

유비쿼터스 데이터 통합관리를 위한 동적 명세 도구개발

박상현^{01,2} 민수영¹ 주현태²

¹전자부품연구원

{shpark, minsy}@keti.re.kr

²한국외국어대학교

cisc@hufs.ac.kr

An Implementation of Dynamic Specification Tool for Ubiquitous Data Integration

Sang Hyun Park^{01,2} Soo Young Min¹ Hyun Tae Ju²

¹Korea Electronics Technology Institute

²Hankuk University of Foreign Studies

요 약

유비쿼터스 컴퓨팅이란 언제, 어디서나, 아무 때나 네트워크를 통해 정보를 접할 수 있는 사회를 말한다. 이러한 유비쿼터스 컴퓨팅 환경에서 사용자가 정보를 다양한 디바이스들로부터 받아들이고 받아들인 정보를 바탕으로 사용자에게 맞는 이벤트가 수행될 수 있는 통합 환경을 "UDI (Ubiquitous Data Integration) 서비스 플랫폼"¹⁾이라 한다. 본 논문에서는 이러한 UDI 서비스 플랫폼의 구성 중 최상위 응용 계층으로 Cyber View 라는 자바 기반 명세도구를 개발하여 사용자가 정의한 이벤트를 다양한 디바이스를 통해 실행할 수 있는 사용자정의 도구 개발에 관한 연구를 기술할 것이다.

1. 서 론

네트워크 및 하드웨어 기술의 혁신적인 발전의 덕택으로 텔레매틱스, 지능형 기전 등을 포함하는 유비쿼터스 컴퓨팅 환경이 가시화되고 있다. 유비쿼터스 컴퓨팅은 다양한 네트워크화 컴퓨터, PDA, 휴대전화, 센서네트워크 등에 이르는 다양한 장치들을 실제 공간상에서 사용하여 사이버 공간을 제공하는 사용자 중심의 컴퓨팅 환경을 의미한다. 이와 같은 유비쿼터스 컴퓨팅 환경에서는 서버, 휴대장비, 데이터, 애플리케이션의 개수 및 복잡도가 증가되므로, 기존의 기술들로 관리할 경우, 다양한 문제들이 발생하고 복잡도가 증가되어 사실상 애플리케이션의 개발과 관리가 불가능하게 된다. 따라서 기존의 XML[1][2][3]과 SyncML[5]에서와 같은 단순한 데이터 표시에 대한 가공 및 일괄적인 동기화를 넘어서, 개개인의 위치 및 컨텍스트에 따른 효과적인 데이터 분배 및 보호 등을 모두 제공하기 위해 UDI(Ubiquitous Data Integration) 서비스 플랫폼이 제안되었다.

본 논문의 2장에서는 관련연구로 UDI 서비스 플랫폼에 대해소개하고, 3장에서는 UDI 서비스 플랫폼의 최상위 응용 계층으로 Cyber View 라는 자바 기반 사용자

명세도구 개발에 관해에 관해 기술하고, 마지막 4장에서는 결론 및 향후 연구계획에 대해 논의하고자 한다.

2. 관련연구

UDI 서비스 플랫폼은 유비쿼터스 컴퓨팅환경에서 언제 어디서, 어느 디바이스에서든지 동일한 데이터 환경을 제공하고, 사용자의 위치나 문맥, 디바이스의 용량 등을 고려하여 가장 적합한 데이터들이 가장 적합한 디바이스에 적재되어 서비스가 제공되도록 하고, 다수의 디바이스들에 분산된 데이터들이 적절히 보호되도록 하고, 다수의 사용자로 콘텐츠가 분배될 때의 네트워크 성능 이슈를 효과적으로 해결하는 서비스 프로토콜과 플랫폼의 개발을 목표로 한다. 이를 위해 UDI에서는 사용자의 데이터, 응용, 이들을 수행하는 컴퓨팅 자원의 관리 정책인 논리적 스키마 Cyber View를 정의하며, 이와 별도로 유비쿼터스 환경을 구성하는 물리적인 네트워크, 서버, 휴대 디바이스, 센서, 액추에이터 등의 물리적 인프라 및 그들의 위치를 Vehicle로 정의한다. 유비쿼터스 환경에서는 응용 및 데이터에 해당하는 Cyber View가 수행되는 환경인 Vehicle이 시시각각 변하게 되는데, 이러한 동적 변화에도 불구하고 사용자 및 애플리케이션 개발자들이 항상 일관적이고 효과적인 Cyber View를 제공받게 하여 손쉬운 애플리케이션 개발 및 관리를 가능하게 한다. 이러한 관리/운용의 편의성을 제공하기 위하여 논리적 스

1) 본 연구는 산업자원부 지원사업인 "전자부품기반기술 개발사업"의 일환으로 "UDI 핵심 모듈 및 시스템 기술 개발"과제 수행에 의해 진행된 연구임.

키마(Cyber View)와 물리적인 장치(Vehicle)를 서로 연계하는 UDI 메타 서비스 미들웨어가 개발된다.

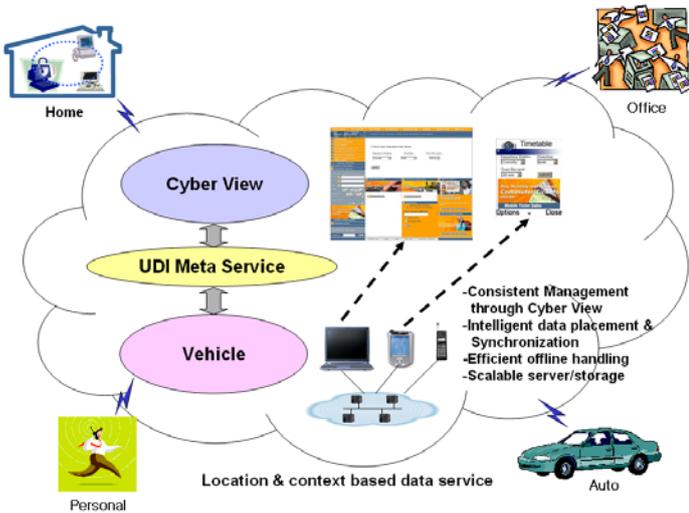


그림 1 UDI 서비스 플랫폼

3. Cyber View 명세 도구

Cyber View 명세 도구 개발은 UDI 서비스 플랫폼의 구성에 있어서 최상위 응용 계층에 해당한다. Cyber View 개발을 위해서는 UDI 서비스 플랫폼의 동작을 정의하는 논리적 스키마 정의와 기본적인 스키마를 동적으로 재구성이 가능하여 사용자가 원하는 동작을 정의할 수 있는 환경을 제공하는 사용자 명세 GUI (Graphic User Interface)의 개발이 필요하다.

3.1 Cyber View 명세

Cyber View는 UDI의 관리/운용의 편의성을 제공하는 논리적인 스키마로 다음과 같은 명세 방법을 개발한다.

첫째, Event, Location, Context 기반 Behavior 명세 기술 개발은 각 Event, Location, Context 별로 필요한 데이터 및 프로그램의 동작을 명세하여 다양한 Vehicle에서 수행 가능하도록 개발하여 다른 Cyber View entity와 통신을 위한 인터페이스를 명세하여 협업을 지원하고 기존 애플리케이션 서버 및 모바일 소프트웨어 환경에 구현될 수 있는 연동성을 제공하는 기술이다.

둘째, 데이터 및 관리 정책 명세는 사용자의 의도를 분석하여 필요할 것으로 예측되는 데이터를 디바이스로 Pre-fetching 또는 캐싱하기 위한 정책, 데이터 속성, 생명주기, 응용특성, 저장비용을 고려하여 데이터 저장위치 결정하는 명세이다.

셋째, 데이터 보호 (data protection) 정책은 데이터의 가용성을 높이기 위한 복제정책 명세 및 복제 본 관리 정책 개발, 데이터의 분실과 도난 시 보호를 위해 암호화 지원한다.

넷째, 콘텐츠 분배정책은 사용자의 콘텐츠에 대한 권한과 분배 설정에 따라 Vehicle내에 데이터 배분을 지원한다.

다섯째, 디바이스 타입별로 요구되는 데이터의 속성 명세는 최소 필요 데이터 (응용 동작을 위한 최소 데이터 집합), 선택적인 데이터 (디스플레이, 메모리크기, 네트워크 링크 등)를 Vehicle의 컴퓨팅 자원에 따라 선택적으로 사용가능한 데이터 속성을 명세 한다.

이러한 명세 방법은 XML 기반의 스키마로 작성되고, XML 형식으로 표현된다. Cyber View 명세 도구는 이렇게 잘 정의된 XML 스키마를 읽어 들여 사용자에게 기본적으로 정의된 내용을 보여주고, 쉬운 인터페이스를 통하여 사용자가 쉽게 원하는 작업을 수행할 수 있도록 GUI를 제공한다. 사용자는 Cyber View 명세 도구를 통하여 UDI의 관리/운용을 편의성을 제공 받을 수 있다.

3.2 스키마 기반의 인터페이스 구현

XML 스키마는 XML 문서에서 각 속성과 요소간의 상호관계를 표현하는 방법 중의 하나로, W3C(World Wide Web Consortium)의 권고사항이다. 비슷한 역할을 하는 것으로 DTD(Document Type Definition)나 SOX(Simple Object XML)와 같은 것들이 있으나 스키마는 XML 문서 형식을 그대로 유지하고 있기 때문에 평범한 XML 문서와 마찬가지로 별도의 도구 없이 XML 파서에 의한 직접적인 처리가 가능하다.

Cyber View 명세 도구에 있어서 스키마는 작업에 필요한 데이터의 종류와 형태 등을 나타내는 역할을 하며, Cyber View 명세 도구에서는 스키마의 내용을 바탕으로 명세 데이터 생성을 위한 인터페이스를 제공한다.

예를 들어, 이벤트 처리에 관한 스키마의 일부는 다음과 같은 형식으로 표현된다.

```
<xsd:complexType name="EventType">
<xsd:sequence>
  <xsd:element name="Schedule" type="ScheduleInfoType">
    정해진 시간에 Job , 시작 시간 및 주기 설정
  </xsd:element>
  .....
```

이 스키마의 내용에 의하면 이벤트 처리에 관한 명세 항목들의 종류와 각 항목에 대한 간단한 설명을 얻을 수 있다. 이 내용을 바탕으로 그림 2와 같은 사용자 인터페이스를 구성하고 사용자가 각 항목에 대한 작업을 수행할 수 있도록 한다. 이벤트 관련 명세항목들뿐만 아니라

다른 항목들도 각각에 대한 스키마를 가지고 있으며, 이 스키마에 의한 명세 인터페이스를 구성한다.



그림 2 Cyber View UI

Cyber View가 필요로 하는 스키마는 그림 3과 같은 계층 구조를 가진다.

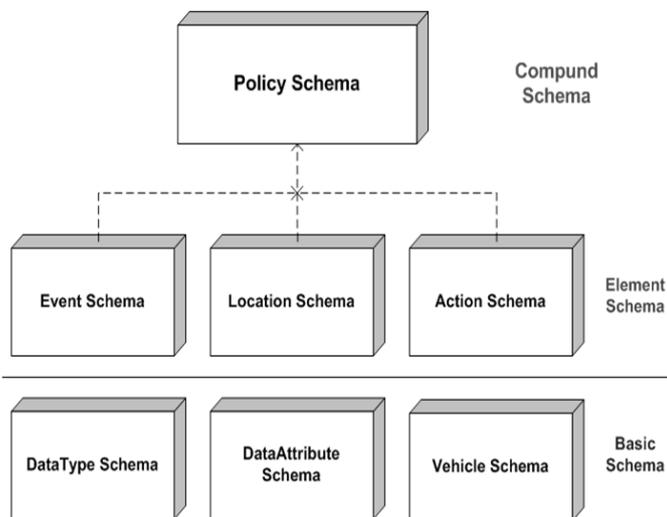


그림 3 Schema Structure

Cyber View는 먼저 정책스키마(Policy Schema)를 참조하여 전체적인 인터페이스를 구성하고 정책스키마에서 정의하고 있는 각 세부 항목들에 대한 스키마 (Event, Location, Action 등)를 다시 참조하여 세부 항목에 대한 인터페이스를 구성하게 된다.

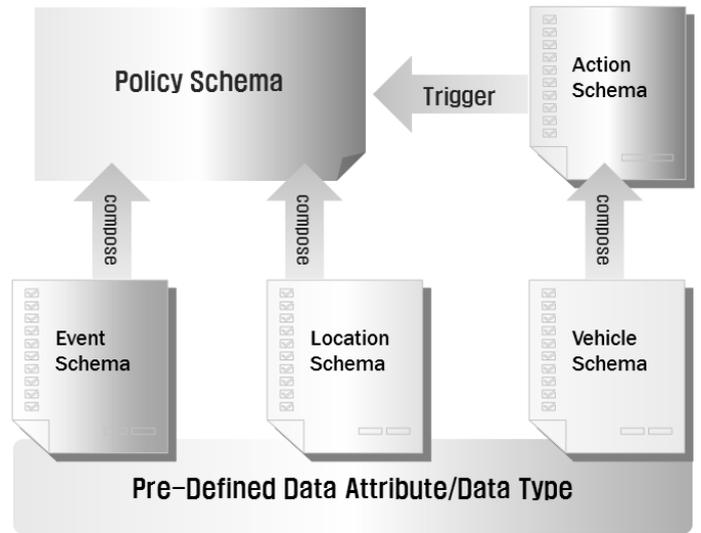


그림 4 Schema 상관관계

그림 4에 도식화한 Event, Location, Action 등의 명세 항목이 존재한다는 사실은 정책스키마를 참조하면 알 수 있고, 여기서 찾아 낸 Event라는 항목에 대해서는 이벤트 스키마(Event Schema)를 참조하면 명세작업이 가능한 항목들과 필요한 데이터에 대한 타입, 입력 방법, 기본 값 등의 정보를 얻을 수 있는 것이다. 물론, 최종적으로 XML 데이터를 생산할 때도 이 스키마의 구조에 의해 결과물을 생성하게 된다. 스키마에 의한 전체적인 명세작업의 과정은 그림 5와 같이 진행된다.

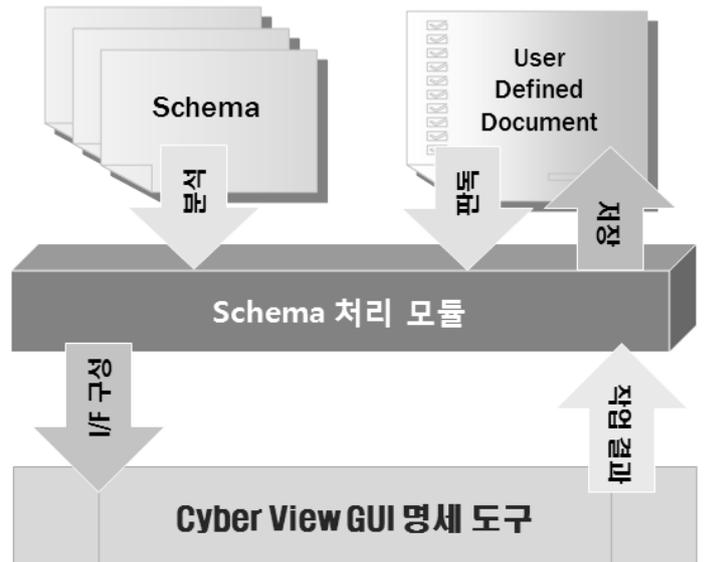


그림 5 Schema/XML 처리과정

스키마는 각 명세항목을 몇 가지로 구분하여 정의하고 있다. simpleType은 그 항목 자체가 하나의 값을 가지게 되는 것을 의미한다. 다음의 simpleType 항목은 문자열 데이터를 의미한다.

```
<xsd:simpleType name="TagInfoType">
  <xsd:restriction base="xsd:string">
    <xsd:length value="14"/>
  </xsd:restriction>
</xsd:simpleType>
```

이런 형태의 simpleType 항목은 사용자가 지정해 주는 문자열 데이터를 가진다. 다음과 같이 여러 개의 항목들 중에서 사용자가 선택한 항목을 가지는 simpleType도 정의할 수 있다.

```
<xsd:simpleType name="ProtocolInfoType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="RSYNC"/>
    <xsd:enumeration value="Local"/>
    <xsd:enumeration value="HTTP"/>
    <xsd:enumeration value="FTP"/>
  </xsd:restriction>
</xsd:simpleType>
```

각 열거(enumeration)항목들이 이 simpleType이 가질 수 있는 선택 가능한 항목들이다.

complexType은 여러 simpleType 혹은 다른 complexType을 포함하는 것을 의미한다. complexType의 예는 다음과 같다.

```
<xsd:complexType name="LANInfoType">
  <xsd:sequence>
    <xsd:element name="Connection"
      type="ConnectionType"/>
    <xsd:element name="IPAddress"
      type="IPAddressType"/>
    <xsd:element name="SubnetMask"
      type="IPAddressType"/>
    <xsd:element name="Gateway"
      type="IPAddressType"/>
    <xsd:element name="DNS"
      type="IPAddressType"/>
  </xsd:sequence>
</xsd:complexType>
```

각 이벤트 항목들이 complexType에 속한 명세항목들이며, name은 각 항목들의 이름, type은 각 항목의 데이터 타입을 의미하는데, 여기서 type은 simpleType이 될 수도 있고 complexType이 될 수도 있다. complexType의 type과 name은 각각 프로그램 언어의 클래스와 인스턴스에 해당하는 것으로 보면 된다.

3.3 XML 문서의 처리와 자료 구조

XML 문서를 처리하기 위해 JDOM1.0[4]을 이용하였다.

전통적으로 XML의 처리에는 다양한 기능을 제공하는 DOM(Document Object Model)과 간편한 처리 방법을 제공하는 SAX 등이 사용되었는데, JDOM(Java DOM)은 DOM과 SAX의 중간 형태를 가지며 Java를 이용하여 XML에 관한 처리를 할 수 있도록 제작된 클래스 라이브러리이다.

Cyber View에서는 명세작업의 최종 결과물인 XML 형식의 문서를 JDOM을 이용하여 생성한다. 앞서 언급한 것과 같이 Cyber View 명세도구의 인터페이스 구성과 XML 문서형식 정의에 필요한 스키마도 XML 형식으로 기술되어 있으므로 특별한 방법이 필요 없이 일반 XML 문서와 같이 JDOM으로 처리할 수 있다.

스키마의 계층 구조를 파악할 수 있도록 하기 위하여 Cyber View에서는 그림 6과 같은 스키마뷰어를 제공한다. 스키마뷰어는 계층구조를 이루고 있는 스키마문서의 구조를 트리형태로 나타내주는 기능을 한다. 기본적으로 스키마와 일반 XML 문서의 구조는 동일하기 때문에 스키마뷰어는 스키마뿐만 아니라 사용자가 정의한 XML 스키마 문서의 내용도 같은 방법으로 표현할 수 있다.



그림 6 Schema Viewer

이와 같이 스키마를 비롯한 XML 데이터의 자료 구조는 계층구조를 가지고 있기 때문에 Cyber View에서는

이 데이터를 Linked-List를 이용한 트리형태의 자료 구조로 처리한다. 스키마의 내용을 바탕으로 명세작업에 필요한 인터페이스를 구성할 때는 스키마 구조를 이루고 있는 스키마를 재귀적인 방법으로 탐색하여 명세문서의 전체적인 틀을 구성한다. 최종 결과물인 XML 데이터를 생성할 때는 스키마를 바탕으로 사용자가 정의한 데이터 내용이 삽입되어 있는 트리형태의 자료구조를 XML 형태로 변환하는데, Parent Node가 Child Node를 가지는 트리구조와 마찬가지로 Parent element가 Child element를 가지는 XML 데이터로 변환해 주게 된다. 그림 7과 그림 8은 같은 내용의 데이터를 트리형태의 자료 구조와 XML 데이터 사이의 관계를 나타낸 것이다.

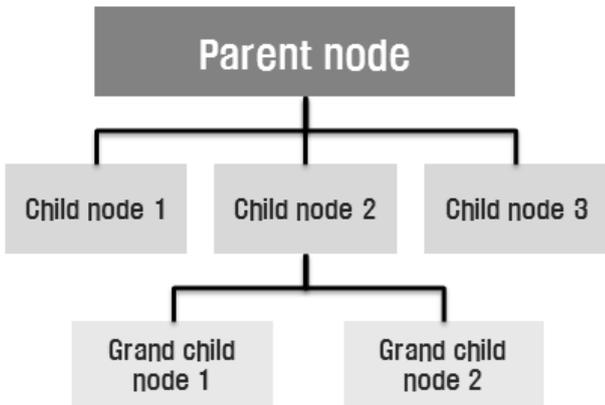


그림 7 Tree 구조의 데이터

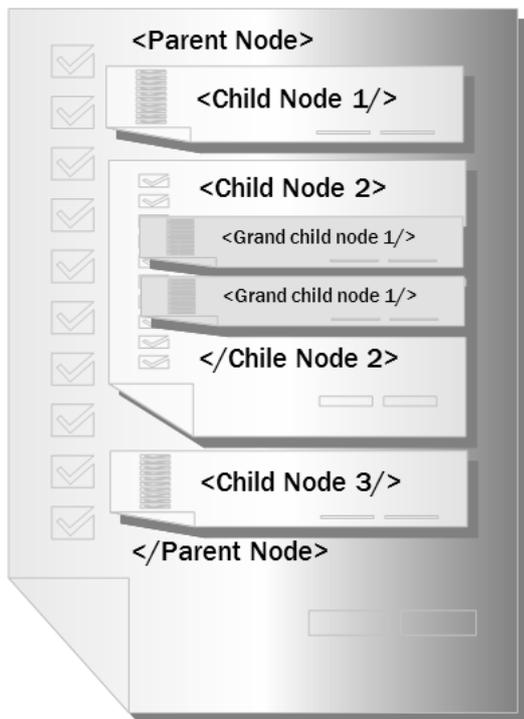


그림 8 XML 데이터 구조

XML 데이터의 각 element는 다음과 같은 형식으로 기술할 수 있다.

```

<element_name attribute_name = value>
  element_text
  <child_element .....>
  .....
  </child_element>
</element_name>
    
```

Element 태그의 가장 첫 번째는 element의 name, 그리고 이어서 element가 필요로 하는 attribute들에 대하여 기술한다. Element 내부에 태그가 아닌 문장들은 이 element의 텍스트라고 한다. Child element들도 이와 같은 방법으로 기술하고, element의 맨 마지막 부분은 </element_name> 태그로 마무리한다. 텍스트나 Child element를 가지지 않는 element는 <element_name> ~ </element_name> 구문 대신 <element_name /> 와 같이 한 줄로 기술할 수 있다. 스키마에서는 각 element의 태그에 주어진 속성 값으로 각 데이터 항목의 name, data type, default value 등을 정의한다.

XML 데이터의 이러한 구조를 위하여 JDOM에서는 XML 데이터를 하나의 Document 객체로 정의한다. 이 Document 객체는 그것이 나타내고 있는 XML 데이터의 최상위 element에 대한 참조를 가지고 있으며, Document 객체의 getRootElement() 메소드 (method)를 이용하여 최상위 element를 구할 수 있다.

최상위 element는 Element 객체로서 표현되는데, 각 Element 객체는 그것이 가지는 child element에 대한 참조를 가지고 있어서 이를 바탕으로 그림 7과 같은 트리 구조 형태의 자료구조를 형성할 수 있다. Cyber View에서는 JDOM이 표현하는 이러한 방식을 바탕으로 Element 객체들을 재귀적인 방법으로 탐색하여 트리구조를 생성하도록 하였다. 또한 Element 객체는 getAttribute()와 getAttributeValue()와 같은 메소드를 제공함으로써 그 element의 속성 값을 얻을 수 있도록 하고 있다.

Cyber View 명세도구에서는 스키마에 대해서는 그 자체의 XML 특유의 트리구조를 곧바로 이용할 수 있도록 하였지만, 새로 생성하게 될 결과물 XML 데이터는 별도의 트리 자료구조를 먼저 생성하여 명세 작업의 처리를 수용하고 최종 결과물을 생성하는 시점에 이 트리 자료구조를 DFS(Depth First Search) 탐색 방법으로 탐색하여 새로운 트리구조의 XML 데이터를 생성하도록 하였다.

3.4 Cyber view 인터페이스

Cyber View는 그림 9와 같은 사용자 인터페이스를 제공하여 명세 작업을 할 수 있도록 한다. 또한 그림 6과

같은 트리구조 형태의 인터페이스를 제공하여 전체적인 XML 데이터의 구조를 한 눈에 살펴볼 수 있는 방법도 제공한다.

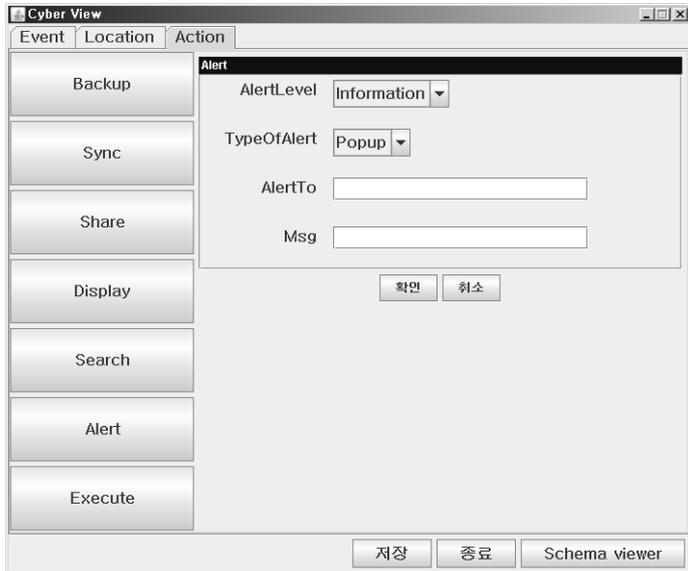


그림 9 Cyber View 명세 인터페이스

4. 결론 및 향후 연구

본 논문의 관련 연구인 UDI 서비스 플랫폼은 유비쿼터스 컴퓨팅환경에서 언제 어디서, 어느 디바이스에서든 동일한 데이터 환경을 제공하고, 사용자의 위치나 문맥, 디바이스의 용량 등을 고려하여 가장 적합한 데이터들이 가장 적합한 디바이스에 적재되어 서비스가 제공되도록 하고, 다수의 디바이스들에 분산된 데이터들이 적절히 보호되도록 하고, 다수의 사용자로 콘텐츠가 분배될 때의 네트워크 성능 이슈를 효과적으로 해결하는 서비스 프로토콜과 플랫폼의 개발을 목표로 한다. 본 논문에서 언급한 Cyber View 명세도구는 UDI 서비스 플랫폼의 최상위 응용계층으로 사용자가 원하는 동작들이 Vehicle 이라는 컴퓨팅 장치를 통해, Vehicle에서 수행될 이벤트를 정의할 수 있는 GUI환경을 제공한다.

현재 개발된 Cyber View는 각 분류별 명세 항목들이 스키마 내용에 의하여 인터페이스를 제공하는 형태로 개발되었다. 명세항목의 대분류인 Event, Location, Action 외에 새로운 분류를 추가할 수 없는 구조로 설계되었다. 그러나 향후 개선된 버전의 Cyber View에서는 모든 명세 작업을 동적 스키마 기반으로 수행하도록 설계하였다. 따라서 스키마 계층의 최상위에 있는 정책스키마의 수정을 통하여 새로운 분류의 명세 항목들을 추가할 수 있게 된다. 또한 명세 대상의 종류나 목적에 맞는 사용자정의 명세 항목들만 XML 결과물로 생성하도록 하여 동적인 명세 구성이 가능하게 설계 개발할 것이다.

참고문헌

- [1] K. Williams 외, Professional XML Databases, 정보문화사, 2001.
- [2] 송정길, XML 프로그래밍, 생능출판사, 2004.
- [3] W3C XML Specification, <http://www.w3.org/XML>
- [4] JDOM Specification, <http://www.jdom.org>
- [5] Open Mobile Alliance, <http://openmobilealliance.org>