
USN 응용 서비스를 위한 이벤트 및 서비스 레지스트리 시스템

염성근* · 김용운** · 유상근** · 김형준** · 정희경*

Event and Service registry system for USN Application Services

Sung-Kun Yeom* · Yong-Woon Kim** · Sang-Keun Yoo** · Hyung-Jun Kim** · Hoe-Kyung Jung*

요 약

유비쿼터스(Ubiquitous) 환경에서는 다양한 공간에 RFID, 센서 네트워크 등으로부터 지속적이고, 주기적으로 사용자 및 사물에 관한 다양하고 방대한 양의 센서 데이터 정보가 발생한다. 이러한 센서 서비스를 제공하는 수많은 제공자들 중에서 사용자는 자신이 원하는 서비스를 찾을 수 있어야 하며, 해당 서비스를 사용 시, 사용자의 요구에 맞는 서비스를 제공받아야 한다. 그러나 현재의 USN(Ubiquitous Sensor Network) 응용 서비스 환경에서는 이런 일련의 서비스들을 검색할 수 있는 레지스트리가 거의 없으며, 있더라도 비즈니스 위주의 UDDI(Universal description discovery, and integration)만이 존재한다. 또한, 서비스를 사용 시 사용자의 요구에 맞는 서비스를 제공해야 하지만, 현재의 서비스들은 하나의 조건 또는 이벤트에 따른 서비스만 제공한다. 따라서 여러 조건이나 이벤트에 따라 서비스를 제공할 수 있는 률 기반의 서비스 이벤트 처리 시스템이 필요하다.

이에 본 논문에서는 기존 UDDI를 참고한 센서 서비스 검색을 위한 서비스 레지스트리 시스템 및 WS-ECA(Web Service Event Condition Action)와 같은 웹서비스(Web Service) 기반의 이벤트 룰을 참고한 USN 응용 서비스를 위한 이벤트 처리 시스템에 대한 연구를 하였다.

ABSTRACT

In ubiquitous environment, various and amount of sensor data information occur continuously and periodically about thing and user from RFID, sensor network of various space. Such, the users from among many of provider who provide sensor service should find service that themselves is wanted. And when service use, service right to user should be provided. But in the current environment of the USN(Ubiquitous Sensor Network) Application Service, there is no the registry to find these services but only business-oriented UDDI(Universal Description Discovery, and integration). In addition, when service use, service right to user should be provided but current services provided only services one of the conditions or events. Therefore, rules-based processing system of service event that can provide service according various conditions and events is need.

In this paper research into service registry for sensor service search with reference to existing UDDI and event processing system for USN application service with reference to event rule based on web service as WS-ECA(Web Service-Event Condition Action).

키워드

Ubiquitous Computing, USN, UDDI, WS-ECA, WS-Eventing, USR

* 배재대학교 컴퓨터 공학과 (교신저자:정희경)

** 한국전자통신연구원

접수일자 2009. 01. 08

심사완료일자 2009. 02. 16

I. 서 론

최근 들어 USN은 미래 유비쿼터스 사회를 구현하는 핵심 인프라로서 대두되고 있다.

이에 기존 연구에서는 측정 데이터와 서비스를 표준화된 형태로 주고 받을 수 있도록 XML(eXtensible Markup Language) 기반의 센서 서비스 기술 언어를 정의하였다. 또한, 다양한 센서 서비스를 불특정 다수의 사용자가 다양한 응용 환경에서 센서 데이터를 표준화된 프로토콜을 사용하여 서비스 받을 수 있도록 웹 서비스를 도입한 USN 응용 서비스를 위한 참조 모델을 제안하였다[2]. 그리고 이용자마다 서로 다른 조건과 관심 정보에 따라 구독예약을 통한 센서 정보를 획득할 수 있도록 WS-Eventing을 적용하여 실제적인 USN 응용 서비스의 가능성을 확장시켰다. 이를 기반으로 서로 상이한 플랫폼과 구현언어에 상관없이 USN 응용 서비스들 간에 표준화된 메시지를 전달하고, 관심 있는 센서 서비스에 대한 맞춤형 정보 전달이 가능하도록 하였다.

그러나, 센서 데이터는 사용 시 특정 이벤트에 관련된 여러 조건이 표현 가능해야 하며, 이를 위해서는 응용서비스 모델의 이벤트 매니저를 통한 이벤트의 등록 및 검색과 룰의 생성 툴이 필요하다. 또한, 수많은 센서 데이터 제공 서비스들을 이용하기 위해 서비스 제공자들이 표준화된 레지스트리에 자신의 서비스를 등록하고, 이용자는 서비스 레지스트리를 검색하여 이용할 수 있어야 한다. 그리고, 기존연구에서 웹 서비스 표준으로 사용한 WS-Eventing과 UDDI는 USN 응용 서비스를 위한 이벤트 및 레지스트리로서 불필요하거나 부족한 점이 있다.

이에 본 논문에서는 기존 연구에서 제안하였던 WS-ECA와 같은 웹 서비스 기반의 이벤트 룰을 참조한 USN 응용 서비스를 위한 이벤트 및 레지스트리의 구조와 API를 바탕으로 시스템을 설계 및 구현하였다.

II. 관련 연구

본 장은 본 연구의 기반 환경인 USN과 서비스 레지스트리와 관련된 UDDI, 그리고 서비스 이벤트와 관련된 WS-ECA에 대해 간략하게 소개한다.

2.1 USN

USN은 모든 사물에 전자 태그를 부착해 사물과 환경을 인식하고, 네트워크를 통해 실시간 정보를 구축하고 활용하기 위한 센서간의 네트워크이다. USN 기술은 전자태그, 리더, 미들웨어, 응용서비스 플랫폼 등을 중심으로 유무선망을 이용한 네트워크로 구성된다 [1][3].

2.2 UDDI

UDDI 레지스트리는 각종 정보들을 생성, 저장, 검색할 수 있는 XML 기반의 자료 저장 장치(소프트웨어+하드웨어)를 말한다. 클라이언트가 UDDI 레지스트리에 접근해서 정보를 저장하고 찾기 위해서는 SOAP(Simple Object Access Protocol) 메시지를 사용하게 되며, 이 SOAP 메시지는 HTTP(Hyperter Text Transfer Protocol) 프로토콜을 사용하기 때문에 클라이언트의 플랫폼과 구현 언어에 독립적이라는 특징을 갖게 된다. UDDI 레지스트리에 있는 서비스를 조회하기 위해서는 SOAP 메시지 형식을 취하고 있는 조회(Inquiry) API를 사용하여 검색하면 된다. UDDI는 일반적인 디렉토리 검색엔진과 닮은 서비스라 할 수 있으며, 웹 서비스 공급자를 검색하기 위해 White/Yellow/Green Page로 구성되었다. 이러한 데이터는 모두 XML 기반으로 보존되어 있다[4].

2.3 WS-ECA

WS-ECA는 기본적으로 ECA 룰과 발행/가입 메커니즘에 기반을 두고 있다.

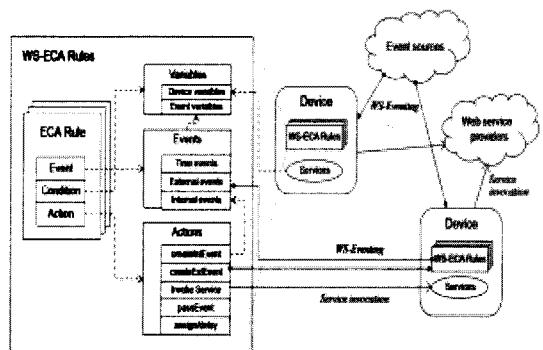


그림 1. WS-ECA 룰의 구성도
Fig 1. Diagram of WS-ECA Rule

ECA 를에 하나의 룰이 다른 룰의 실행을 위한 이벤트 발생을 가능하게 하는 이행 속성을 추가하여 디바이스의 서비스 조합을 지원한다. 또한, 조합 서비스의 실행을 위해 필요한 디바이스간의 호출은 발행/가입 메커니즘을 통해 구현된다.

ECA 를은 전통적인 데이터베이스에 능동적인 특성을 구현하기 위한 방법으로 고안되었다. ECA 를 표현 방법은 이벤트, 조건, 액션으로 나뉘고 이벤트는 시스템의 룰이 적용되는 시점을 결정하는 요소이다. 이 이벤트는 시스템 내부, 외부, 주기적 변화나 사건의 발생을 의미하며, 조건은 시스템의 상태를 참조하여 룰의 실행 여부를 결정하는 요소이고, 액션은 이벤트와 조건으로 표현된 특정 상황에 대응적으로 수행할 액션들을 정의하는 부분이다[5]. 그림 1은 WS-ECA 를 언어의 전체 구조를 나타낸 것이다.

III. 시스템 설계

본 시스템은 센서 서비스가 필요할 때 제공자는 센서 정보를 등록하고 요청자는 등록된 센서 정보를 검색하며, 그 정보를 가지고 제공자에게 접속 후 해당 서비스를 이용할 수 있도록 구현하였다. 또한, 제공자와 요청자 간의 서비스 요청 시 이벤팅 룰을 사용하여 원하는 이벤트 시 해당 정보를 제공 받을 수 있도록 설계하였다.

3.1 USR(Usn Service Registry)

기존 서비스 레지스트리는 비즈니스 위주의 데이터 구조와 API를 사용하여 센서 정보를 저장하고 검색하기에는 미흡한 부분이 있다. 이에 본 논문에서는 기존 서비스 레지스트리를 수정하여 센서 정보를 제공하기 위한 구조로 변형하였다.

3.1.1 데이터 구조

현재의 서비스 레지스트리는 제공자의 이름과 서비스 내용을 저장하는 데이터 구조를 가지고 있지만 센서 정보를 제공하는 데이터 구조는 없는 상태이다. 이에 센서 정보를 저장할 수 있는 새로운 엘리먼트를 추가하여 센서 정보를 저장 할 수 있도록 하였고, 제공자의 정보를 하위 엘리먼트에 포함하는 구조로 설계하였다. 그리고 기존의 비즈니스 서비스를 기술하는 부분을 선

서가 제공하는 정보를 저장할 수 있도록 수정하였다. 각각의 엘리먼트는 tModel을 참조하는 구조를 가지고 있으며, tModel에 센서 서비스를 분류할 수 있도록 확장하였고, 서비스를 제공할 수 있는 기술 정보를 tModel에 두고 이를 참조할 수 있도록 하는 구조로 설계하였다. 그림 2는 센서 정보를 서비스하기 위해 설계한 데이터 구조이다.

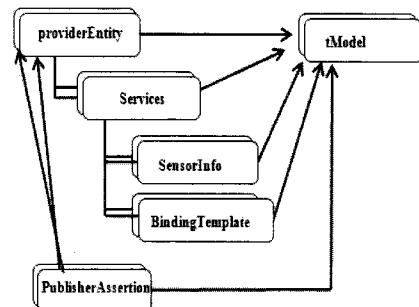


그림 2. USR 데이터 구조
Fig 2. Structure of USR Data

3.1.2 USR API

센서 정보를 제공받기 위해서는 제공자가 정보를 저장하고 저장한 정보를 사용자가 검색해서 필요한 정보를 얻어야 한다. 여기에 등록 및 검색에 필요한 API가 필요하며, 이를 위해 기존의 비즈니스 정보를 등록, 검색하는 기능을 수행하는 API를 센서 정보에 맞게 생성 및 재정의 할 필요가 있다. 표 1은 센서 정보를 등록 및 검색하기 위한 API들이다.

표 1. USR API
Table. 1 USR API

등록 (Publishing) API	저장 (save_xx)	providerEntity,rService, sensorInfo, binding, tModel
	삭제 (delete_xx)	providerEntity, SensorService, sensorInfo, binding, tModel
	기타 (get_xx)	registryInfo, assertionStatusReport, publisherAssertions
검색 (inquiry) API	검색 (find_xx)	providerEntity, SensorService, sensorInfo, binding, tModel
	상세열람 (get_xx)	providerDetail.sensorDetail, operationInfo, ServiceDetail, bindingDetail, tModelDetail

등록 API에서 저장 부분은 `save`로 시작하며, 센서 기본적인 정보 즉 센서ID, 측정범위, 위치 등과 같은 정보를 저장하는 기능을 수행하는 `SensorInfo`, 제공자가 제공하는 서비스에 대한 표현을 담당하는 `Service`, 제공자의 정보를 저장하는 `providerEntity`, 서비스를 이용하기 위한 필요한 기술 정보를 저장하는 `Binding`, 실제적인 기술 정보를 저장하는 `tModel` 등이 있다. 삭제 API는 저장된 정보를 삭제시키는 API로써 `delete`로 시작한다. 검색 API는 `find`로 시작하며 저장된 정보를 검색하기 위한 기능을 수행한다. 상세 열람 검색은 `get`로 시작하며 자세한 정보 검색을 원할 때 사용한다.

3.1.3 USR 시스템 구조

USR은 기존의 UDDI의 데이터 구조를 변형하여 센서 정보를 표현할 수 있도록 하였다. USR의 동작 방식은 기존의 UDDI와 비슷하다.

그림 3은 USN 응용 서비스 레지스트리의 시스템 구조이다.

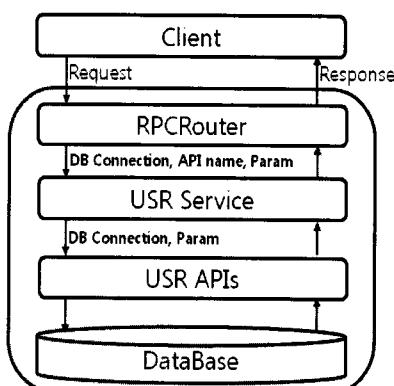


그림 3. USR 시스템 구조
Fig 3. Structure of USR System

RPCRouter는 Client의 요청 메시지를 처리하고, SOAP 메시지를 파싱해 Body 블록 내에 있는 주 메시지를 분리하여 파라미터 정보를 저장한다. 또한 API 이름을 추출한다.

USR Service는 RPCRouter에서 전달받은 DB connection, API name, 파라미터 정보에서 API name으로 해당 API를 호출하며, 이때 DB connection과 파라미터

정보를 호출되는 API에 인자로 넘겨준다.

3.2 서비스 이벤트

실제 이벤트들을 처리하기 위해서는 하나의 이벤트를 수신할 때도 다양한 조건을 제시할 수 있도록 하여, 동일한 이벤트를 수신 시 다양한 조건식에 따라 원하는 작업을 수행할 수 있도록 ECA 룰을 처리, 검증, 저장한다.

3.2.1 서비스 이벤트 기술 언어

서비스 이벤트 기술 언어는 요청자가 제공자에게 서비스 요청 시 보내게 되는 서비스 기술 문서에 import하여 사용할 수 있도록 XML 포맷으로 설계된 언어이다. 요청자가 제공자에게 다양한 조건 및 이벤트를 요청할 수 있도록 하여, 효율적인 서비스 정보를 제공받을 수 있도록 한다. 그림 4는 서비스 이벤트 기술 언어의 전체 스키마를 나타낸다.

USN 서비스 이벤트 기술 언어는 크게 네 부분으로 나뉜다. 첫째 `variables`는 이벤트 변수를 표현하는 엘리먼트이며, 둘째 `events`는 이벤트 내용을 표현하는 엘리먼트이다. 셋째 `actions`은 이벤트 동작 방식을 표현하며, 넷째 `rules` 엘리먼트는 이벤트에 대한 룰을 표현하는 엘리먼트이다.

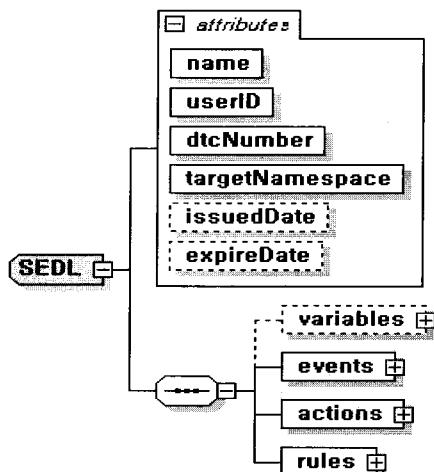


그림 4. 서비스 이벤트 기술언어 스키마
Fig 4. Schema of Service Event Description Language

3.2.2 시나리오

서비스 이벤트의 시나리오는 그림 5와 같다.

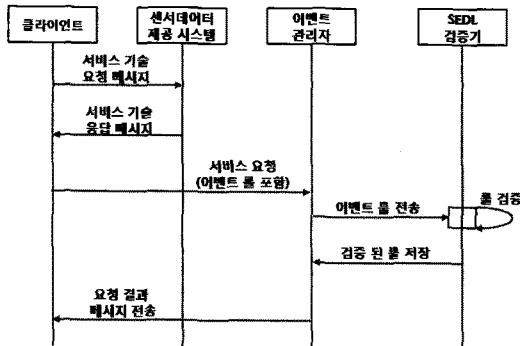


그림 5. 시스템 시나리오 시퀀스 다이어그램
Fig 5. Sequence diagram of system scenario

3.2.3 설계

서비스 이벤트의 세부 모듈은 그림 6과 같다.

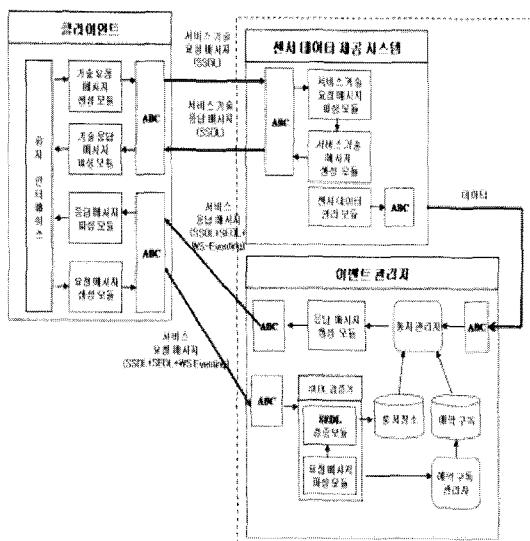


그림 6. 서비스 이벤트 프로토 타입 세부 모듈
Fig 6. Service Event Prototype module

클라이언트는 센서 데이터 제공 시스템에게 접속하여 서비스 기술 요청 메시지를 요청한다. 그리고 전송 받은 서비스 기술 응답 메시지를 파싱하여 서비스 제공자가 제공하고 있는 서비스의 정보를 취득한다. 이 정보를

기반으로 사용자가 원하는 센서들의 요청 메시지를 구성하여 결과적으로 필요한 센서 데이터를 사용자에게 제공하게 된다.

센서 데이터 제공 시스템은 실질적인 센서를 관리하고 정보를 서비스해주는 시스템이다. 이벤트 관리자에게 상황마다 생기는 센서 데이터를 전송한다.

이벤트 관리자는 WS-Eventing에 필요한 여러 메시지를 처리한다. 그리고 요청되는 예약 구독을 관리한다. 그리고 필터링을 통해 요구하는 센서 데이터를 선별하여 서비스 요청자에게 전달하는 역할을 한다. 크게 서비스 이벤트 룰 검증, 서비스 이벤트 룰 저장소, 통지 관리자, 예약 구독 저장소, 예약 관리자로 나누어 설계하였다.

예약 관리자는 여러 클라이언트의 예약 구독 메시지 (Subscribe, UnSubscribe, Renew, GetStatus)를 요청 받아 예약 구독 저장소에 저장하여 관리한다.

예약 구독 저장소는 예약 구독에 관련된 사항을 링크드리스트 형태의 메모리상에 저장하도록 설계하였다.

SEDL 검증기는 해당 서비스 요청 메시지를 받으면 서비스 기술언어와 서비스 이벤트 기술언어를 나누어 서비스 기술언어 부분은 예약 구독 관리자에게 전송하고, 서비스 이벤트 기술언어는 서비스 이벤트 룰 검증 모듈을 통해 해당 이벤트 룰의 적합 여부를 판단하여 룰 저장소에 저장하는 역할을 한다.

통지 관리자는 발생된 이벤트를 각각의 저장소에 저장되어 있는 정보를 이용하여 해당하는 각각의 클라이언트에게 전송해주는 전체적인 관리 역할을 하도록 설계하였다.

IV. 시스템 구현

본 시스템은 유비쿼터스 응용 서비스 시스템과 네트워크 노드 사이에 위치하는 미들웨어(Middleware) 구조를 갖는다. 또한 웹 서비스의 기본 구조를 가지며 센서 정보의 등록과 검색 요청 및 제공 서비스를 서플레이션하기 위해 시스템을 구현하였다.

4.1 전체 시나리오

시나리오는 다음과 같은 순서를 따른다.

1) 센서 서비스 제공자는 USR에 자신의 정보 및 서비

스 정보를 등록

- 2) 센서 서비스 요청자는 USR에 접속하여 원하는 센서 데이터를 제공하는 제공자에 대해 검색하고 해당 센서 데이터 정보를 제공해 주는 제공자에 대한 접속 정보를 받음
- 3) 제공자에게 접속이 성공하면 서비스 기술 데이터를 제공받아 서비스를 제공받기 위한 조건이 맞는지 확인한 후 조건이 맞으면 서비스 이용을 요청
- 4) 서비스가 제공하는 센서 정보를 확인하고, 요청 메시지와 이벤트 룰을 작성한 후 서비스 제공자에게 전송
- 5) 요청 메시지와 이벤트 룰을 받은 서비스 제공자의 센서 데이터 처리 시스템은 요청된 내용과 해당 이벤트 룰에 따라 측정 데이터를 생성하고 응답 메시지를 구성
- 6) 구성된 메시지를 서비스 요청자에게 전송

4.2 USR

USR 개발 환경은 JAVA 1.5 JDK, APACHE Tomcat 5.5, MS-SQL을 이용하여 구현하였다. 그림 7은 USR에 등록된 내용을 검색 후 그에 따른 XML 문서를 받아 볼 수 있도록 하였으며 그 결과를 나타낸 것이다.

```
<?xml version='1.0' encoding='UTF-8'?>
<providerDetail xmlns="urn:uddi-org:api" generic="2.0" operator="mie.pcu.ac.kr" truncated="false">
  <ProviderEntity authorizedName="C087C5B0-8FCC-11DD-AC37-E779EAD7778D" providerKey="CBA43B00-8F8A-11DD-AC37-F2E2C50E5D50" operator="mie.pcu.ac.kr">
    <name>Paichai Uml</name>
    <description>test weather</description>
    <discoverURLs>
      <discoverURL
        useType="providerEntity">http://203.250.143.171:8080/openwebbusr/getprovider.jsp?providerKey=CBA43B00-8F8A-11DD-AC37-F2E2C50E5D50</discoverURL>
    <discoverURL
        useType="ProviderEntity">http://203.250.143.171:8080/openwebbusr/getprovider.jsp?providerKey=CBA43B00-8F8A-11DD-AC37-F2E2C50E5D50</discoverURL>
    <discoverURL
        useType="providerEntity">http://203.250.143.171:8080/openwebbusr/getprovider.jsp?providerKey=CBA43B00-8F8A-11DD-AC37-F2E2C50E5D50</discoverURL>
    </discoverURLs>
  <Services />
  <contacts>
    <contact>
      <personName>yeon sung kyun</personName>
      <phone useType="cell">01064192972</phone>
      <email useType="private">sungkun83@nate.com</email>
    </contact>
  </contacts>
</ProviderEntity>
```

그림 7. USR 출력 결과
Fig 7. Output of USR

4.3 서비스 이벤트

서비스 이벤트는 .Net Framework 런타임 환경에서 C#을 이용한 ASP.NET 컨테이너를 통하여 웹서비스를 구현하였다.

```
<ssdi:Message>
  <ssdi:ServiceCode>0000-0000-0000-0000</ssdi:ServiceCode>
  <ssdi:RuleName>Rainy - alarm - rule</ssdi:RuleName>
  <ssdi:Phenomenon ResponsePhenomenonID="Weather-0001">
    <ssdi:SensorCharacteristics>
      <ssdi:SensorType>Rainy Detect</ssdi:SensorType>
      <ssdi:SensorID>0000-0000-0000-0001</ssdi:SensorID>
      <ssdi:Unit>mm</ssdi:Unit>
      <ssdi:Scale>1</ssdi:Scale>
      <ssdi:Resolution>0.1</ssdi:Resolution>
      <ssdi:OffsetCorrection>0.1</ssdi:OffsetCorrection>
      <ssdi:MinValue>0.0</ssdi:MinValue>
      <ssdi:MaxValue>1000</ssdi:MaxValue>
    </ssdi:SensorCharacteristics>
    <ssdi:Value Count="3" Time="18:00:00" Date="2008-10-10">25</ssdi:Value>
  <ssdi:ErrorCode>ErrorCode의 String 값</ssdi:ErrorCode>
  </ssdi:Sensor>
  <ssdi:RightNotice>
    <ssdi:ExpirationDate>2008-11-20</ssdi:ExpirationDate>
    <ssdi:LimitationCount>5</ssdi:LimitationCount>
    <ssdi:RuleName>Temperature Rule</ssdi:RuleName>
    <ssdi:Phenomenon ResponsePhenomenonID="Weather-0002">
      <ssdi:Sensor SensorID="0000-0000-0000-0002">
        <ssdi:SensorCharacteristics>
          <ssdi:SensorType>Temperature</ssdi:SensorType>
          <ssdi:SensorID>0000-0000-0002</ssdi:SensorID>
          <ssdi:Unit>deg</ssdi:Unit>
          <ssdi:Scale>kilo</ssdi:Scale>
          <ssdi:Resolution>1</ssdi:Resolution>
          <ssdi:OffsetCorrection>1</ssdi:OffsetCorrection>
          <ssdi:MinValue>-1000</ssdi:MinValue>
          <ssdi:MaxValue>1000</ssdi:MaxValue>
        </ssdi:SensorCharacteristics>
        <ssdi:Value Count="2" Time="10:00:00" Date="2008-11-20">300</ssdi:Value>
      </ssdi:Sensor>
    </ssdi:Phenomenon>
    <ssdi:RightNotice>2008-11-20</ssdi:ExpirationDate>
    <ssdi:LimitationCount>30</ssdi:LimitationCount>
    <ssdi:RuleName>Temperature Rule</ssdi:RuleName>
  </ssdi:RightNotice>
</ssdi:Message>
</ssdi:SEDL>
```

그림 8. 서비스 이벤트 결과
Fig 8. Output of Service Event

그림 8은 서비스 요청자가 제공자에게 서비스에 대한 요청 메시지를 보내는 부분으로 서비스 이벤트 기술 언어를 import시켜 XML 문서 형식으로 출력한 것이다.

V. 고찰 및 결론

USN의 센서 데이터는 사용 시 특정 이벤트에 관련된 여러 조건이 표현 가능해야 하며, 이를 위해서는 응용 서비스 모델의 이벤트 매니저를 통한 이벤트의 등록 및 검색과 룰의 생성 룰이 필요하다. 또한 수많은 센서 데이터 제공 서비스들을 이용하기 위해 서비스 제공자들이 표준화된 레지스트리에 자신의 서비스를 등록하고 이용자는 서비스 레지스트리를 검색하여 이용할 수 있어야 한다. 현재 이를 위한 웹서비스의 표준으로 WS-Eventing과 UDDI가 있지만, USN 응용 서비스를 위한 이벤트 및 레지스트리로서 다소 미흡하다.

이에, 본 논문은 센서 정보를 저장할 수 있는 데이터 구조를 구성하였고, 그에 따른 API의 재정의 및 추가 생성과 이에 대한 시스템을 설계 및 구현하였다. 이는 제공자가 서비스 정보만 제공하는 기존 시스템의 단점에서 센서에 대한 정보를 등록하는 부분을 추가 하여, 서비스에 해당하는 센서 정보를 검색할 수 있게 함으로써, 센서 정보를 제공하는 제공자에 대한 보다 정확한 검색과 서비스 정보를 제공받을 수 있게 되었다.

또한, 기존의 이벤팅 구조에 WS-ECA 를 적용한 표준화된 서비스 기술 언어를 설계하여, 유연한 이벤팅 처리가 가능하도록 하였고, 이를 사용하여 사용자가 다양한 조건을 제시하여 원하는 정보를 제공받을 수 있도록 설계 및 구현하였다. 이는 기존의 WS-Event에서의 이벤트 조건의 제한을 탈피하여 제공자가 제공하는 서비스 정보를 요청할 때, 각 이벤트에 대한 조건 및 액션을 정의하여 보다 효율적인 서비스 등록을 할 수 있으며, 이에 따라 요청자가 원하는 서비스 정보를 제공받을 수 있게 되었다.

향후 연구 과제로는 센서 데이터 정보를 더 효율적으로 제공하기 위해서서 USR의 개선 및 이벤트 를을 효과적으로 표현할 수 있도록 스키마 및 내부 속성들을 개선해야 할 것이다.

참고문헌

- [1] Weiser. M, "The Computer for the 21st Century", Scientific American seminal article, 1991
- [2] 이상조, "USN 응용서비스를 위한 참조 모델", 한국해양정보통신학회, 2007
- [3] 유승화, "무선인식, RFID란?", eBizKorea, 통권82호, 2006
- [4] UDDI.org "UDDI Technical White Paper", http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf, Sept.2000.
- [5] 이원석, "The Protocol on WS-ECA FrameWork" 한국인터넷 정보학회 (8권 6호), 2007

저자소개



염성근(Sung-Kun Yeom)

2008년 배재대학교 컴퓨터공학과

(공학사)

2008년 배재대학교 컴퓨터공학과

(석사과정)

※ 관심분야: XML, USN, Web Service



김용운(Yong-Woon Kim)

1989년 동아대학교 전자공학과

(공학사)

1994년 포항공과대학교

정보통신공학과(공학석사)

1990년~1991년 : 삼성항공

1995년~2001년 : 한국전자통신연구원

2001년~2002년 : ZTE 퓨처텔

2002년~2004년:이니텍

2005년 ~ : 한국전자통신연구원

※ 관심분야: 센서 네트워크, RFID 시스템, 컴퓨터네트워크



유상근(Sang-Keun Yoo)

1997년 충남대학교 컴퓨터공학과

(공학사)

1999년 충남대학교 컴퓨터공학과

(공학석사)

1999년~2000년 : 시그마텍

2001년~2001년 : 한국전자통신연구원

※ 관심분야: 센서 네트워크, RFID 시스템, 정보보호시스템, 컴퓨터 네트워크



김형준(Hyoung-Jun Kim)

1986년 광운대학교 컴퓨터공학과

(공학사)

1988년 광운대학교 컴퓨터공학과

(공학석사)

2007년 충남대학교 컴퓨터과학과(이학박사)

1988년~ 현재 : 한국전자통신연구원

※ 관심분야: 센서 네트워크, 모바일 RFID 서비스, 컴퓨터 네트워크, 차세대인터넷



정회경(Hoe-Kyung Jung)

1985년 광운대학교 컴퓨터공학과

(공학사)

1987년 광운대학교 컴퓨터공학과

(공학석사)

1993년 광운대학교 컴퓨터공학과(공학박사)

1994년~현재 배재대학교 컴퓨터공학과 교수

※ 관심분야 : 멀티미디어 문서정보처리, XML, Web Services, SVG, Semantic Web, MPEG-21, Ubiquitous Computing, USN