 하도록 유도한다는 공통점을 가지고 있다. 결과적으로 사용자는 SQL문과 유사한 구문을 이용하여 특징값을 추출하고 추출된 데이터를 프로그래밍 언어를 이용하여 최종적으로 결과물을 산출한다. 그러나 이러한 방법은 raw-data로부터 추출한 특징값의 데이터 구조가 분석도구에 종속적이기 때문에 데이터의 상호 운용성이 부족하며, 생체신호가 가져야 하는 데이터 호환성에 단점을 가지고 있다.

따라서 본 논문에서는 생체신호 raw-data로부터 추출한 특징값을 표준화된 HL7(Health Level Seven)의 RIM (Reference Information Model)[9] 기반으로 표현하고 저장하며[10], 저장된 특징값을 활용하여 생체신호를 분석하기 위해 수학적 명세 언어인 OCL(Object Constraint Language)를 이용하는 모델을 제안한다. 표준화된 방법으로 생체신호를 저장함으로써 데이터의 호환성 및 상호운용성을 확보할 수 있으며 다양한 형태로의 변환이 가능하다. 또한 수학적 명세언어인 OCL을 사용함으로써 표현이 간결하며 제약조건을 명시로 데이터의 추출이 용이하며 가독성이 우수하다는 장점을 가진다. 본 논문의 구성은 2장 관련연구로서 OCL의 특징 및 기존의 생체신호 특징값 추출 모델에 대해 기술하고, 3장에서는 제안하는 생체신호 분석 모델과 이를 심전도 신호를 예로 하여 모델의 활용 가능성을 확인하고, 끝으로 4장에서는 향후 연구 과제를 포함하여 결론을 맺는다.

## II. 관련연구

### 2.1. OCL

OCL(Object Constraint Language)은 OMG(Object Management Group)에서 표준으로 제정하고 UML (Unified Modeling Language)에서 모델의 구조적 제약사항을 체계적으로 기술하기 위한 정형적 명세언어이다[11]. 기존의 모델에서 가지지 못하는 제약조건을 명시하는데 사용하며, 특히 메타모델 기반으로 작성된 데이터 모델에 적용하여, 객체의 상태를 추출하거나, 그 객체가 가져야할 제약조건을 명시하는 사용한다. 즉 메타모델 기반으로 표현된 데이터에는 OCL을 이용하여 규칙이나 제약사항등을 표현함으로써 의미를 명확하게 전달할 수 있고, 특정한 프로그래밍 언어에 종속적이지 않는다는 장점을 가진다. 더불어 표현법이 풍부하고, 명확하여 가독성이 높아지며, 수정이 간편하고, 표준화를 유도할 수 있다.

본 논문에서는 메타모델 기반의 생체신호 특징값 객체로부터 그 객체의 상태를 가져오고 제약조건을 명시하여 부가적인 기능을 담당하는데 OCL을 사용한다.

## 2.2. 빅데이터 기반 생체신호 분석 모델

현재 빅데이터 기반의 생체신호를 분석하는데 특화된 모델은 그 예를 찾아보기 힘들다. 하지만 기존의 텍스트 마이닝에서 사용하는 모델을 이용하여 생체신호를 분석하는 방법들을 많이 이용하고 있다. 일반적으로 생체신호 raw-data로부터 추출된 특징값은 그림 1과 같은 형태로 저장되고 분석된다. 웨어러블 디바이스 또는 생체신호 측정기로부터 수집된 raw-data를 전처리 및 신호분석 알고리즘을 통하여 특징값을 추출하고, 이를 빅 데이터 플랫폼에 저장한다. 저장된 특징값을 검색하거나 분석하기 위해 별도의 소셜 네트워크 분석 도구인 HIVE, Spark, Pig[12], Drill[13], Impala[14]를 이용한다. 이러한 분석도구들은 자체의 구조를 가진 데이터웨어 하우스에 저장되고, 유사 SQL문을 이용하여 데이터에 액세스한다. 이와 같은 방법은 소셜 네트워크 분석에서 사용하는 일반적인 방법으로 생체신호가 가져야 하는 데이터 호환성의 문제, 특정 플랫폼에 종속적인 데이

터 구조를 가짐으로써 상호운용성을 보장하지 못하는 단점을 가지고 있다.

## III. 생체신호 분석 모델 및 평가

### 3.1. 생체신호 분석 모델

빅데이터 기반의 생체신호 분석모델은 데이터의 상호운용성을 보장하고 데이터의 추출 및 분석이 간단하고 다양한 형태로의 데이터 변환이 가능해야 한다. 따라서 다음의 요구사항을 만족하는 모델을 설계한다.

- ① 생체신호는 표준화된 메타모델 기반으로 표현되어야 한다.
- ② 생체신호는 다양한 형태로 가공 또는 처리할 수 있어야 한다.
- ③ 생체신호의 특징값을 추출하기 위한 방법이 간단하고 명확해야 한다.

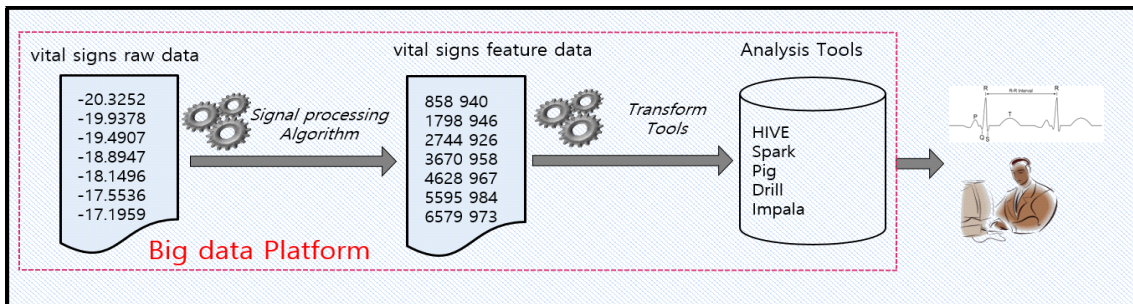


Fig. 1 Vital signs analysis model based on big-data

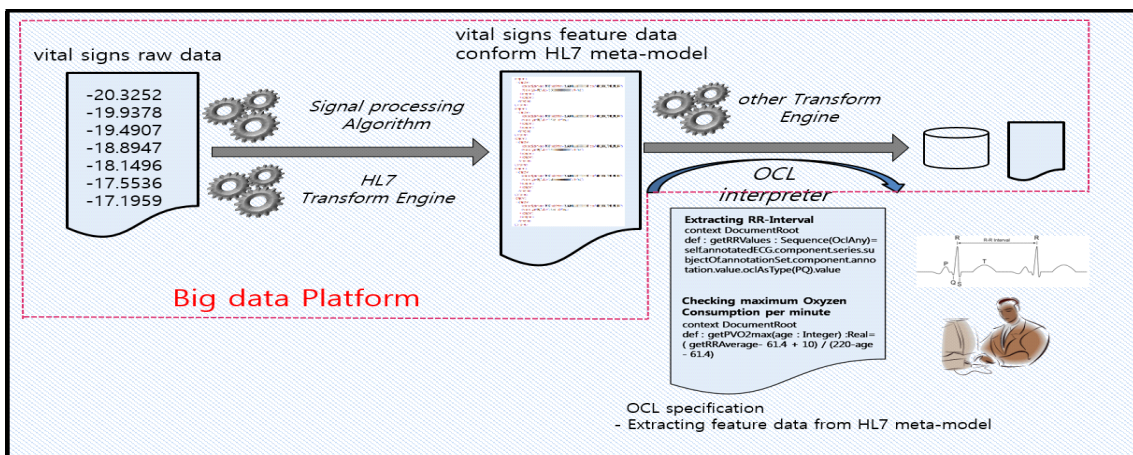


Fig. 2 Vital signs analysis model based on big-data using OCL

④ 특징값을 추출하기 위한 명세는 자체로서 의미를 가짐과 동시에 실행 가능해야 한다.

웨어러블 기기 또는 생체신호 측정기기에서 수집된 생체신호 데이터를 HL7의 표준 메타모델로 표현함으로써 ①②번의 조건을 만족한다. 어떠한 데이터를 표현하기 위해 메타모델을 사용한다는 것은 시스템간의 상호운용성을 보장하고 데이터를 풍부하게 표현할 수 있다. 생체신호의 특징값을 추출하고 분석하는 데에는 OMG(Object Management Group)에서 제정한 표준 명세언어인 OCL을 이용하여 표현함으로써 위 ③④번 요구사항을 만족한다. 따라서 본 논문에서는 그림 2와 같은 빅데이터 기반의 생체신호 분석 모델을 제안한다.

### 3.2. 심전도 신호 분석 모델 평가

이번 장에서는 제안한 생체신호 분석 모델을 평가하기 위하여 심전도(ECG, Electrocardiogram) 데이터를 HL7 메타모델기반으로 표현하고, 심전도 데이터를 기반으로 다양한 지표를 정의하여 적용해 본다. 심전도 신호를 기반으로 진단 가능한 알려진 지표는 HRV(Heart Rate Variability)를 이용하여 최대산소소모량을 예측하고, 이를 칼로리 소모량을 구하는 방법으로 활용하거나, LF/HF(Low Frequency/High Frequency)의 비율을 측정하여 자율신경계에 대한 반응도를 예측하여 스트레스를 평가는 방법으로 활용된다. 본 논문에서는 칼로리 소모량을 측정하기 위해 OCL을 이용하여 지표를 명세하고 적용해 본다.

#### 3.2.1. 심전도 데이터 변환 및 OCL명세

그림 3은 심전도 데이터로부터 R-Peak값의 간격을 의미하는 RR-interval값을 특징값으로 저장하고 이를 HL7의 aECG(Annotated ECG)로 표현하는 과정을 제안한 모델을 기반으로 표현하였다.

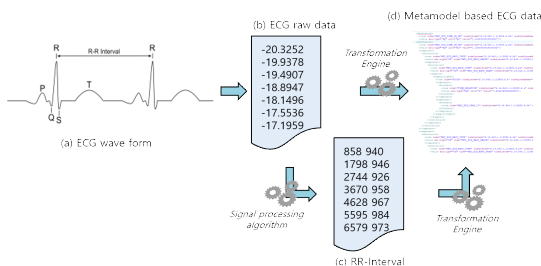


Fig. 3 A Example of ECG signal based on purposed model

(a)에서 유입되는 심전도 신호를 (b)와 같이 raw-data 형태로 빅 데이터기반으로 저장하고 이를 신호처리 알고리즘을 통하여 R-Peak값을 산출하고 (c)와 같은 aECG로 저장한다. 그림 4는 심전도 raw-data를 HL7 메타모델 기반으로 변환하는 소스코드의 일부이다. HL7에서 제공하는 aECG 메타모델을 이용하여 변환한다.

```
import org.hl7.v3.impl.PORTMT020001ComponentImpl;
import org.hl7.v3.util.V3XMLProcessor;

public class AnnotationRR {
    private PORTMT020001Component root;
    private PORTMT020001AnnotationSet ann;

    public AnnotationRR(PORTMT020001AnnotatedECG aecg) {
        aecg.getComponent().get(0).getSeries().
        getDerivation().clear();
        for(Iterator i = aecg.eAllContents(); i.hasNext(); ) {
            EObject o = (EObject)i.next();
            if(o instanceof PORTMT020001AnnotationSet) {
                PORTMT020001AnnotationSet as =
                (PORTMT020001AnnotationSet)o;this.root =
                as.getComponent().get(0);
                this.ann = this.root.getAnnotation();
            }
        }
    }
    public AnnotationRR(PORTMT020001Component root) {
        this.root = root;
        this.ann = root.getAnnotation();
    }

    public PORTMT020001Component getComponent() {
        return this.root;
    }

    public String getRR() {
        PQ v = (PQ)ann.getValue();
        return v.getValue().toString();
    }

    public void setRR(String value) {
        PQ v = (PQ)ann.getValue();
        v.setValue(new BigDecimal(value));
    }
}
```

Fig. 4 Source code for Transformation Engine with java

그림 5는 심전도 raw-data로부터 추출된 RR-interval 특징값이 저장된 aECG기반의 XML문서를 이용하여 최대산소소모량을 구하는 OCL코드이다. 수학적 명세 언어인 OCL을 이용하여 지표를 정의하고 건강상태를 체크함으로써 언제나 변화가능하며 새로운 건강상태를 체크하기 위한 모듈을 추가하기 쉽다. 또한 프로그래밍 언어를 이용한 정의보다 가독성이 높으며, 추가 및 삭제가 용이하고, 재 컴파일 등의 추가적인 작업이 필요하지 않다는 장점을 가진다.

**a) Metamodel based XML Document**

```

<ANNOTATION>
  <code code="MDC_ECG_TIME_PD_RR" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
  <value xsi:type="PQ" unit="ms" value="1.120833333333327"/>
</ANNOTATION>
<ANNOTATION>
  <code code="MDC_ECG_TIME_PD_RR" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
  <value xsi:type="PQ" unit="ms" value="1.158333333333327"/>
</ANNOTATION>
<component>
  <annotation>
    <code code="MDC_ECG_WAVC_TYPE" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
    <value xsi:type="CE" code="MDC_ECG_WAVC_RVAIVE" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
    </annotation>
  </component>
  <annotation>
    <code code="MDC_ECG_WAVC_TYPE" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
    <value xsi:type="CE" code="MDC_ECG_WAVC_PEAR" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
    </annotation>
  </component>
  <supportingROI>
    <code code="ROI_P5" codeSystem="2.16.840.1.113883.5.4" codeSystemName="
    </supportingROI>
  </component>
  <boundary>
    <code code="TIME_RELATIVE" codeSystem="2.16.840.1.113883.5.4" codeSystemName="
    <value xsi:type="PQ" unit="s" value="7.254166666666666"/>
    </boundary>
  </component>
  <boundary>
    <code code="MDC_ECG_LEAD_I1" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
    </boundary>
  </component>
  </supportingROI>
</support>
</annotation>
</component>
</component>
<annotation>
  <code code="MDC_ECG_WAVC_TYPE" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
  <value xsi:type="CE" code="MDC_ECG_WAVC_RVAIVE" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
  </annotation>
</component>
<annotation>
  <code code="MDC_ECG_WAVC_TYPE" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
  <value xsi:type="CE" code="MDC_ECG_WAVC_PEAR" codeSystem="2.16.840.1.113883.6.24" codeSystemName="
  </annotation>
</component>
</component>
  
```

**b) OCL Document**

```

context DocumentRoot
def : getRRValues :
Sequence(OclAny)=
self.annotatedECG.component.series.
subjectOfAnnotationSet.component.
annotation.value.oclAsType(PQ).value

Checking maximum Oxygen
Consumption per minute
context DocumentRoot
def : getPVO2max(age :
Integer) :Real=
( getRRAverage- 61.4 + 10) / (220-
age - 61.4)
  
```

Fig. 5 A Example of OCL code and meta-model based ECG signal

### 3.2.2. 칼로리 소모량 분석을 위한 OCL 명세

심전도 신호를 이용하여 칼로리 소모량을 측정하는 일반적인 방법은 그림 6과 같이 HRV를 이용하여 최대 산소 소모량을 구하고 남성과 여성의 칼로리 소모량 공식에 따라 구할 수 있다. 심전도 데이터를 이용한 최대 산소 소모량으로 칼로리 소모량을 측정하기 위한 그림 6 공식의 지표표를 OCL을 이용하여 표현하면 그림 7과 같이 정의할 수 있다. 그림 7의 OCL 명세 중 `getRRValues` 모듈은 XML로 표현된 HL7의 aECG로부터 RR-interval 특징값을 추출하는 부분이며, `getRRMax`와 `getRRMin` 모듈은 OCL내에서 순차적 데이터 접근 방법과, 조건문

을 사용한 예를 보여준다. 이처럼 OCL로 작성된 모듈은 `sequence`, `iterate`와 같은 데이터 구조, 다양한 데이터 타입, 조건과 기본 연산을 제공함으로써 단순히 데이터를 추출하는 기능을 넘어 하나의 기능을 완벽히 수행하는 독립적인 모듈로서의 기능을 제공한다.

$$\%VO_{2max} = \frac{HRV - HRV_R + 10}{HRV_M - HRV_R}$$

$$Male: CAL = \frac{(3.83t + 13) \times M \times \%VO_{2max} \times t}{0.2}$$

$$Female: CAL = \frac{(3.83t + 13) / 8 \times M \times \%VO_{2max} \times t}{0.2}$$

$\%VO_{2max}$  = Maximal Oxygen consumption per minute  
 HRV = Average HRV  
 HRVR = Average HRV according to age group  
 HRVM = Maximal HRV according to age group  
 M = User's weight in kg  
 t = During exercise for a given period in min

Fig. 6 Estimates calorie burned using HRV

```

package v3
context DocumentRoot
def : getRRValues : Sequence(OclAny) =
self.annotatedECG.component.series.subjectOfAnnotationSet.
component.annotation.value.oclAsType(PQ).value

context DocumentRoot
def : getRRMax(values : Sequence(OclAny)) : Real =
values->iterate(value : Real; answer: Real = values->first() |
if answer < value then value else answer endif )

context DocumentRoot
def : getRRMin(values : Sequence(OclAny)) : Real =
values->iterate(value : Real; answer: Real = values->first() |
if answer > value then value else answer endif )

context DocumentRoot
def : getPVO2max(age : Integer) : Real =
( getRRAverage- getPool + 10) / (getHRVmax(age) - getPool)

context DocumentRoot |
def : getMaleCal(age : Integer, weight : Integer,
kcal : Integer, t : Integer) :
Integer = if ((3.83 * t + 13) * weight *
getPVO2max(age) * t) / 0.2 > kcal * 1000
then t
else getMaleCal(age, weight, kcal, t+1)
endif

context DocumentRoot
def : getFemaleCal(age : Integer, weight : Integer,
kcal : Integer, t : Integer) :
Integer = if ((3.83 * t + 13) / 8 * weight *
getPVO2max(age) * t) / 0.2 > kcal * 1000
then t
else getFemaleCal(age, weight, kcal, t+1)
endif

context DocumentRoot
def : getMomentum(age : Integer, weight : Integer,
kcal : Integer, sex : Integer) :
Integer = if sex = 0 then getMaleCal(age, weight, kcal, 0)
else getFemaleCal(age, weight, kcal, 0) endif

endpackage
  
```

Fig. 7 Definition of calorie burned using OCL

표 1에서는 그림 7에서 표현한 OCL 명세의 일부를 모듈별로 설명하고 있다.

**Table. 1** Explanation of index using OCL

Module name	Explanation
gerRRValues	Extraction of R Peak
gerRRAverage	Calculate of R-Peak average
getPVO2max	value of maximal oxygen consumption
getMaleCal	calculate of male calorie burned
getFemaleCal	calculate of female calorie burned
getMomentum	calculate of calorie burned using weight, age, gender

### 3.2.3. OCL 명세의 실행

그림 8은 심전도 데이터로부터 칼로리 소모량을 측정하기 위해 OCL을 이용하여 작성된 그림 7의 명세를 자바 기반의 OCL Parser API[15]와 OCL Interpreter[16]를 이용한 소스코드의 일부이다. setXMLEnvironment 메서드를 통해 HL7의 aECG파일을 설정하고, setOCLDefDocument 메서드를 이용하여 OCL로 작성된 지표 명세를 파일이름으로 하여 설정하고 최종적인 결과를 얻는다.

```

import v3.util.V3XMLProcessor;
public class OCLEvaluator {
    private OCL ocl;
    private OCLHelper helper;
    private EObject root;

    public void setXMLEnvironment(String xmlFileName) {
        V3XMLProcessor xml = new V3XMLProcessor();
        try {
            Resource resource = xml.load(xmlFileName, null);
            ResourceSet resourceSet = resource.getResourceSet();
            this.setOCLEnvironment(resourceSet);
            root = (EObject)resource.getContents().get(0);
        } catch (IOException e) { e.printStackTrace(); }
    }

    public void setOCLDefDocument(String defName) {
        FileInputStream fileInput = null;
        try {
            fileInput = new FileInputStream(defName);
        } catch (FileNotFoundException e1) { e1.printStackTrace(); }

        OCLInput oclIn = new OCLInput(fileInput);
        try {
            ocl.parse(oclIn);
        } catch (Exception e) {}
    }

    public Object checkQueryInRoot(String query) {
        Object result = null;
        helper.setContext(root.eClass());
        OCLExpression expression;
        try {
            expression = helper.createQuery(query);
            result = ocl.evaluate(root, expression);
        } catch (ParserException e) {
            e.printStackTrace();
        }
        return result;
    }
}

```

**Fig. 8** Java source code for OCL Parser API and OCL Interpreter

## IV. 결론

오늘날 방대한 크기의 생체신호는 빅데이터의 특징을 가지게 되었으며, 생체신호 분석을 위하여 빅데이터의 탄생배경에 기인한 소셜 네트워크 분석 시스템을 차용하는 것이 일반적이다. 하지만 생체신호를 컴퓨터 데이터로 변환하고 이를 활용하는 측면에서 다소 부족한 면이 있다. 분석 시스템마다 서로 다른 포맷을 사용하여 생체 데이터를 저장, 관리하고 있으며, 건강상태를 체크하는 방법과 표현방법이 각기 다르다. 또한 벤더마다의 특성으로 인하여 통합된 분석 환경 구축이 어렵다. 이것은 하나의 시스템에서에서 측정된 생체 데이터, 진단 방법이 다른 시스템에서에서 사용할 수 있는 상호운용성이 부족하며 생체신호가 가져야 하는 데이터 호환성을 보장하지 않는다.

본 논문에서 제안한 생체신호 분석 모델은 생체신호를 HL7 메타모델기반으로 표현함으로써 시스템간의 상호운용성을 보장하고, 생체신호의 형태 및 종류에 따른 각기 다른 시스템을 구축해야 하는 단점을 보완하였다. 또한 생체 데이터를 XML 형식으로 표현함으로써 데이터의 변환이 쉽고, 다양한 뷰어로의 활용을 기대할 수 있다. 그리고 건강상태를 체크하는데 필요한 지표의 정의를 수학적 명세 언어인 OCL을 사용하여 정의함으로써 그 의미가 명확하고, 가독성이 향상된다. 명확한 지표의 정의를 제공함으로써 질병 예방에 관한 정보를 체계적으로 관리하고, 긍정적인 건강 행동의 변화를 유도할 수 있다는 장점을 가진다. 뿐만 아니라 제안한 모델의 효용성을 확인하기 위해 심전도 데이터를 이용하여 이를 HL7의 aECG기반의 표준화된 XML 형태로 저장하고 심전도 기반의 칼로리 소모량을 측정하기 위해 OCL을 이용하여 명세하였다. 또한 OCL 명세는 정의에만 그치지 않고 OCL parser API[15]와 OCL Interpreter[16]를 이용하여 건강상태를 체크하는데 사용됨으로써 시스템의 활용성을 확인하였다.

현재 최대산소소모량을 이용한 칼로리 소모량 측정을 위한 지표 정의에 국한되어 있지만 심전도로부터 측정할 수 있는 다양한 지표를 정의하는 것이 필요하겠다. 또한 호흡신호에 대한 메타모델과 지표를 적용해 봄으로써 효용성을 검증해야겠다.

## References

- [ 1 ] Y. Khan, A. E. Ostfeld, C. M. Lochner, A. Pierre, and A. C. Arias, "Monitoring of vital signs with flexible and wearable medical devices," *Advanced Materials*, vol. 28, issue 22, pp. 4373-4395, June 2016.
- [ 2 ] A. Pantelopoulous, and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, issue 1, pp. 1-12, Jan. 2010.
- [ 3 ] T. W. Kim, S. H. Yi, K. H. Park, and H. C. Kim, "A big data framework for u-Healthcare systems utilizing vital signs," in *2014 International Symposium Computer, Consumer and Control (IS3C) IEEE computer society*, pp. 494-497, June 2014.
- [ 4 ] W. Raghupathi, and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health Information Science and Systems*, vol. 2, no. 3, pp. 1-10, Feb. 2014.
- [ 5 ] J. Y. Choi, "Utilization value of medical Big Data created in operation of medical information system," *Journal of the KIECS*, vol. 10, no. 12, pp. 1403-1410, 2015.
- [ 6 ] C. Vidaurre, T. H. Sander, and A. Schlögl, "BioSig: the free and open source software library for biomedical signal processing," in *Computational intelligence and neuroscience*, pp. 1-12, Mar. 2011.
- [ 7 ] P. E. McSharry, and G. D. Clifford, "Open-source software for generating electrocardiogram signals," in *arXiv preprint physics/0406017*, 2004.
- [ 8 ] M. I. Joo, D. H. Ko, and H. C. Kim, "Development of Smart Healthcare Wear System for Acquiring Vital Signs and Monitoring Personal Health," *Journal of Korea Multimedia Society*, vol. 19, no. 5, pp. 808-817, May 2016.
- [ 9 ] M. Yuksel, and A. Dogac, "Interoperability of medical device information and the clinical applications: an HL7 RIM based on the ISO/IEEE 11073 DIM," in *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 4, pp. 557-566, July 2011.
- [10] S. U. Heo, S. W. Choi, G. H. Kim, and A. S. Oh, "The Study on HL7 Message improvement for Supporting of Efficient Healthcare Service," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 18, no. 6, pp. 1388-1394, June 2014.
- [11] P. Ziemann, and M. Gogolla, "Validating OCL specifications with the use tool an example based on the BART case study," *Electronic Notes in Theoretical Computer Science*, vol. 80, pp. 157-169, Aug. 2003.
- [12] Apache Group. Apache Pig. Project [Internet]. Available: <https://pig.apache.org/>.
- [13] Apache Group. Apache Drill Schema-free SQL Query Engine for Hadoop, No-SQL and Cloud Storage [Internet]. Available: <https://drill.apache.org/>.
- [14] Apache Group. Apache Impala is the open source, native analytic database for Apache Hadoop[Internet]. Available: <https://impala.apache.org/>.
- [15] Eclipse Platform. Parsing OCL Documents[Internet]. Available:<https://help.eclipse.org/kepler/index.jsp?topic=%2Forg.eclipse.ocl.doc%2Fhelp%2FParsingDocuments.html>.
- [16] Eclipse Platform. Working with the Classic OCL, Parsing OCL Expressions[Internet]. Available: <https://help.eclipse.org/luna/topic/org.eclipse.ocl.doc/help/OCLInterpreterTutorial.html>.



김태웅(Tae-Woong Kim)

2009년 인제대학교 전산학과 전산학박사

현재: 신라대학교 컴퓨터교육과 조교수

※관심분야: 소프트웨어 공학, 역공학, 리팩토링, 헬스케어 시스템, 빅 데이터