DOI: 10.3745/KIPSTD.2008.15-D.2.247

유비쿼터스 환경에서 센서 데이터와 서비스의 연계를 표현하는 마크업 언어

이 훈 순[†] · 진 성 일[††]

요 약

유비쿼터스 시대에는 우리 주변에 산재하는 수많은 스마트 객체들이 끊임없이 방대한 양의 센서 데이터를 생성해 낸다. 인터넷이 널리 보급됨에 따라 인터넷으로 연결된 정보의 바다로부터 유용한 정보를 찾는데 도움을 주는 인터넷 정보 검색 엔진이 등장한 것처럼, 스마트 객체가 생성한 센서 데이터로부터 우리 삶을 윤택하게 하는데 활용될 정보를 추출하여 응용 서비스에 전달하는 일을 함으로써 유비쿼터스 서비스를 개발하는데 도움을 주는 센서 데이터 스트림 처리 미들웨어들이 등장하고 있다. 이러한 센서 데이터 스트림 처리 미들웨어를 이용하여 유비쿼터스 서비스를 사람들에게 제공하기 위해서는 제공될 서비스와 관련된 정보를 미들웨어에게 알려 주어야 한다. 이에 본 논문에서는 센서 데이터를 이용하는 유비쿼터스 서비스를 나타내는데 필요한 정보를 구분하고, 이를 효과적으로 표현하는 컨텍스트 드리븐 서비스 마크업 언어를 제안한다. 제안하는 컨텍스트 드리븐 서비스 마크업 언어를 이용하면 다양한 상황에서 제공되어야 하는 다양한 유비쿼터스 서비스를 쉽게 표현할 수 있다.

키워드 : 유비쿼터스, 유비쿼터스 서비스, 컨텍스트, 컨텍스트-드리븐 서비스, 센서 데이터, 미들웨어

A Markup Language for Describing the Linkage between Sensor Data and Service in the Ubiquitous Environment

Lee Hun Soon[†] · Jin Seung Il[††]

ABSTRACT

In the ubiquitous environment, it is scattered all over our neighboring in many smart objects. These smart objects constantly produce the information and the amount of the generated information is massive. As the internet search engine came out to help us to find the useful data from the sea of the information connected to the internet, the sensor data stream processing middleware is appearing to make us to develop the ubiquitous service easily by extracting the meaningful information from the massive sensor data and delivering the extracted information to the application which makes our life convenient.

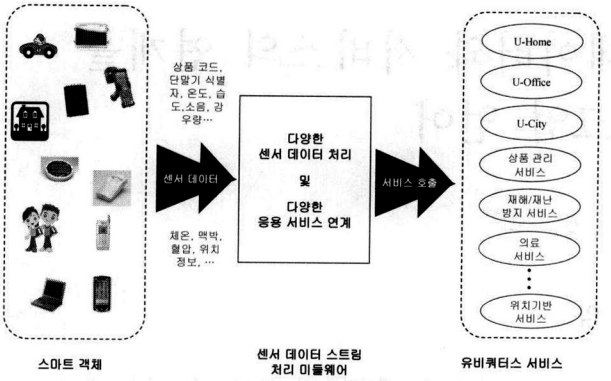
We have to inform the information relating to the provided service to a middleware so that the ubiquitous service can be provided by using sensor data stream processing middleware. In this paper, we classify the information which is needed to express the ubiquitous service which uses sensor data for the service providing. And we propose a distinct markup language called Context-driven Service Markup Language (CSML) to effectively describe this information. We can easily express the various ubiquitous services which have to be provided in the various situations using proposed CSML.

Key Words : Ubiquitous, Ubiquitous Service, Context, Context-driven Service, Sensor Data, Middleware

1. 서 론

반도체, 컴퓨팅, 통신, 센서, 센서 네트워크 기술의 발달로 인해 마크와이저가 유비쿼터스 컴퓨팅 환경이 되기 위해 전제 조건으로 두었던 것[1]들이 현실화되고 있다. 유비쿼터스 환경에서 우리 주변에는 셀 수 없을 정도로 많은 스마트 객체(센서 혹은 컴퓨팅 디바이스를 포함하고 있는 물체)들이 산재하며, 이들 스마트 객체들은 끊임없이 정보를 생성해 낸다. 이러한 유비쿼터스 환경의 스마트 객체들이 생성해 내는 데이터(이하 센서 데이터)를 이용하여 우리 인간의 삶을 윤택하게 하는 다양한 유비쿼터스 서비스가 출현하고 있다. 이러한 유비쿼터스 서비스들은 센서를 통해 수집된 정보를 이용하여 서비스가 제공되어야 하는 컨텍스트인지를 파악한 후 자동으로 서비스를 제공한다고 하여 컨텍스트-드

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 IT 신성장동력 핵심기술개발사업의 일환으로 수행하였음. [2005-S405-02, 차세대 인터넷 서버기술]
† 정 회 원 : 한국전자통신연구원 선임연구원
†† 종신회원 : 충남대학교 전기정보통신공학부 교수
논문접수 : 2007년 5월 8일, 심사완료 : 2007년 12월 7일



(그림 1) 센서 데이터 스트림 처리 미들웨어 개념도

리븐 서비스[2]라고도 한다.

인류는 인터넷이 등장하여 널리 보급되면서 정보의 홍수로 인해 충격을 받았었다. 하지만, 센서 데이터들의 방대함은 인터넷 데이터와는 비교할 수 없을 정도이다. 이러한 센서 데이터로부터 우리 생활을 윤택하게 하는데 이용할 수 있는 의미있는 데이터를 찾아 내는 것은 매우 복잡하고 지루한 일이다. 또한, 정보의 양이 이전과 비교할 수 없을 정도로 많아짐에 따라 불필요하고 잘못된 정보들이 사용될 가능성이 더 높아지고 있다. 이러한 이유로 인터넷 시대에 인터넷 정보 검색 엔진이 등장했듯이, 유비쿼터스 시대에는 (그림 1)에서 보이는 것처럼 스마트 객체로부터 끊임없이 생성되어 스트림 형태로 들어오는 센서 데이터로부터 의미있는 정보를 뽑아서 응용에서 이용할 수 있게 해주는 센서 데이터 스트림 처리 미들웨어들이 등장하고 있다. 이들 센서 데이터 스트림 처리 미들웨어들은 유비쿼터스 환경에서 서비스 개발자들이 공통으로 해야 하는 일들을 담당한다. 즉, 센서를 관리하며, 센서 데이터로부터 의미있는 데이터를 추출하여 응용 서비스에 전달하는 역할을 한다. 서비스 개발자가 이러한 센서 데이터 스트림 처리 미들웨어를 이용하는 유비쿼터스 서비스 즉, 컨텍스트-드리븐 서비스를 개발하기 위해서는 미들웨어에게 어떠한 상황에서, 어떠한 정보를, 어떻게 가공하여, 어떤 서비스에 전달할 것인가를 알려주는 방법이 필요하다.

RFID응용 개발자와 RFID 미들웨어 개발자를 위해 RFID 응용에 특화된 EPC(Electronic Product Code) 데이터에 대해 관심있는 데이터를 추출하여 응용 서비스에 전달하는 것을 표현할 수 있는 ALE(Application Level Events) 규격[3]이 EPCglobal[4]에 의해 제안되었다. 또한, 상용 센서 데이터 스트림 처리 미들웨어[5][6]에서는 API 혹은 GUI 도구를 이용하여 센서 데이터와 서비스의 연관 관계를 표현하고 있다. 하지만 기존의 연구들은 특정 유형의 서비스나 입력 데이터 유형에 특화되어 있으며, 유비쿼터스 환경의 다양한 센서들이 생성해 내는 비정형의 데이터, 다양한 컨텍스트, 다양한 서비스, 컨텍스트와 서비스의 다양한 연계, 서비스의 생명 주기 등에 대한 고려가 부족하다.

이에 본 논문에서는 센서 데이터를 이용하는 유비쿼터스 서비스를 나타내는데 필요한 정보를 분류하고, 이를 효과적으로 표현할 수 있는 컨텍스트-드리븐 서비스 마크업 언어(Context-driven Service Markup Language, 이하 CSML)를 제안한다. 제안하는 CSML을 이용하면, XML 스키마[7]를 이용하여 그 형식을 정의할 수 있는 다양한 비정형의 형식을 가지는 센서 데이터에 대해 서비스가 제공되어야 하는 컨텍스트 조건과 서비스에 이용할 데이터를 쉽게 표현할 수 있으며, 컨텍스트 조건이 만족되었을 때 수행되어야 하는 다양한 유비쿼터스 서비스와 연계 정보를 쉽게 표현할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고, 3장에서는 유비쿼터스 서비스를 표현하는데 필요한 정보를 분류하고, 4장에서는 컨텍스트-드리븐 서비스 마크업 언어를 제안한다. 그리고, 5장에서는 제안하는 CSML과 관련 연구간의 정성적인 비교 평가를 하고, 6장에서는 제안하는 CSML을 수용하도록 구현한 사례 시스템에 대해 살펴본 후, 7장에서 결론을 맺는다.

2. 관련 연구

유비쿼터스 컴퓨팅의 핵심 기술로 주목 받고 있는 RFID 기술의 발달로 인해 이를 이용한 유비쿼터스 서비스들이 먼저 등장하고 있다. 이에 EPCglobal에서 주도하는 컨소시움에서 RFID를 이용한 서비스 개발자와 RFID 미들웨어 개발자를 위해 RFID 데이터에 대해 필터링하고 수집하여 서비스에 전달하는 것을 지원하는 시스템의 인터페이스에 대한 산업 표준인 ALE규격[3]을 제안했다. ALE 규격은 이벤트 사이클 스펙(event cycle specification)과 define, subscribe, unsubscribe, poll, immediate등의 11개의 API들로 구성되어 있다. 이벤트 사이클 스펙은 어떤 주기로 필터링 조건이 만족되는지 평가하여 서비스를 호출하는 지를 나타내는 이벤트 사이클 바운더리 스펙(event cycle boundary specification)과, 필터링 조건과 그 필터링 조건을 만족하는 데이터를 어떻게 가공하여 전달하는지를 표현하는 이벤트 사이클 리포트 스펙(event cycle report specification)으로 구성된다. 필터링 조건은 EPC로 인코딩된 데이터에 대해 결과에 포함되어야 하는 패턴 혹은 불포함되어야 하는 패턴을 정의함으로써 표현된다. 이벤트 사이클 스펙을 만족하는 결과는 알림을 받아야 하는 URI로 등록된 곳으로 주기적으로 전달된다. 알림을 받아야 하는 URI의 등록은 subscribe API를 이용하는데 이에 입력되는 인자는 HTTP, TCP, 혹은 FILE 프로토콜을 따르는 서비스에 대한 URI 이다. 하지만 ALE 규격은 대상이 되는 데이터가 EPC로 표현된 것으로만 한정되어 있을 뿐 아니라, 조건을 만족하는 데이터를 가공하는 방법으로 카운팅하고 그룹핑하는 정도의 기능 밖에 없다는 단점이 있다. 또한, 이벤트 사이클 스펙에 서비스가 호출되는 주기를 시간에 기반하여 명시할 수 있지만 서비스의 생명 주기를 직접 나타낼 수 없고 API(subscribe, unsubscribe)를 이용하여야 한다.

또한, 센서 데이터를 이용하여 서비스를 개발하는데 도움

을 주는 미들웨어들[5][6]이 시장에 진입한 상태이다. SUN 사의 Sun Java System RFID Software[5]나 ORACLE 사의 Oracle Sensor Edge Server[6]는 컨텍스트 조건을 표현하고, 컨텍스트 조건을 만족하는 데이터를 받아들여 서비스에 연결하기 위한 인터페이스를 제공한다. 미리 정의된 단순 필터를 이용하여 센싱된 데이터로부터 원하는 데이터를 추출할 수 있으며, 복잡한 로직을 가지는 필터를 사용할 수 있도록 하기 위해 사용자 정의 필터를 개발하는 도구를 제공한다. 또한, 걸러진 데이터를 JMS, 웹 서비스, HTTP, FILE, TCP 프로토콜을 이용하여 응용 서비스에 전달하는 것을 나타낼 수 있다. Sun Java System RFID Software는 입력 데이터로 RFID 이벤트만을 고려했지만, Oracle Sensor Edge Server는 임의의 센서 데이터 스트림이 입력 데이터가 될 수 있음을 고려했다는 점이 다르다. 두 시스템 모두 단순히 컨텍스트와 서비스의 연결만을 표현할 수 있을 뿐 어떤 주기로 서비스를 호출하는 지와 서비스의 생명 주기 관련 정보를 표현하는 방법을 제공하지 않는다. 또한, 컨텍스트 조건을 만족하는 데이터 자체를 전달하는 것만 표현할 수 있고, 이들 중 일부를 추출한다거나 다수를 하나로 병합하는 것과 같이 결과를 가공하여 전달하는 것은 표현하기가 어렵다.

뿐만 아니라 컨텍스트 모델링과 관련하여 CML(context modeling language) [8]과 CSCP (Comprehensive Structured Context Profile) [9]와 같은 연구가 있었다. 컨텍스트-어웨어 서비스가 제공되어야 하는 상황을 묘사하기 위해 제안된 CML은 초기 컨텍스트 모델의 단점인 응용 로직과 밀접하게 연관있으며 고정적이라는 점을 극복하고, 응용간 컨텍스트 정보와 컨텍스트 센싱 인프라스트럭처를 공유하도록 하기 위해 비즈니스 도메인에서 모델링을 위해 주로 사용되었던 ORM(Object-Role Modeling)[10]을 확장했다. CML에서는 서로 다른 소스로부터 오는 정보를 구별할 수 있도록 했고, 정보의 불완전성과 시간 속성을 가지는 데이터에 대한 표현을 할 수 있게 했으며, 정보에 대한 소유 등을 나타낼 수 있도록 했다. 서비스 제공시 컨텍스트 정보가 반영되도록 하기 위해 고안된 CSCP는 RDF(Resource Description Framework)[11]에 근거를 둔 컨텍스트 정보를 표현하기 위한 언어이다. CSCP는 서비스가 제공되어야 하는 디바이스 정보, 사용자의 선호도나 관심 사항 같은 사용자 특정 정보, 그리고 네트워크 연결 정보와 같이 서비스가 제공되는 환경을 표현할 수 있다. CML과 CSCP는 다양한 상황에서 제공되어야 하는 다양한 유비쿼터스 서비스를 표현하는데 있어서 서비스가 제공되어야 하는 컨텍스트(상황) 정보는 잘 표현할 수 있으나 서비스에 전달할 내용이나 서비스, 그리고 서비스 제공의 생명 주기와 같은 실제 서비스와 관련된 내용을 표현하기 위한 것이 포함되어 있지 않다.

3. 유비쿼터스 서비스 표현 정보

유비쿼터스 서비스는 제공받는 위치, 제공 받는 시간, 제공받는 자의 행위, 주변 상황 등에 특화되어 서비스가 제공되어야 한다. 유비쿼터스 환경에서 이러한 특화된 정보들은 스마트 객체를 통해 생성될 수 있다. 이러한 점을 고려하여 다양한 스마트 객체가 끊임없이 생성하는 센서 데이터를 이용하는 유비쿼터스 서비스를 표현하는데 필요한 정보를 분류하면 다음과 같다.

(1) 서비스가 제공되어야 하는 다양한 컨텍스트 정보

다수의 다양한 데이터 소스로부터 들어오는 비정형의 센서 데이터 스트림에 대해 서비스가 호출되어야 하는 상황(context)을 판단할 수 있는 정보가 포함되어야 한다. 이때 데이터 소스로 실시간으로 센싱되어 들어오는 센서 데이터뿐 아니라 정보 저장소에 있는 데이터도 상황 판단의 대상이 될 수 있어야 한다.

(2) 컨텍스트 조건이 만족될 때 호출될 다양한 서비스

컨텍스트 조건이 만족되었을 때 호출되어야 하는 서비스 정보가 포함되어야 한다. 호출되는 서비스 표현 정보는 현재 많이 사용되는 다양한 서비스들뿐 아니라 앞으로 새로운 유형의 서비스가 출현했을 때 이를 수용할 수 있어야 한다.

(3) 서비스에 이용할 데이터

센서 데이터로부터 서비스에 이용할 데이터를 추출한 후 가공하여 서비스에 전달하는 것을 포함하여야 한다. 서비스에 전달할 데이터로는 센서 데이터로부터 추출하는 동적 데이터뿐 아니라 미리 그 값이 정해진 정적 데이터도 포함된다.

(4) 서비스가 수행되는 주기 정보

서비스 호출을 해당 컨텍스트 조건이 만족되었을 때마다 호출하는지 혹은 일정 주기(예, 5번 만족했을 때 혹은 매 1시간마다) 로 호출하는 지와 같은 서비스가 수행되는 주기 정보를 포함해야 한다.

(5) 서비스가 의미있는 시간뿐 아니라 종료 시간 정보

서비스를 제공하는데 있어서 하루 중 서비스가 의미있는 시간대 정보와 서비스를 더 이상 제공하지 않고 철회하는 시간 정보를 포함해야 한다.

(6) 서비스를 인증을 위한 사용자 정보

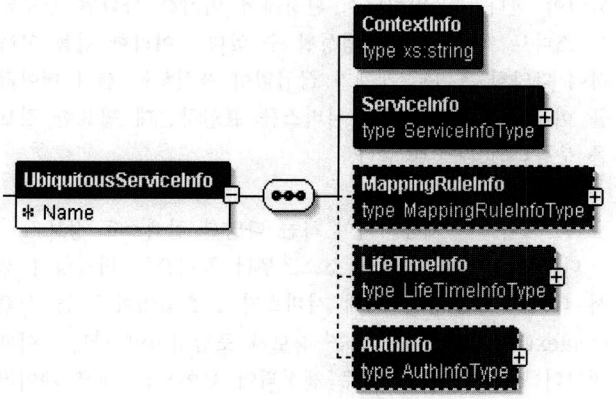
서비스를 이용하는데 있어서 인증이 필요한 경우라면 서비스 인증을 위한 사용자 정보도 포함해야 한다.

4. 컨텍스트-드리븐 서비스 마크업 언어

본 장에서는 3장에서의 살펴본 유비쿼터스 서비스를 나타내는데 필요한 정보를 효과적으로 표현할 수 있는 컨텍스트-드리븐 서비스 마크업 언어인 CSML을 제안하고, 제안하는 CSML을 이용하여 센서 데이터와 서비스의 연계를 표현한 예를 든다.

4.1 CSML 구조

제안하는 CSML은 XML[12] 형식을 따르는데 (그림 2)와 같이 서비스가 호출되어야 하는 상황을 인지하기 위한 컨텍스트 정보 <ContextInfo> (3장의 (1), (3)에 해당), 컨텍스트



(그림 2) CSML 구조

조건이 만족되었을 때 수행되어야 하는 서비스 정보 <ServiceInfo> (3장의 (2)에 해당), 컨텍스트로부터 추출한 결과를 서비스의 인자로 매핑시키는 규칙을 나타내는 매핑 규칙 정보 <MappingRuleInfo>(3장의 (3), (4)에 해당), 컨텍스트-드리븐 서비스가 의미있는 시간을 나타내는 생명 주기 정보 <LifeTimeInfo> (3장의 (5)에 해당), 그리고 서비스 사용을 위한 인증 정보 <AuthInfo> (3장의 (6)에 해당)로 구성된다.

4.1.1 컨텍스트 정보

현재 인터넷 상의 데이터 교환의 표준으로 XML이 자리를 잡고 있으므로, 유비쿼터스 환경에서 대부분의 센서 데이터들도 XML 형태로 전송될 것으로 예상된다. 따라서, 유비쿼터스 서비스를 제공하기 위해서는 XML로 표현된 센서 데이터로부터 서비스가 호출되어야 하는 상황인지를 판단하여야 하며, 서비스가 호출되어야 하는 상황에서 서비스 제공을 위해 필요한 정보를 추출하여 가공한 후 서비스에 전달해야 한다.

이에 본 논문에서 제안하는 CSML에서는 컨텍스트 정보 <ContextInfo>를 이용하여 센서로부터 생성되어 전달되는 XML 형태의 비정형 센서 데이터로부터 서비스를 제공해야 하는 상황을 인식하기 위한 것뿐 아니라 서비스 제공을 위해 필요한 값들을 동적으로 얻어오는 것도 함께 표현한다. 유비쿼터스 환경에서 끊임없이 생성되는 모든 센서 데이터를 컨텍스트 평가의 대상으로 할 수 없으므로, 평가 대상에 포함되는 데이터를 한정해야 한다. 뿐만 아니라 유비쿼터스 서비스 제공을 위한 컨텍스트에 대한 평가는 미리 정해진 데이터에 대해 한 번 수행되는 것이 아니라 흘러가는 센서 데이터에 대해 반복적으로 수행되어야 한다. 따라서 유비쿼터스 서비스를 위한 컨텍스트는 끊임없이 생성되어 들어오는 센서 데이터에 대해 반복적으로 수행되는 연속 질의 형태로 표현된다.

본 논문에서 제안하는 CSML에서는 컨텍스트 정보를 위한 연속 질의를 표현하기 위해 XQuery[13]를 확장한 질의 언어인 XQueryStream(XQuery for Stream Data)을 고안하여 사용한다. XQueryStream은 끊임없이 생성되어 들어오는

```
<Query>            ::= <QueryTarget> <QueryBody>
<QueryTarget>      ::= "using" <SourceDefinition> ( "," <SourceDefinition> ) *
<SourceDefinition> ::= <SourceName> "as" <SourceVariable> (<WindowDefinition>) ?
<WindowDefinition> ::= "within" "(" <WindowRange> ( "," <TumblingLength> )?
                       ( <Unit> ) ? ")"
<WindowRange>      ::= <from> "to" <to>
<Unit>             ::= "event" | "min" | "sec" | "msec"
<QueryBody>        ::= .. .. ..
```

(그림 3) XQueryStream 규격

센서 데이터에 대해 질의 평가 대상을 논리적으로 한정할 수 있도록 하기 위해 시간과 사건에 기반한 윈도우를 지원한다.

(그림 3)은 EBNF(Extended Backus-Naur Form)[14]로 표현된 XQueryStream 규격의 일부분이다. 컨텍스트를 표현하는 질의문 <Query>은 질의 대상을 정의하는 부분 <QueryTarget>과 질의 조건을 나타내는 부분 <QueryBody>으로 구성된다. 사용자는 질의 대상 정의 부분 <QueryTarget>을 통해 질의 조건 평가의 대상이 되는 데이터를 한정한다. 즉, 질의 수행의 입력이 되는 데이터를 정의한다. 또한, 질의 조건 정의 부분 <QueryBody>을 통해 서비스가 수행될 상황과 이때 서비스에 전달할 데이터의 특성을 표현하는데 이는 XQuery 규격을 따른다. <QueryTarget>은 서비스가 수행되어야 하는 상황 판단에 다수의 센서 데이터 스트림 입력을 이용할 수 있도록 하기 위해 키워드 "using" 이후에 소스 정의 부분 <SourceDefinition>이 하나 이상 온다. 소스 정의 부분 <SourceDefinition>은 소스 이름 <SourceName>, 소스 변수 이름 <SourceVariable>, 윈도우 정의 <Window Definition> 부분으로 구성된다. 소스 이름 <SourceName>은 컨텍스트 질의 평가의 대상이 되는 데이터 소스를 나타내는데 논리적 센서 이름이 해당된다. 논리적 센서는 하나 혹은 다수의 물리적 센서로 구성된다. 컨텍스트 평가의 대상이 되는 데이터는 센서가 실시간으로 생성한 센서 데이터뿐 아니라 XML 형태의 뷰를 제공하는 곳에 저장된 과거 이력 데이터도 포함된다. 따라서, 과거 이력 데이터에 대한 XML 뷰를 제공하는 정보 저장소도 하나의 논리적 센서로 가정한다. 만약 윈도우 정의 <WindowDefinition>가 명시되지 않은 경우에는 하나의 센서 데이터가 입력될 때마다 컨텍스트 질의 조건 <QueryBody>을 평가하는 윈도우가 정의된 것으로 간주한다. 윈도우 정의 <WindowDefinition> 부분은 주어진 스트림에서 윈도우 영역 <WindowRange>과 건너띄기 길이 <TumblingLength>, 그리고 윈도우 유형 <Unit> 정보를 가진다. XQueryStream의 윈도우 정의 부분을 이용하여 슬라이딩 윈도우와 랜드마크 윈도우[15]를 표현할 수 있다. <WindowRange>의 <From> 값이 -1 인 경우에는 랜드마크 윈도우임을 나타낸다. 윈도우는 주어진 기간인 <TumblingLength> 간격으로 계속 반복적으로 설정되어 컨텍스트 질의 조건 평가의 입력이 되는 데이터를 한정한다. <WindowRange>와 <TumblingLength>는 <Unit>에 설정된 값에 기반하여 해석된다. <Unit> 값이 "event"인 사

건 기반 윈도우에서는 입력 센서 데이터의 수에 기반하여 질의 대상이 한정되고, <Unit> 값이 "min"(분), "sec"(초), "msec"(1000분의1초)인 시간 기반 윈도우는 시간에 기반하여 질의 대상이 한정된다.

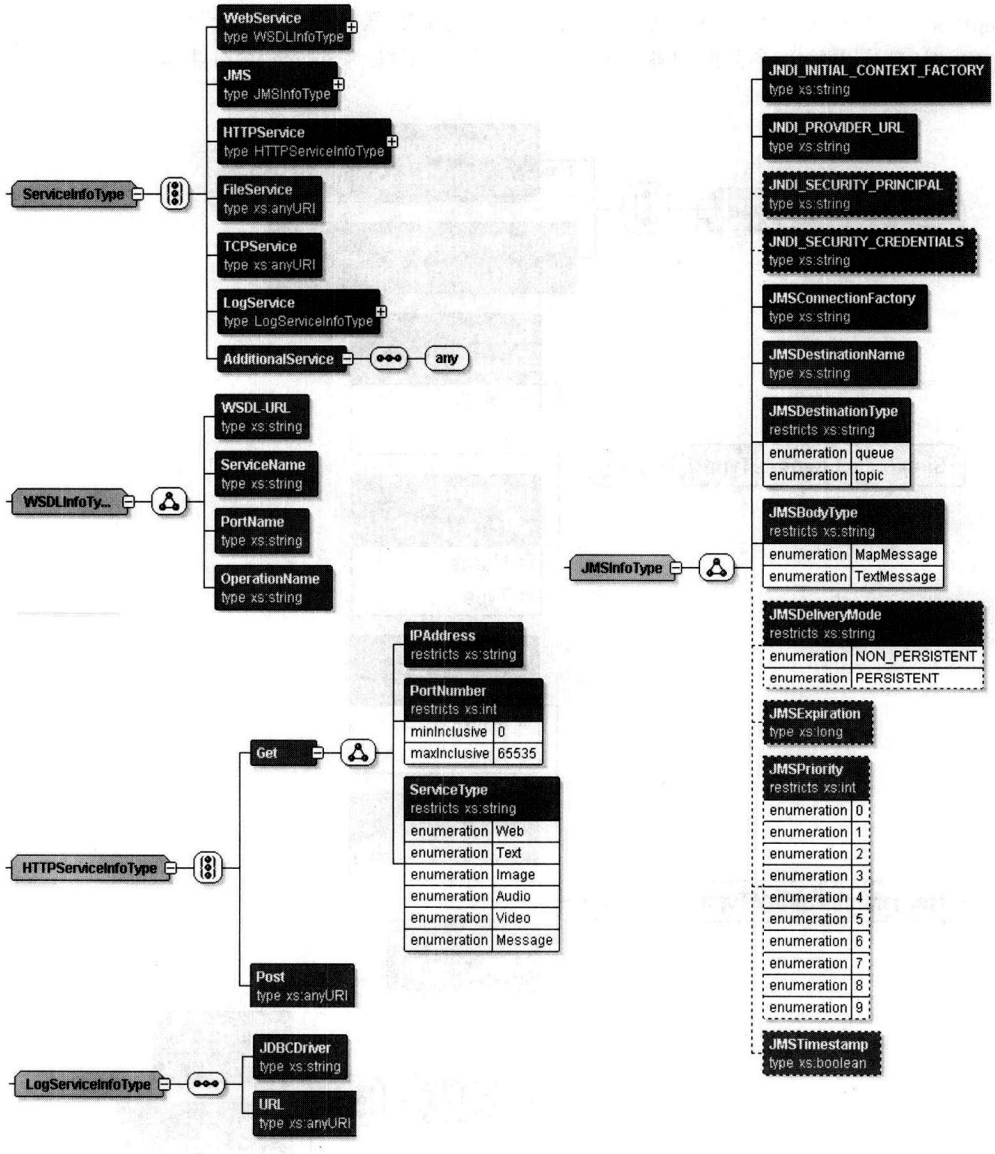
XQueryStream을 이용하여 "회의실(ConferenceRoom)의 최근 10분간의 평균 온도를 매 5분마다 확인하여 20도보다 낮거나 30도보다 높으면 평균온도를 반환하라"는 질의를 표현한다면 다음과 같이 겹침이 있는 슬라이딩 윈도우를 포함하는 질의로 간단히 표현될 수 있다.

```
using "ConferenceRoom" as $src within (0 to 10, 5 min)
let $avgTemp := avg($src/temperature)
where $avgTemp > 30 or $avgTemp <20
return
<Temperature> $avgTemp </Temperature>
```

4.1.2 서비스 정보

서비스 정보 <ServiceInfo> 는 명시된 컨텍스트가 만족되었을 경우에 연계되어 호출될 서비스에 대한 정보를 표현하는데 (그림 4)와 같은 정보로 구성된다. 제안하는CSML의 <SerivceInfo> 앨리먼트을 이용하여 요즘 서비스 개발에 가장 많이 사용되고 있는 유형인 웹 서비스 <WebService>, 자바 메시징 서비스[16] <JMS>, HTTP[17] 서비스 <HTTPService>, 파일 서비스 <FileService>, TCP 서비스 <TCPService> 형태로 컨텍스트에 연계되어 수행될 서비스를 쉽게 표현할 수 있다. 또한, 센서가 센싱한 정보를 다른 응용에서 이용할 수 있도록 하기 위해 JDBC 인터페이스를 지원하는 관계형 데이터베이스에 저장하는 로그 서비스 <LogService>도 표현할 수 있다. 뿐만 아니라 새로운 유형의 서비스가 등장하더라도 <AdditionalService> 구조를 통해 쉽게 표현할 수 있게 했다.

호출되어야 하는 웹 서비스를 표현하기 위한 정보



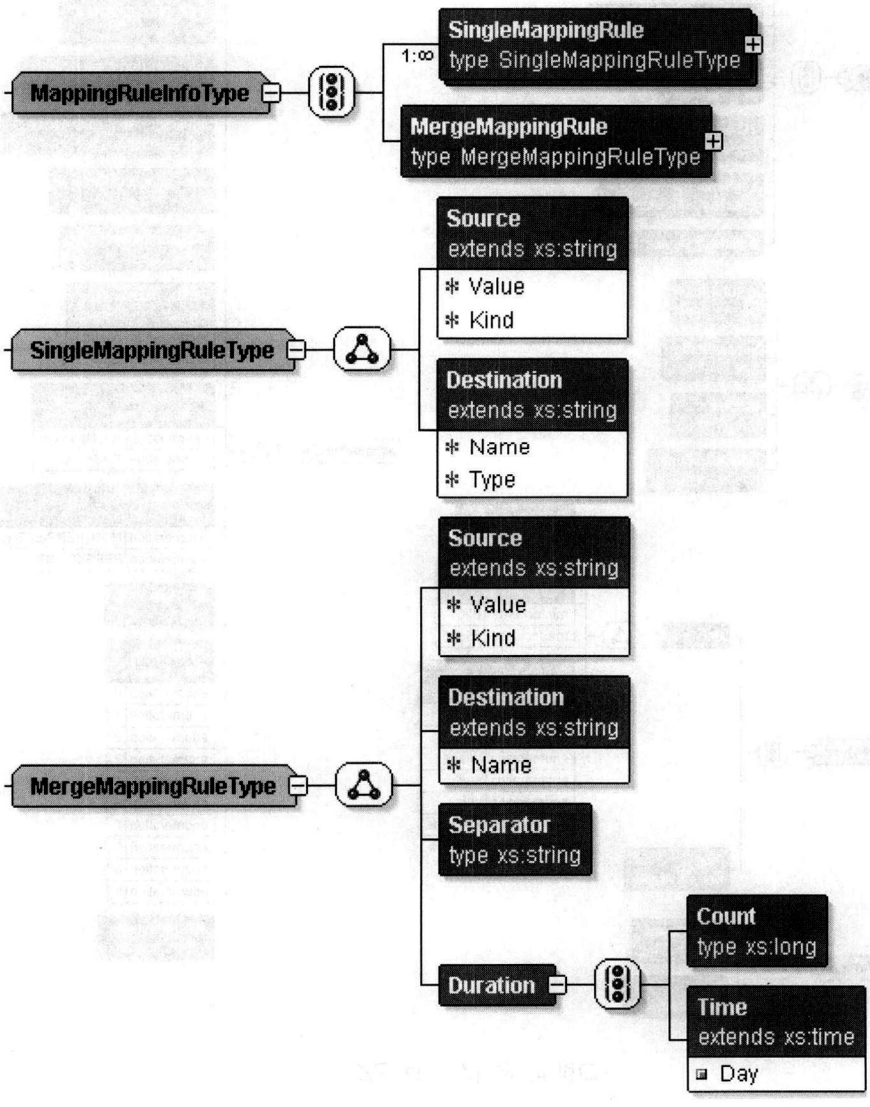
(그림 4) 서비스 정보 구조

<WebService>는 <WSDLInfoType>의 구조를 가지는데, 이는 웹 서비스 호출 메시지 작성을 위해 필요한 정보를 나타낸다. 이 정보는 WSDL(Web Service Description Language)[18] 문서의 위치인 URL(Uniform Resource Locator)[19]을 나타내는 <WSDL-URL>, 서비스 이름인 <ServiceName>, 포트 이름인 <PortName>, 오퍼레이션 이름인 <OperationName>으로 구성되어 있다. 다른 서비스를 표현하는 자세한 정보에 대한 설명은 지면 관계상 생략한다.

4.1.3 매핑 규칙 정보

매핑 규칙 정보 <MappingRuleInfo>는 컨텍스트의 결과로부터 서비스 호출을 위한 입력 정보를 만들기 위한 규칙 정보로 (그림 5)와 같이 하나 이상의 단일 매핑 규칙 <SingleMappingRule> 혹은 하나의 병합 매핑 규칙 <MergeMappingRule>으로 표현된다. 두 규칙 모두 컨텍스트의 결과로부터 서비스의 인자를 만들기 위한 규칙이지만 서비스가 호출되는 주기가 다르다. 단일 매핑 규칙 <SingleMappingRule>이 사용되면 컨텍스트 <ContextInfo>를 만족할 때 사용자 관심 사항을 추출한 결과가 도착하는 즉시 이를 서비스의 인자로 하여 서비스가 호출되지만, 병합 매핑 규칙 <MergeMappingRule>이 명시되면 컨텍스트 <ContextInfo>를 만족하는 결과가 도착하는 즉시 서비스가 호출되는 것이 아니라 일정한 개수의 결과 혹은 일정 시간 동안의 결과를 병합하여 하나의 서비스 입력 인자로 변환하여 서비스를 호출한다.

단일 매핑 규칙 <SingleMappingRule>은 컨텍스트의 결과로부터 서비스 호출을 위한 하나의 인자를 만드는 규칙 정보로 <Source>와 <Destination>으로 구성되는 구조를 가진다. 컨텍스트의 결과로부터 어떤 값을 취하는지에 대한 정보를 나타내는 <Source>는 속성으로 value와 kind를 가진다. 속성 value는 <Destination>의 입력으로 사용될 값을 나타내며, 속성 kind는 컨텍스트로부터 추출한 정보를 이용할 것인지 혹은 고정된 임의의 값을 이용할 것인지를 나타낸다. 속성 value를 표현하는데 있어서 속성 kind 값이 "dynamic"인 경우에는 <ContextInfo>를 통해 수집한 정보로부터 서비스 수행에 필요한 정보를 추출하기 위한 경로식 정보가 들어가며, "static"인 경우에는 임의의 고정된 값이 들어간다. <Source>에서의 값이 서비스의 어떤 인자에 매



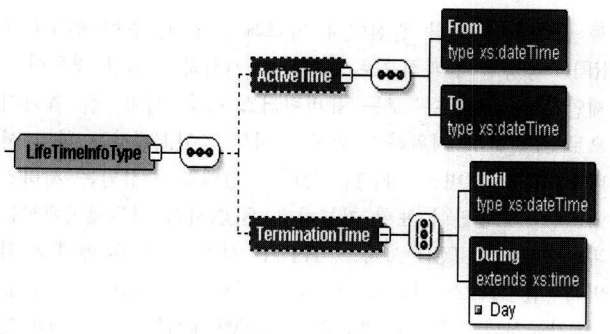
(그림 5) 매핑 규칙 정보 구조

핑되는가의 정보를 나타내는 <Destination>은 속성으로 name과 type을 가진다. 속성 name은 서비스 인자 이름을 나타내고, 속성 type은 그 인자의 타입 정보를 나타낸다.

병합 매핑 규칙 <MergeMappingRule>은 단일 매핑 규칙에 더하여 여러 결과를 어떻게 병합할 것인지에 대한 규칙을 추가적으로 가진다. <Separator>는 여러 개의 컨텍스트 처리 결과를 하나의 서비스 입력 인자로 변환하여 서비스에 전달할 때 각 처리 결과를 구분하는데 사용된다. 또한, <Duration>을 통해 병합될 결과를 시간(<Time>)과 개수(<Count>)에 기반하여 한정하는 방법을 제공한다.

4.1.4 생명 주기 정보

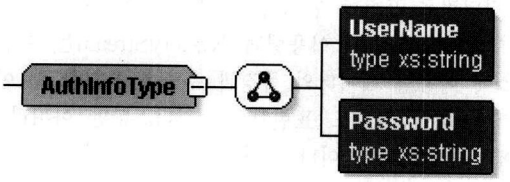
생명 주기 정보 <LifeTimeInfo>는 서비스가 실제 제공되는 시간 관련 정보로 (그림 6)과 같이 <ActiveTime>과 <TerminationTime>으로 구성된다. <ActiveTime>은 서비스가 하루 중 언제부터(<From>) 언제까지(<To>) 의미가 있는지를 나타낸다. <TerminationTime>은 사용자에게 서비스가 더 이상 제공되지 않고 철회되는 시간을 나타내는데 <Until> 혹은 <Duration>으로 구성된다. <Until> 앨리먼트를 이용하여 철회되는 특정 시점을 나타낼 수 있고, <Duration> 앨리먼트로는 서비스가 제공되는 얼마 동안의 기간(예, 10일 10시간)을 명시할 수 있다. 또한, 이를 확장하여 서비스 종료 조건을 횟수로 제한할 수도 있다.



(그림 6) 생명 주기 정보 구조

4.1.5 인증 정보

인증 정보 <AuthInfo>는 사용자에게 유비쿼터스 서비스를 제공하기 위해 접근 제어가 필요한 경우에 사용되는 정보로, (그림 7)과 같이 사용자 이름 <UserName>과 패스워드 <Password>로 구성된다.



(그림 7) 인증 정보 구조

인증 정보를 포함하여 CSML을 통해 표현된 유비쿼터스 서비스를 나타내는 정보는XML 문서의 무결성과 기밀성 보장을 위해 전자서명과 XML 암호화를 통해 암호화되어 미들웨어에 전달될 필요성이 있다.

4.2 CSML을 이용한 센서 데이터와 서비스 연계 표현 예

유비쿼터스 응용의 예로 많이 알려진 유비쿼터스 샵에서 이용될 수 있는 다음과 같은 할인 쿠폰 제공을 위한 컨텍스트 드리븐 서비스(이하, 할인 쿠폰 제공 서비스)가 있다고 가정하자.

"매장 개장 10주년을 맞이하여 4월 30일까지 매일 오전 10시부터 11시 사이에 매장에 입장하는 고객에게 일부 물건을 10% 할인된 가격에 구매할 수 있는 쿠폰을 제공하는 서비스를 한다. 쿠폰은 고객이 밀고 있는 쇼핑 카트에 장착된 웹패드를 통해 전달된다."

```
<UbiquitousServiceInfo>
 <ContextInfo><![CDATA[
  using "DS_Entrance" as $src
   for $cart in $src/Cart
   return
     <Result>
       <CartID> {$cart/CartID/text()}
</CartID>
     </Result>
  ]]>
 </ContextInfo>
 <ServiceInfo>
  <WebService>
   <WSDLInfo>
    <WSDL-URL>
http://192.168.1.10:8080/UShop.wsdl </WSDL-URL>
     <ServiceName>Coupon</ServiceName>
     <PortName>UShop</PortName>
     <OperationName> issueDiscountCoupon
</OperationName>
   </WSDLInfo>
  </WebService>
 </ServiceInfo>
 <MappingRuleInfo>
   <SingleMappingRule>
    <Source value="/Result/CartID"
kind="dynamic" />
    <Destination name="CartId" type="string"/>
  </SingleMappingRule>
  <SingleMappingRule>
    <Source value="10" kind="static" />
    <Destination name="DiscountRate"
type="long"/>
  </SingleMappingRule>
 </MappingRuleInfo>
 <LifeTimeInfo>
  <ActiveTime>
     <From>10:00:00</From> <To>11:00:00 </To>
    </ActiveTime>
  <TerminationTime>
     <Until> 2007-04-30T24:00:00.000+09:00
</Until>
   </TerminationTime>
 </LifeTimeInfo>
</UbiquitousServiceInfo>
```

(그림 8) 유비쿼터스 샵 응용에서 할인 쿠폰 제공 서비스를 위한 CSML

할인 쿠폰 제공 서비스에서 컨텍스트는 손님이 매장을 방문하는 것이고, 서비스는 매장에 입장한 고객에게 10% 할인 쿠폰을 발행하는 것이다. 이를 CSML로 표현하면 (그림 8)과 같다. 매장 입구에 설치된 센서를 통해 센싱되어 입력되는 센서 데이터로부터 매장에 들어온 고객과 관련된 쇼핑카트 정보를 추출하는 것은 <ContextInfo> 앨리먼트와 같이 XQueryStream 질의문으로 표현될 수 있다. 또한, 고객에게 할인 쿠폰을 발행해주는 웹 서비스는 서비스 이름이 Coupon, 포트 이름이 UShop이고, 오퍼레이션 이름이 issueDiscountCoupon 인데 이는 <ServiceInfo> 앨리먼트와 같이 표현될 수 있다. 서비스의 인자로 쇼핑 카트 정보와 할인율 정보가 필요하다. 쇼핑 카트 정보 CardId는 <MappingRuleInfo> 앨리먼트에 표현된 것처럼 <ContextInfo> 앨리먼트에 표현된 XQueryStream 질의문의 수행 결과로부터 /Result/CartID 경로식을 가지는 값을 추출하여 전달하고, 할인율 정보 DiscountRate로는 정적 값인 10을 전달한다. 할인 쿠폰 제공시간은 2007년 4월 30일까지 매일 오전 10시부터 11시 사이인데 이는 <LifeTimeInfo> 앨리먼트로 표현할 수 있다.

5. 정성적인 비교

본 장에서는 제안하는 CSML과 관련 연구간의 유사성과 차이점을 분명히 하기 위해 3장에서 살펴본 유비쿼터스 서비스를 표현하는데 필요한 정보 분류를 기준으로 하여 다양한 유비쿼터스 서비스를 얼마나 잘 표현할 수 있는지를 정성적인 관점(qualitative perspective)에서 비교 평가를 했다. <표 1>은 제안하는 CSML과 관련 연구들간의 정성적인 평가 요약으로, 항목에서의 숫자((1), (2), (3), (4), (5), (6))는 3장에서의 해당 숫자가 가리키는 분류에 비교 항목이 연관있음을 의미한다.

첫째, 서비스가 제공되어야 하는 다양한 컨텍스트 정보를 얼마나 잘 표현할 수 있는지의 관점에서 살펴 보았다. 서비스가 수행되어야 하는 상황을 묘사하는데 있어서 제안하는 CSML은 모든XML 데이터에 대해 표현할 수 있는데 비해, ALE 규격[3]은 EPC 인코딩된 데이터에 대해서만, SUN사의 미들웨어[5]의 방법에서는 관계형 모델을 가지는 데이터에 대해서만 표현할 수 있다. 관련 연구들에서는 실시간으로 수집된 데이터만이 컨텍스트의 대상이 될 수 있지만, 제안하는 CSML은 실시간 센서 데이터뿐만 아니라 XML 뷰를 제공할 수 있는 곳에 저장된 과거 이력 데이터도 컨텍스트의 대상이 될 수 있다. 또한, 관련 연구의 방법들은 EPC 패턴 정의 혹은 미리 정의된 필터와 연산자를 통해 제한된 형태로만 관심있는 컨텍스트에 대한 표현을 제공하고 있지만, 제안하는 방법은 XQuery를 XML 데이터 스트림에 적용 가능하도록 확장한XQueryStream을 이용하여 서비스가 제공되어야 하는 다양한 컨텍스트를 표현할 수 있다.

둘째, 컨텍스트 조건이 만족될 때 호출될 다양한 서비스

〈표 1〉 유비쿼터스 서비스 표현 방법의 정성적인 관점에서 비교

항목 \ 방법		제안하는 CSML	ALE	SUN 사의 미들웨어 방법
(1)	센서 데이터 형식	XML 로 표현된 데이터	EPC 인코딩된 데이터	관계형 모델을 가지는 데이터
	컨텍스트의 대상	실시간 센서 데이터뿐 아니라 과거 이력 데이터	실시간 EPC 데이터	실시간 센서 데이터
	관심 컨텍스트 표현 방법	연속 질의 언어 XQueryStream 을 이용한 질의	이벤트 사이클 리포트 스펙을 이용한 EPC 패턴 정의	미리 정의된 단순 필터 연결
	컨텍스트 표현력	풍부	빈약	보통
(2)	목표 서비스	범용 유비쿼터스 응용 서비스	RFID 응용 서비스	RFID 응용 서비스
	표현 가능한 서비스 구성 유형	File, HTTP, TCP, JMS, Web Services, Log 서비스	File, TCP, HTTP	File, HTTP, TCP, JMS, Web Services
(3)	서비스에 이용할 데이터 표현	XQueryStream을 통한 관심 데이터 가공	이벤트 사이클 리포트 스펙	Filter 연산자 연결로 표현
	컨텍스트와 서비스 연계	매핑룰 이용하여 다양한 연계 표현	Subscribe API 이용하여 단순 연계 표현	Filter와 logger 연산자 연결로 단순 연계 표현
(4)	서비스 호출 주기	시간과 개수에 기반하여 표현	시간에 기반하여 표현	표현 불가
(5)	서비스 제공 시간	표현 가능	표현 불가	표현 불가
	서비스 종료 시간	표현 가능	표현 불가	표현 불가
(6)	서비스 사용 위한 인증	아이디와 패스워드 정보에 기반하여 표현 가능	표현 불가	표현 불가

를 표현하는 능력 관점에서 비교해 보았다. 관련 연구들은 RFID 응용을 효과적으로 표현하고 지원하기 위한 것들이고, 제안하는 CSML은 모든 유비쿼터스 응용 서비스를 효과적으로 표현하고 지원하기 위한 것이다. ALE에서는 파일 서비스, TCP 서비스, HTTP 서비스 형태로 구성되는 서비스에 대한 호출만이 표현 가능하고, SUN사의 미들웨어에서는 파일 서비스, TCP 서비스, HTTP 서비스, 자바 메시징 서비스, 웹 서비스 형태로 구성되는 서비스에 대한 호출이 표현 가능하다. 하지만 제안하는 CSML에서는 파일 서비스, TCP 서비스, HTTP 서비스, 자바 메시징 서비스, 웹 서비스뿐만 아니라 로그 서비스 형태로 구성되는 서비스에 대한 호출도 표현할 수 있다.

셋째, 서비스에 이용할 데이터에 대한 표현 능력 관점에서 비교해 보았다. ALE에서는 이벤트 사이클 리포트 스펙이라는 형식을 이용한 EPC 패턴 정의를 통해 서비스에 이용될 데이터의 특징을 표현할 수 있고, SUN 사의 미들웨어에서는 단순 필터 연산자 연결로 서비스에 이용할 데이터에 대한 필터링을 표현할 수 있다. 반면에 제안하는 CSML에서는 컨텍스트 표현에 이용했던 XQueryStream을 이용하여 서비스에 이용할 데이터의 특징과 추출을 표현할 뿐 아니라 데이터에 대한 가공도 표현할 수 있다. 뿐만 아니라 관련 연구에서는 API 호출이나 연산자 연결을 통해 서비스에 이용할 데이터를 단순히 서비스에 전달하는 것만을 표현할 수 있지만, 제안하는 CSML에서는 다양한 매핑룰을 이용하여

컨텍스트로부터 추출한 동적인 정보나 정적인 정보를 서비스 호출시 인자로 활용하는 것을 나타낼 수 있게 함으로써 다양한 연계를 표현할 수 있다.

넷째, 서비스가 수행되는 주기 정보를 표현할 수 있는지의 관점에서 살펴 보았다. SUN 사의 미들웨어에서는 서비스 호출 주기에 대한 표현이 불가능하고, ALE에서는 이벤트 사이클 바운더리 스펙을 통해 시간에 기반한 주기를 표현할 수 있다. 하지만, 제안하는 CSML에서는 시간에 기반한 서비스 호출 주기뿐만 아니라 이벤트 개수에 기반한 서비스 수행을 위한 호출 주기를 표현할 수 있다.

다섯째, 서비스에 대한 생명 주기 정보의 표현 능력 관점에서 비교해 보면, 관련 연구에서는 필요시 서비스에 대한 등록과 삭제를 요청함으로써 서비스의 생명 주기를 부여해야 한다. 하지만, 제안하는 CSML에서는 시간과 개수에 기반하여 서비스 제공 시간과 서비스 종료 시간을 표현함으로써 서비스에 대한 다양한 생명 주기를 부여할 수 있다.

여섯째, 서비스 사용에 대한 인증을 위한 정보의 표현 관점에서 살펴 보면, 제안하는 CSML에서는 아이디와 패스워드에 기반하여 서비스 사용을 위한 인증을 표현할 수 있다. 하지만, 관련 연구에서는 이를 표현할 수 없다.

이상과 같이 정성적인 비교 평가를 해볼 때 제안하는 CSML이 서비스가 호출되어야 하는 다양한 컨텍스트에 대한 표현력, 컨텍스트 조건이 만족되었을 때 수행되어야 하는 서비스에 대한 표현력, 서비스 호출에 이용할 데이터에 대한 표현력, 다양한 컨텍스트와 서비스와의 연계에 대한 표현력, 서비스가 수행되는 생명 주기에 대한 표현력, 서비스 인증을 위한 정보 표현력 측면에서 관련 연구의 방법들보다 우수하다. 따라서, 제안하는 CSML이 관련 연구의 방법들 보다 우리 주변에 산재하는 다양한 센서들이 생성하는 데이터를 서비스에 이용하는 다양한 유비쿼터스 서비스를 표현하는데 더 쉽고 효과적이라고 할 수 있다.

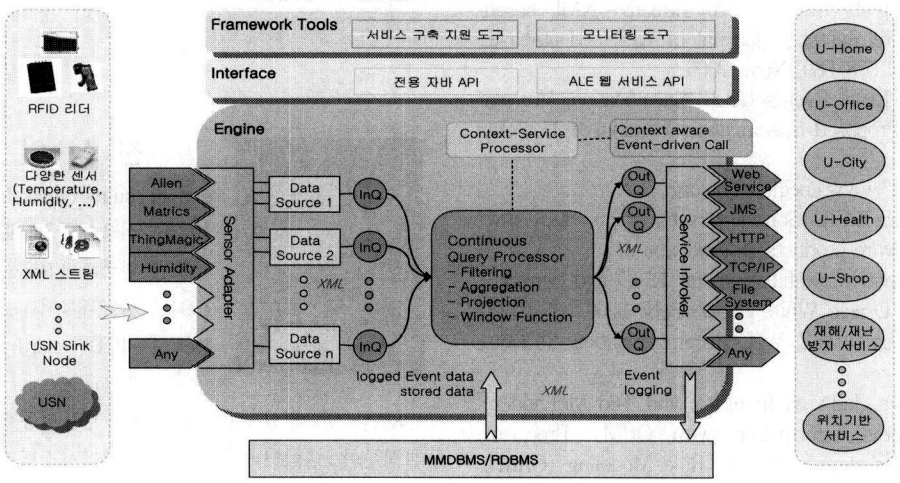
6. 사례 시스템

제안하는 CSML의 초기 버전(생명 주기 정보 관련 내용이 포함되지 않음) 규격을 수용하는 센서 데이터 스트림 처리 미들웨어인 UbiCore[2]를 개발하였다. UbiCore는 유비쿼터스 환경에서 다양한 센서들로부터 생성된 데이터를 이용하여 사람들에게 편의를 제공하는 유비쿼터스 서비스를 개발하는데 도움을 주는 미들웨어이다. UbiCore는 유비쿼터스 환경에서 운영될 수 있는 이기종 센서들을 통합 관리할 수 있으며, 이들 센서로부터 생성되는 대량의 센서 데이터 스트림에 대하여 XML 기반의 연속 질의 처리를 수행할 수 있고, 연속 질의 처리 결과 생성되는 결과 스트림을 원하는 외부 서비스 호출을 통해 전달하는 기능이 있다.

UbiCore를 이용하여 유비쿼터스 서비스를 개발하는 경우, 서비스 개발자는 어플리케이션 프로그램을 개발한 후 서비스 구축 지원 도구나 전용 자바 API를 이용하여 유비쿼터스 서비스에 사용될 논리적 센서인 데이터 소스(Data Source)를 등록하고, 해당 논리적 센서들로부터 들어온 센서 데이터를 이용할 유비쿼터스 서비스를 CSML로 표현하여 등록하기만 하면 된다. CSML이 등록되는 과정에서 컨텍스트 정보는 UbiCore 엔진의 연속 질의 처리기(Continuous Query Processor)에게 전달되고, 서비스 정보, 매핑 규칙 정보, 생명 주기 정보, 인증 정보와 같은 서비스 호출을 위한 정보는 서비스 호출자(Service Invoker)에게 전달되어 처리된다. (그림 9)에 나타난 것처럼 UbiCore에서는 센서 데이터가 센서 어뎁터(Sensor Adapter)를 통해 입력 큐(InQ)에 들

어오면 연속 질의 처리기가 서비스 호출을 위한 컨텍스트를 만족하는지를 평가하여, 만족하면 서비스 호출을 위한 정보를 추출하여 출력 큐(OutQ)에 넣는다. 서비스 호출자는 서비스 호출 주기 조건을 만족하는 지를 확인한 후 출력 큐의 값을 이용하여 서비스 호출을 위한 인자를 구성하여 서비스를 자동으로 호출한다.

뿐만 아니라 CSML의 효용을 검증하기 위해 UbiCore를 이용하여 초고층 건설 현장에서 활용될 수 있는 RFID 기반 자재 및 노무 관리, 콘크리트 양생 관리, 그리고 센서 네트워크 기반의 기둥 수축 관리를 위한 시범 시스템을 구축해 보았다[20]. 제안하는 CSML을 이용하여 초고층 건설 현장



(그림 9) UbiCore의 구조 및 센서 데이터 처리 흐름

에서 활용될 수 있는 다양한 유비쿼터스 서비스들을 쉽게 표현할 수 있었다.

7. 결론 및 향후 연구

본 논문에서는 센서 데이터를 이용하는 유비쿼터스 서비스를 나타내는데 필요한 정보를 분류하고, 이를 효과적으로 표현하는 컨텍스트-드리븐 서비스 마크업 언어인 CSML을 제안했다. 제안하는 CSML은 XML 형식을 따르는데 서비스가 호출되어야 하는 상황을 인지하기 위한 컨텍스트 정보, 컨텍스트 정보 조건이 만족되었을 때 수행되어야 하는 서비스 정보, 컨텍스트로부터 추출한 결과를 서비스의 인자로 매핑시키는 규칙을 나타내는 매핑 규칙 정보, 컨텍스트-드리븐 서비스가 의미있는 시간을 나타내는 생명주기 정보, 그리고 서비스 사용을 위한 인증 정보로 구성된다. 본 논문에서 제안하는 CSML을 이용하면 다양한 비정형의 형식을 가지는 센서 데이터에 대해 서비스가 제공되어야 하는 컨텍스트 조건과 서비스에 이용할 데이터의 특성을 쉽게 표현할 수 있으며, 컨텍스트 조건이 만족되었을 때 수행되어야 하는 다양한 유비쿼터스 서비스와 연계 정보를 쉽게 표현할 수 있다.

상용 센서 데이터 스트림 처리 미들웨어들이 본 논문에서 제안하는 컨텍스트-드리븐 서비스 마크업 언어와 같은 공통된 규격을 수용하도록 구현된다면 유비쿼터스 서비스 개발자들은 어떤 미들웨어를 사용할지에 대한 고민없이 서비스 개발에 집중할 수 있게 될 것이다. 따라서, RFID 응용 개발자와 미들웨어 개발자들의 편의를 위해 EPCglobal을 중심으로 EPC 데이터에 대한 것을 필터링 및 수집 인터페이스에 대한 표준화를 한 것처럼, 센서 데이터와 유비쿼터스 서비스의 연계를 표현할 수 있는 방법에 대한 표준화 활동이 필요하다.

참 고 문 헌

[1] Mark Weiser, "The computer for the 21st Century," Scientific American, Sept., 1991.

[2] Hun Soon Lee et al, "UbiCore : An Effective XML-based RFID Middleware System," Journal of KISS : Database, Vol.33, No.6, pp. 578-589, Nov., 2006.

[3] The Application Level Events (ALE) Specification, Version 1.0. EPCglobal Proposed Specification Version of 11 Feb., 2005.

[4] EPCglobal, http://www.epcglobalinc.org/

[5] Sun Java System RFID Software, version 2.0, http://www.sun.com/software/products/rfid/index.xml

[6] Enterprise Information Architecture for RFID and Sensor-Based Services, Oracle White Paper, Feb., 2006.

[7] XML Schema Part 0, 1, 2, W3C Recommendation, 28 Oct., 2004.

[8] Karen Henricksen, Jadwiga Indulska, and Ted McFadden, "Modelling context information with ORM," Proc. of Iinternational Workshop on Object-Role Modeling (ORM), volume 3762 of Lecture Notes in Computer Science, pages 626-635. Springer, 2005.

[9] Sven Buchholz, Thomas Hamann, Gerald Hubsch, "Comprehensive Structured Context Profiles (CSCP): Design and Experiences," Proc. of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.

[10] Object Role Modeling, http://www.orm.net

[11] Resource Description Framework, http://www.w3.org/RDF/

[12] Extensible Markup Language (XML) 1.0 (Fourth Edition), W3C Recommendation, 16 August, 2006.

[13] XQuery 1.0: An XML Query Language, W3C Recommendation 23 January, 2007.

[14] ISO/IEC 4977, Information technology - Syntatic metalanguage Extended BNF, Dec. 1996.

[15] S. Chandrasekaran, A. Deshpande, M. Franklin, J. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, and M. Shah. "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World," In Proc. of Conference on Innovative Data Systems Research, Jan., 2003.

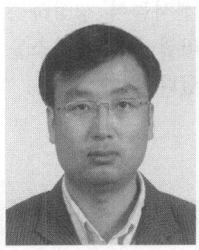
[16] Java Message Service Specification 1.1, Sun Microsystems Inc., 12 April, 2002.

[17] RFC2616, Hypertext Transfer Protocol HTTP/1.1, http://rfc.net/rfc2616.html

[18] Web Service Description Language, http://www.w3.org/TR/wsdl/

[19] RFC1738, Uniform Resource Locators (URL), http://rfc. net/rfc1738.html

[20] 이미영 외, "첨단 유비쿼터스 물류 관리 및 유지 관리의 핵심 기술 스마트 객체 처리 기술," 대한 건축 학회지, 제 51권 제 6호, pp. 33~36, 2007년 6월.



이 훈 순

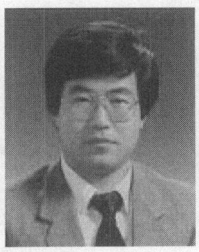
e-mail : hunsoon@etri.re.kr

1997년 충남대학교 컴퓨터과학과(이학사)

1999년 충남대학교 대학원 컴퓨터과학과 (이학석사)

1999년~현 재 한국전자통신연구원 선임연구원

관심분야 : 데이터베이스, 데이터 스트림 처리, 자료저장시스템



진 성 일

e-mail : sijin@cnu.ac.kr

1978년 서울대학교 계산통계학과(학사)

1980년 한국과학기술원 전산학과(이학석사)

1994년 한국과학기술원 전산학과(이학박사)

1987년~1989년 Northwestern대학 전산학과 객원교수

1983년~현 재 충남대학교 전기정보통신공학부 교수

관심분야 : 데이터베이스, 멀티미디어시스템, 소프트웨어공학, 시뮬레이션모델링