
XML을 활용한 멀티모달 센서기반 실시간 컨텍스트 통합 시스템

A Real-time Context Integration System for Multimodal Sensor Networks using XML

양성익, Sungihk Yang*, 홍진혁, Jinhyuk Hong**, 조성배, Sung-Bae Cho***

요약 ~ 최근 유비쿼터스 환경에서의 다양한 서비스에 대한 연구가 활발히 진행되고 있다. 이러한 서비스들은 사용자의 컨텍스트를 해석하는 것이 중요하며, 이를 위해 PDA, GPS, 가속도 센서 등 다양한 센서들이 활용되고 있다. 가속도 센서와 같은 저급의 연속적인 데이터를 수집하는 센서는 그 데이터를 직접 활용하는 것이 어려우므로 전처리하는 것이 중요하다. 하지만 실시간으로 전처리하고 컨텍스트를 통합하는 시스템에 대한 연구가 미흡하다. 본 논문에서는 XML을 활용하여 가속도 센서와 생리적 신호 센서 등과 같은 저급의 데이터를 전처리하고 컨텍스트를 통합하는 시스템을 제안한다. XML을 통하여 센서 네트워크의 구조를 정의하고, 각 센서별로 수집되는 데이터와 데이터에 해당하는 컨텍스트의 룰을 정의할 수 있다. 제안하는 시스템은 XML을 통하여 센서 네트워크에 센서를 추가할 때와 컨텍스트의 룰을 수정할 때 소스의 수정을 줄여준다. 그리고 이 시스템을 활용하여 실시간으로 데이터를 모니터링하는 시스템을 구현하여 난해한 데이터의 변화를 그래프로 표현해 데이터의 검증에 도와주며, 실시간으로 전처리의 결과를 확인하여 변화가 가능하고, 다른 외부 서비스나 어플리케이션에 사용자의 컨텍스트의 제공을 용이하게 해준다.

Abstract ~ As the interest about ubiquitous environment is increasing, there are many researches about the services in this environment. These services have important issues in interpreting the users' context, using many kinds of sensors, like PDA, GPS and accelerometers. Low level raw data, which sensors like accelerometers calibrates, are hard to use, and to provide real-time services preprocessing and interpreting the data into context, in real-time, is important. This paper describes a context integrate system which can integrate these sensors and also sensors which has raw data, like accelerometers and physiological sensors, and define the context interpret rule with XML. The proposing system reduces programming operations when adding a sensor to the sensor network or modifying the context interpreting rule by using XML. By using this system, we implemented a real-time data monitoring system which can describe the numeric data into graphs, and assist the user to validate the data and results of the preprocess phase, and also support the external services and applications to use the context of the user.

핵심어: context-aware, real-time integrate system, preprocessing, activity recognition

*주저자 : 연세대학교 컴퓨터과학과 석사 e-mail: unikys@sclab.yonsei.ac.kr

**공동저자 : 연세대학교 컴퓨터과학과 박사 e-mail: hjinh@sclab.yonsei.ac.kr

***교신저자 : 연세대학교 컴퓨터과학과 교수; e-mail: sbcho@yonsei.ac.kr

1. 서론

최근 유비쿼터스 환경에서 다양한 컴퓨팅 자원을 활용하여 사용자의 컨텍스트를 기반으로 하는 인간과 컴퓨터 상호작용 기반의 서비스에 대한 연구들이 활발히 진행되고 있다 [1]. 이런 서비스들은 사용자 중심으로 이루어져 있으며, 이를 위해서 사용자의 현재 상태, 위치, 행동 그리고 감정 등을 인식하는 것이 중요하다. 이를 위해 가속도 센서와 생리적 신호 센서 등 다양한 센서가 필요한데, 이 센서들의 데이터 특성도 다양하다. 기존에 실시간으로 데이터를 전처리하는 시스템에 대한 연구[2]가 많이 진행되어왔지만, 고정적인 센서 네트워크에 대한 고려만 해주고 있다. 그리고 센서를 활용하는 서비스를 개발하거나 데이터를 수집할 때, 센서가 수집하고 있는 값은 직접 모니터링하기 난해한 수치 데이터가 많기 때문에 어플리케이션을 개발할 때 어려움이 있다. 특히, 데이터에 의존적인 행동 인식이나 컨텍스트를 활용하는 서비스를 제공할 때, 기존의 일괄처리 통합시스템을 활용할 경우, 데이터에 대한 검증은 데이터를 수집하고 전처리와 특징 추출에 대한 파라미터를 설정하고 시스템의 결과를 확인한 후, 다시 데이터를 수집하거나 전처리와 특징 추출을 반복해야하는 어려움이 있다.

본 논문에서는 저급의 센서 데이터를 실시간으로 통합하여 다양한 컨텍스트로 해석하고 응용 어플리케이션과 서비스에 제공하는 컨텍스트 통합 시스템을 제안한다. 제안하는 컨텍스트 통합 시스템은 XML의 정의를 통해 고정적인 센서 네트워크에서 소스의 수정을 최소화하며 센서의 추가를 용이하게 하고, 이 시스템을 활용한 실시간 모니터링 시스템의 구현을 통하여 실시간으로 수집되는 데이터를 시각적인 그래프로 보고, 전처리의 파라미터 수정 결과를 실시간으로 확인하여 센서로 수집된 데이터에 대한 검증을 할 수 있도록 한다.

2. 배경

2.1 센서 네트워크를 통한 서비스

인간과 컴퓨터 상호작용을 기반으로 한 서비스들 중 사용자의 행동 인식을 통한 서비스와 컨텍스트 기반의 서비스가 대표적이며, 그 중 사용자 행동 인식에 대한 다양한 연구가 활발히 진행되고 있다. 대부분이 가속도 센서를 기반으로 하고 있고[3], 생리적 신호를 기반으로 행동을 인식하는 연구[4]나 GPS를 활용하는 연구[5]도 진행되었다. 행동을 인식하기 위해 다양한 센서들이 활용되는데, 다양한 센서를 활용하여 컨텍스트를 해석하고 행동을 인식을 위해 가속도 센서와 생리적 신호 센서 등이 활용되며, 이들로부터 수집되는 저급의 데이터는 잡음이 많고 연속적인 값을 그대로 활용하기 어렵기 때문에 전처리 과정이 중요하며, 서비스를 제공하기

위해서는 실시간으로 사용자의 동작 정보를 수집하고 전처리 및 특징추출 하는 것을 고려해야 한다. 각 센서에서 수집되는 데이터의 특성도 센서 별로 다르기 때문에 컨텍스트 해석할 때 센서의 들어오는 수치의 크기와 변화량을 고려해야 한다. 그림 1은 본 논문에서 활용하고 있는 센서들로 가속도 센서와 생리적 신호 센서 및 환경 센서를 보여준다.



그림 1. 가속도센서(좌측), 생리적신호 센서(중앙), 환경 센서(우측)

2.2 데이터 전처리 및 컨텍스트 통합 시스템

컨텍스트란 사용자와 어플리케이션 사이의 상호작용에 고려되어야 할 사람, 장소, 사물의 상황을 설명하는 정보로 정의된다[6]. 컨텍스트를 추출하기 위해 다양한 센서들이 활용되며, 각 센서들을 통해 추출된 컨텍스트를 통합하는 시스템에 대한 연구가 활발하다. Lieberman의 연구에서는 PDA와 GPS를 이용하여 사용자의 컨텍스트를 인공지능을 활용하여 정의하고 효율적으로 관리하는 시스템인데[7], 실시간 환경에서의 서비스에 대하여 고려하지 않고 있다. Raento의 연구는 Lieberman의 연구보다 다양한 센서를 사용해서 실제 환경에 적용하기 적합한 시스템이다[8]. 이 두 가지 연구들은 사용자의 현재 위치와 주변 환경에 대한 정보를 획득할 수 있지만, 사용자 개인에 대한 신체 행동이나 변화에 대하여 정보를 얻을 수 없는 약점이 있으며, 가속도 센서 등과 같이 저급의 데이터 형태를 가지는 센서의 통합을 고려하고 있지 않다. 따라서, 본 논문에서는 그림 2와 같이 XML을 활용하여 컨텍스트 통합시스템의 구조를 정의하고, 멀티모달의 센서 네트워크에 대하여 센서를 통합할 수 있으며, 내부에서 전처리 및 특징 추출을 한 뒤, 행동 인식하고 이 결과들을 서비스나 어플리케이션에서 활용하는 시스템을 제안한다.

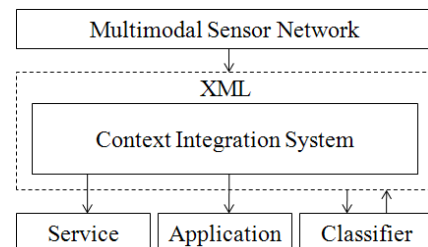


그림 2. 제안하는 시스템 개요

2.3 실시간 센서 모니터링 시스템

센서 모니터링 시스템은 센서로 수집된 데이터와 서비스를 구현할 때 원하는 동작에 대한 검증을 도와준다. Hartmann은 패턴 인식을 위한, 실시간으로 수집된 데이터

를 수정하고 점검할 수 있는 센서와 상호작용할 수 있도록 하는 시스템을 제안했고[9], Dey는 동적 베이지안 네트워크를 분류기로 활용하여, 실시간으로 분류기를 학습하고 테스트 가능한 시스템을 제안하였다[10]. 이 연구들은 개발자가 프로토타이핑이 용이한 시스템을 제안하는 연구들로, 수집한 데이터의 시간적인 모니터링을 통한 편리함을 제공하지만, 고정된 센서네트워크와 전처리 방식나 분류기의 선정이 고정되어 있으며, 확장성이 떨어진다. 본 논문에서 제안하고 있는 컨텍스트 통합 시스템을 활용하여 확장성 있는 실시간 센서 모니터링 시스템을 제안한다.

3. 행동 인식을 위한 컨텍스트 통합 시스템

본 논문에서는 기존의 연구들과 달리 연속적인 저급의 데이터를 다루는 센서들을 통합하며, 실시간으로 이를 전처리하고 컨텍스트를 통합하는 시스템을 제안한다. 이 시스템은 다양한 형태의 센서를 통합하기 위하여 XML로 컨텍스트 통합시스템의 구조를 정의하며, 사용자의 컨텍스트를 수집할 수 있다. 통합하는 다양한 센서들은 실시간으로 데이터를 수집하기 위해 인터페이스별 변경이 가능한 입력 모듈이 있으며, 이 모듈을 통해 실시간으로 수집된 데이터를 통합하고, 전처리 및 특징 추출을 하여 가공된 데이터로 변환한다. 이 데이터는 사용자의 행동 및 감정 인식 모듈에 제공하게 되고, 그 인식 결과는 기존의 컨텍스트에 추가한다. 이렇게 가공된 데이터를 통하여 해석할 수 있는 컨텍스트와 행동 및 감정 인식 결과로 만들어진 사용자 컨텍스트는 다른 어플리케이션과 서비스에서 활용할 수 있다. 그림 3은 전체 시스템의 흐름도를 나타낸다. 센서를 통해서 들어온 데이터는 먼저 통합 모듈로 통합하게 되고, 각 전처리 모듈과 특징 추출 모듈은 데이터를 교환하며 데이터를 가공한 뒤, 컨텍스트 해석 모듈에서 컨텍스트를 해석한다. 이 컨텍스트는 외부 모듈로 있는 행동 인식 모듈에서 활용 하게 된다.

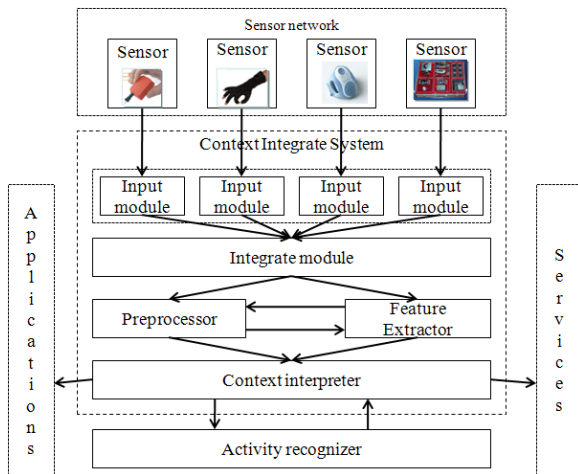


그림 3. 제안하는 컨텍스트 통합 시스템의 흐름도

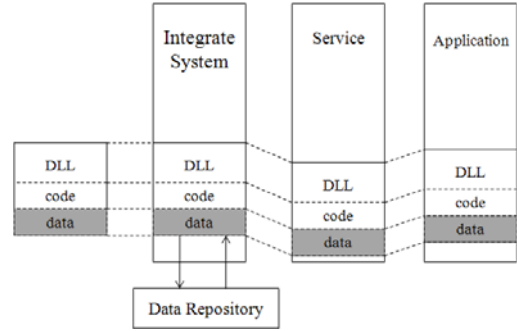


그림 4. 데이터 공유 방식

제안하는 시스템은 센서와 개발자의 상호작용을 위하여 컨텍스트 통합 시스템을 모듈화하여 상호작용을 하며 센서 네트워크의 데이터 통합 및 컨텍스트 해석을 용이하게 제공한다. 컨텍스트 통합 시스템은 동적 라이브러리를 통해서 데이터 정보를 공유하여 여러 개의 서비스와 어플리케이션에서 동시에 접근하여 사용 가능하다. 개발자는 그림 4와 같이 서비스와 어플리케이션에 적합한 행동 인식 모듈을 설계하고 실시간으로 센서 데이터 값과 컨텍스트 해석 결과 그리고 행동 인식의 결과를 라이브러리의 공유 데이터로 접근 가능하다. 공유 데이터는 컨텍스트 통합 시스템만 수정할 수 있고 서비스와 어플리케이션에서는 참조만 할 수 있고 수정이 불가능하다.

3.1 컨텍스트 통합시스템을 위한 XML

제안하는 시스템에서는 XML(eXtensible Markup Language)을 통하여 사용자가 센서 네트워크 및 컨텍스트의 룰을 정의하여 제안하는 컨텍스트 통합시스템을 구조화해서 정의할 수 있다. 먼저, 센서 네트워크를 정의함으로써 각 센서 데이터를 수집하는 라이브러리와 XML을 통한 정의를 통해서 추가가 가능하다. 센서 수집하는 라이브러리는 RS232 시리얼 통신을 이용하는 경우와 콘솔을 통하여 입력 받는 경우 그리고 그 외의 경우로 나누어서 정의할 수 있다. RS232 시리얼 통신을 활용하는 경우에는 시리얼 포트를 통하여 센서를 초기화하는 매크로 명령어와 입력되는 포맷 그리고 센서의 동작을 멈추는 매크로 명령어로 나누어서 설정이 가능하고, 콘솔을 통하여 입력받는 경우에는 프로세스 실행 명령과 실행할 경우 초기에 뜨는 문자열 그리고 입력되는 포맷을 설정하여 데이터를 수집 할 수 있다. 그 외의 경우에는 다른 제공하는 SDK(Standard Develop Kit)을 통하여 데이터가 수집 되는 경우나 센서가 아닌 다른 장치를 통해 수집하는 경우 등이 해당되며, 초기화하는 함수와 센서 데이터를 가져오는 함수 그리고 센서를 종료시키는 함수를 정의할 수 있으며, 이는 동적 연결 라이브러리를 통하여 연결 된다. 사용자는 이렇게 XML을 통하여 센서 네트워크를 정의하고 센서 수집하는 라이브러리의 작성만 하면 쉽게 센서 통합 시스템에 추가할 수 있다. 그림 5는 센서의 종류에 대하여 처리하는 흐름도이다. 각 센서의 XML의 파싱 결과

에 따라 입력 모듈이 결정되며, 기타 입력 모듈(Other)의 경우에는 통합시스템과 다른 SDK나 다른 수집 프로그램의 중간에서 매개체 역할을 하여 데이터를 주고 받는다.

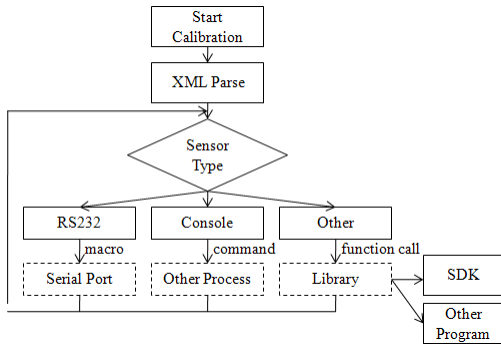


그림 5. 센서 측정 흐름도

또한, 사용자는 XML을 통해 각 데이터 형과 컨텍스트의 룰을 정의를 할 수 있다. 각 센서의 데이터 형을 정의함으로써 멀티모달 센서 네트워크에 대한 확장성을 고려하고 있다. 데이터 형은 숫자형과 그 외의 경우로 나누어서 저장하게 되며, 숫자형의 경우 전처리 단계와 컨텍스트의 룰을 정의할 수 있다. 전처리 단계의 정의는, 먼저 전처리라이브러리에서 해당하는 전처리의 입력하는 함수를 정의하고, 각 센서의 데이터 별로 전처리할 단계들을 정의하여 사용하게 된다. 순서와 상관없이 전처리하기 위해 그림 6과 같은 프로토콜을 정해서, XML로 정의된 순서에 따라 라이브러리에서 함수명을 호출한다. 컨텍스트의 룰을 정의는 룰로 "크다", "작다", "중간에 있다", "같다", "같지 않다"를 활용하고 있으며, 각 룰에 대하여 비교하는 값을 설정할 수 있다. 해당하는 상황이 참이 되면 같이 정의한 컨텍스트로 설정 된다.

```
fnPreprocess(const float* orgData,
             int& orgDataSize,
             float* &objData);
```

그림 6. 전처리 프로토콜

이렇게 사용자가 정의할 수 있는 XML은 그림 7과 같은 문서형 정의(DTD, Document Type Definition)을 따르고 있다. 그림 7의 DTD를 적용한 XML의 예제는 그림 8과 같이 나타낼 수 있다. 그림 8은 MTx 가속도 센서에 대한 정의를 요약한 내용으로 센서 라이브러리는 "MTx.dll" 파일을 열고 초기화, 측정 및 종료 함수를 정의하고 있다. 데이터는 데이터명과 형식 그리고 데이터의 인덱스의 정보와 함께 하위 정보로 컨텍스트에 대한 정의를 가지고 있다. 컨텍스트는 rule과 함께 비교할 값을 가지고 있다. 예제에서는 "AccX"에 대한 컨텍스트는 7보다 클 경우(rule="gt")에 "fast", 3과 7 사이에 있는 경우(rule="bt")에 "normal", 그리고 3보다 작을 경우(rule="lt")에는 "slow"가 부여되도록 정의한 것이다.

```
<?xml version="1.0" encoding="euc-kr"?>
<ELEMENT integratesystem (sensorlist , preprocesslist)>
<ELEMENT sensorlist (sensor)+>
<ELEMENT sensor ((console|rs|other) , datalist)>
<ELEMENT console (consolefailstring)*>
<ELEMENT consolefailstring (#PCDATA)>
<ELEMENT rs (#PCDATA)>
<ELEMENT other (#PCDATA)>
<ELEMENT datalist (data)+>
<ELEMENT data ((preprocessid)*,(context)*)>
<ELEMENT context (#PCDATA)>
<ELEMENT preprocesslist (preprocess)+>
<ELEMENT preprocess (#PCDATA)>
<ELEMENT preprocessid (#PCDATA)>

<!ATTLIST sensor name CDATA #REQUIRED>
<!ATTLIST sensor library CDATA #REQUIRED>
<!ATTLIST sensor type CDATA #REQUIRED>
<!ATTLIST sensor port CDATA #REQUIRED>
<!ATTLIST other startfunction CDATA #REQUIRED>
<!ATTLIST other calibrationfunction CDATA #REQUIRED>
<!ATTLIST other endfunction CDATA #REQUIRED>
<!ATTLIST data name CDATA #REQUIRED>
<!ATTLIST data type CDATA #REQUIRED>
<!ATTLIST data index CDATA #REQUIRED>
<!ATTLIST console header CDATA #REQUIRED>
<!ATTLIST console format CDATA #REQUIRED>
<!ATTLIST rs startcommand CDATA #REQUIRED>
<!ATTLIST rs terminatecommand CDATA #REQUIRED>
<!ATTLIST rs format CDATA #REQUIRED>
<!ATTLIST context id CDATA #REQUIRED>
<!ATTLIST context name CDATA #REQUIRED>
<!ATTLIST context rule CDATA #REQUIRED>
<!ATTLIST context value1 CDATA #REQUIRED>
<!ATTLIST context value2 CDATA #REQUIRED>
<!ATTLIST preprocess functionname CDATA #REQUIRED>
<!ATTLIST preprocess id CDATA #REQUIRED>
```

그림 7. 컨텍스트 통합시스템을 위한 XML의 문서형 정의

```
<?xml version="1.0" encoding="euc-kr"?>
<!DOCTYPE integratesystem SYSTEM "Sensor.dtd">
<integratesystem>
  <sensor name="MTx" library="MTx.dll" type="other" port="6">
    <other startfunction="InitMTx"
           calibrationfunction="GetData"
           endfunction="EndMTx">
    </other>
    <datalist>
      <data name="AccX" type="float" index="0">
        <context id="2" name="Fast" rule="gt" value1="7"/>
        <context id="1" name="Normal" rule="bt" value1="3"
                value2="7"/>
        <context id="0" name="Slow" rule="lt" value1="3"/>
      </data>
      .....
    </datalist>
  </sensor>
  .....
</integratesystem>
```

그림 8. XML의 정의 예제

3.2 멀티모달 센서를 위한 입력 모듈과 데이터 통합

다양한 입력 방식의 센서들을 통합하기 위해서는 다양한 입력 모듈이 필요하다. 본 논문에서는 센서 네트워크에서 많이 활용되고 있는 가속도 센서나 생리적 신호 센서들과 같이 RS232 통신 포트와 특정한 콘솔을 통해 입력하는 센서들을 통합하는 것 뿐만 아니라 사진 등과 같은 수치로 표현할

수 없는 센서의 데이터에 대해서도 데이터의 통합과 확장이 가능하도록 지원하며, 다양한 데이터 형식이 있는 멀티모달 센서 네트워크를 지원하기 위해 로그 데이터는 수치형 데이터와 그 외의 데이터별로 나누어서 연결리스트 방식으로 저장하므로, 다양한 데이터형을 동적으로 통합하는 것이 가능하다.

이렇게 각 센서의 데이터가 수집되면 그 데이터를 바로 사용하기 어렵다. 저급의 순수 데이터는 잡음이 포함되어 있거나 데이터의 크기가 클 수도 있으며 연속적인 데이터 값을 의미적인 이산적인 값으로 변환하기 위해서는 실시간으로 들어오는 데이터를 전처리하여 활용해야 한다. 이렇게 실시간 전처리를 하기 위해서는 실시간으로 입력되는 각 데이터 마다 결과가 하나씩 나와야하며, 통계치를 활용하기 위해서는 이전에 수집한 데이터를 바탕으로 초기화 시켜야한다. 제안하는 시스템에서는 초기화 되어있는 파라미터 값들을 실시간으로 수정하는 것이 가능하며, 전처리한 데이터는 다시 컨텍스트로 해석하거나 행동 인식하는 분류기에서 활용할 수 있다.

3.3 실시간 컨텍스트 해석 및 행동 인식

사용자가 센서의 데이터에 대하여 룰을 설정해서 컨텍스트를 해석하며, 각 센서의 데이터에 따라 그 룰을 각각 달리한다. 컨텍스트의 룰은 임계치나 데이터의 변화 값 또는 변화 추이를 활용하여 컨텍스트를 해석할 수 있다. 컨텍스트는 행동의 패턴 인식이나 다른 서비스나 어플리케이션에서 컨텍스트 통합 시스템의 라이브러리를 호출하여 활용 가능하다.

행동 인식을 위한 분류기는 외부 모듈을 호출해서 사용하며, 모듈의 내부를 수정하면 다른 분류기나 알고리즘을 활용하여 다른 결과를 얻을 수 있다. 외부 모듈을 사용하기 때문에 분류기에 대한 수정은 컨텍스트 통합 시스템에 아무런 영향을 미치지 않으며, 행동 인식을 통한 분류 결과를 다시 입력으로 받아서 컨텍스트에 추가한다. 컨텍스트에 추가한 행동은 다시 다른 분류기에서 사용하여 증기변수로 활용 가능하다. 행동 인식을 위한 분류기와의 연동은 통합 시스템에서 실시간으로 수집된 데이터를 분류기로 보내어 실시간으로 학습 데이터를 수집하거나 파일을 통해서 분류기를 학습시킨 뒤, 실시간의 테스트 데이터를 분류기에 적용시키는 방식이다.

4. 응용 어플리케이션

4.1 실시간 데이터 모니터링 시스템

본 논문에서 제안하고 있는 컨텍스트 통합시스템을 활용하여 실시간으로 모니터링 가능한 응용 어플리케이션을 구

현하였다. 이 어플리케이션은 센서를 통해 수집되는 데이터 별로 원하는 수치를 모니터링 하도록 지원해준다. 모니터링 하는 센서 데이터는 전체리 및 특징 추출의 결과를 동시에 보여주어 센서의 입력 값에 대해 모니터링 할 수 있다. 모니터링하고 있는 데이터를 레이블링하여 파일로 저장할 수 있다.

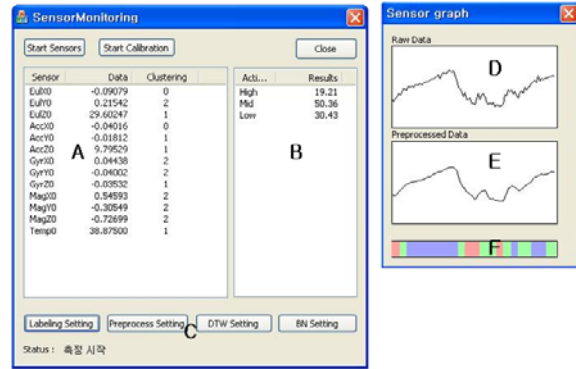


그림 9. 메인 실행화면(좌측)과 모니터링 화면(우측)

그림 9는 본 논문에서 제안하고 있는 컨텍스트 통합 시스템을 활용하여 구현한 실시간 모니터링 시스템의 화면이다. 좌측의 화면이 센서 통합하는 메인 화면으로 A에는 센서의 아이디와 값을 나타내고 있고, 클러스터링을 적용 시키면 클러스터링의 결과를 동시에 보여준다. B에는 실시간으로 행동 인식 분류기를 적용시킨 결과를 보여주는 리스트이며, 현재는 베이지안 네트워크를 이용했다. C에 있는 버튼들은 설정 화면을 띄우는 설정 메뉴이다. 우측의 화면은 실시간으로 들어오는 데이터 값을 모니터링하는 화면으로 A에서 보고자 하는 센서의 아이디를 선택해서 더블 클릭하면 원하는 센서 만큼 새로운 창을 띄워서 동시에 여러 개의 센서를 모니터링할 수 있도록 지원한다. D는 들어온 순수한 데이터를 보여주는 그래프이고 E는 순수 데이터를 전처리한 그래프로서의 그래프에서 평활화가 이루어진 것이 확인 가능하다. F는 룰 기반으로 컨텍스트를 해석한 결과로 파란색은 증가, 빨간색은 감소, 초록색은 값의 유지를 나타내는 그래프이다.

4.2 실시간 파라미터 수정 및 데이터 검증

제안하는 시스템을 활용한 응용 어플리케이션은 전처리의 처리 파라미터들을 실시간으로 수정하고 그 결과를 모니터링 시스템을 통해서 바로 검증이 가능하다. 센서 데이터로 들어오는 값의 변화에 대한 결과가 비교 가능하고, 전처리 및 특징 추출 파라미터 수정으로 인한 결과의 변화도 실시간으로 검증 가능하며, 실시간으로 행동 인식의 결과가 확인 가능하다.

그림 10은 각 설정 화면을 나타내는 것으로 (a)는 Dynamic time warping알고리즘을 활용하여 실시간 패턴 인식하는 화면이다. A는 실시간으로 학습시킨 패턴의 모양을 나타내고 있으며, B는 패턴 인식의 결과로 나타내 레이

블링한 값을 나타내준다. (b)는 전처리의 파라미터를 실시간으로 설정하는 화면으로 C에 있는 슬라이드바를 수정하면 그 결과가 적용되어 그림 9의 우측 그래프에 바로 나타내게 된다. (c)는 베이지안 네트워크를 설정하는 화면으로 추론의 주기를 설정하고 파일로의 학습과 실시간 학습의 선택 가능하다. (d)는 레이블링 화면으로 데이터 수집을 용이하게 제공한다.

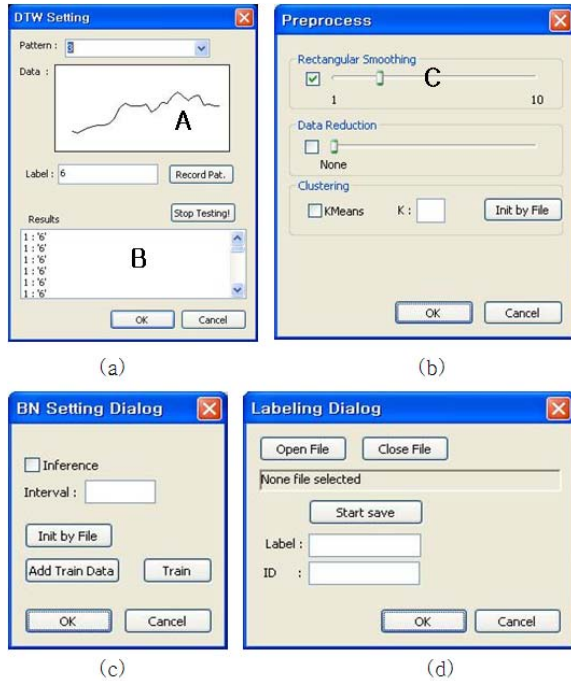


그림 10. (a) DTW설정화면, (b) 전처리 설정화면, (c) 베이지안 네트워크 설정화면, (d) 레이블링 설정화면

5. 결론

본 논문에서는 멀티모달 센서 네트워크의 통합을 하며, 데이터들을 실시간으로 전처리를 하고 컨텍스트를 해석하는 시스템을 제안했다. 제안하는 시스템은 XML을 기반으로 하여 컨텍스트 통합시스템을 구조화하여 정의할 수 있도록 하여, 통합시스템에 센서를 추가할 때와 각 데이터에 대한 전처리 단계와 컨텍스트 룰을 정의하여 데이터 가공을 할 때 소스의 수정을 줄여주고 있다. 제안하는 시스템은 센서에서 저급의 데이터를 수집하고 통합한 뒤, 가속도 센서, 생리적 신호 센서와 환경 센서 등 다양한 센서들을 통합하여 행동 인식 등과 같은 서비스에서 활용이 용이하도록 지원해주며, 행동 인식의 결과를 사용자 컨텍스트로 활용할 수 있도록 한다. 또한, 제안하는 시스템을 활용하여 실시간 모니터링 어플리케이션을 구현하여, 실시간으로 데이터의 전처리 파라미터를 설정하고 그 변화를 확인할 수 있으며, 실시간으로 행동 인식 결과를 확인할 수 있는 시스템을 보였다.

향후에는 컨텍스트의 해석 룰을 실시간으로 정의할 수 있도록 해주며, 컨텍스트 생성 룰을 다양화 시키고 컨텍스트의 조합을 통한 복합 컨텍스트의 생성을 지원해줘서 컨텍스트 통합시스템의 활용도를 높이고, 전처리 모듈의 다양화와 그 전처리들에 대한 파라미터 설정을 XML로 정의할 수 있도록 할 것이다.

참고문헌

- [1] J. Kolari, T. Laakko, T. Hiltunen, V. Ikonen, M. Kulju, R. Suihkonen, S. Toivonen and T. Virtanen, "Context-aware services for mobile users," *Technology and User Experiences, Espoo 2004*, VTT publications 539, 2004.
- [2] P. Nurmi, M. Przybilski, G. Linden and P. Floreen, "An architecture for distributed agent-based data preprocessing," *Autonomous Intelligent Systems: Agents and Data Mining*, pp. 123~133, 2005.
- [3] N. Ravi, N. Dandekar, P. Mysore and M. Littman, "Activity recognition from accelerometer," *Proc. 17th Innovative Applications of AI Conf.*, pp. 11~18, 2005.
- [4] S. Yang and S. Cho, "Activity recognition based on physiological signals with fuzzy Bayesian networks," *Proc. of the 34th KIISE Fall Conference*, vol. 34, no. 2(A), pp. 72~73, 2007.
- [5] A. Subramanya, A. Raj, J. Bilmes, and D. Fox, "Recognizing activities and spatial context using wearable sensors," In *Proc. of Conf. on Uncertainty in Artificial Intelligence*, no. 53, 2006.
- [6] A. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4~7, 2001.
- [7] H. Lieberman and T. Selker, "Out of context: Computer systems that adapt to, and learn from, context," *IBM Systems Journal*, vol. 39, no. 3&4, pp. 617~632, 2000.
- [8] M. Raento, A. Oulasvirta, R. Petit and H. Toivonen, "ContextPhone: A prototyping platform for context-aware mobile applications," *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 51~59, 2005.
- [9] B. Hartmann and S. Klemmer, "Reflective physical prototyping through integrated design, test, and analysis," *ACM Symposium on User Interface Software and Technology*, pp. 299~308, 2006.
- [10] A. Dey, R. Hamid, C. Beckmann, I. Li and D. Hsu "a CAPpella: Programming by demonstration of context-aware applications," *Proc. of CHI 2004*, pp 33~40, 2004.