

모바일 환경에서 공간데이터 동기화 시스템 설계

이혜진^o 김진석
 한국전자통신연구원 우정기술연구원
 {lhjin^o, kimjs}@etri.re.kr

Design of Spatial Data Synchronization System in Mobile Environment

Hyejin Lee^o Jinsuk Kim
 Postal Technology Research Center, Electronics and Telecommunications Research Institut

요 약

본 연구는 공간데이터 동기화 표준 프로토콜 명세인 SyncML(Synchronization Markup Language)을 이용하여 모바일 장비와 서버사이에 공간데이터를 동기화 구조를 정의하여 상호운용성 확보를 위하여 기존에 사용되어온 GML(Geographic Markup Language) 데이터와의 통합 및 다양한 데이터 소스에 대한 접근 가능성을 제시한다. 뿐만 아니라 Metadata 및 Catalog Service를 이용한 공간데이터 동기화 시스템을 제시함으로써 분산되어 있는 다양한 형태의 공간데이터, 비공간 데이터의 관계성을 고려한 데이터 연동을 가능하게 한다.

1. 서론

무선인터넷이 발전함에 따라 장비의 PDA, 노트북, 핸드폰 등의 장비 등의 활용분야가 다양해짐으로써 서비스의 이동성(Mobility)과 다양한 데이터 소스로부터의 데이터 신뢰성 확보의 중요성이 더욱 강조되었다. 특히 이러한 이동성 확보를 통한 GIS기반의 위치기반서비스(LBS: Location Based Service)는 최근 급부상하고 있는 서비스 분야라 할 수 있겠다.

그러나, 이러한 위치기반서비스는 공간데이터의 지리적 특성으로 인하여 이동성을 확보한 모바일 장비에서 업데이트시 공간데이터의 방대함, 공간데이터 소스의 분산, 이동형 단말기의 저장공간의 부족 등의 원인으로 모바일 클라이언트와 서버사이의 데이터 일관성 유지의 어려움이 발생한다.

본 연구에서는 최근 모바일 산업계 업체들을 중심으로 진행 중인 동기화 표준 프로토콜 명세인 SyncML(Synchronization Markup Language)을 활용하여 동기화 프로토콜 및 메시지를 정의하며, GML(Geographic Markup Language) 및 Catalog Service를 이용한 모바일 환경에서의 공간데이터 동기화 방안을 제시하고자 한다. 뿐만 아니라, 동기화 엔진에서는 Metadata와 Catalog Service를 활용하여 다양한 형태로 분산되어 있는 공간데이터, 비공간 데이터에 대한 관계성을 고려한 데이터 연동이 가능하다.

2. 관련연구

데이터 동기화란 서로 다른 장비 및 응용간에 활용하는 데이터의 일치시키는 중재 동작을 의미한다[1]. 2000년 2월에 IBM, Motorola, Nokia 등의 전세계 모바일 관련업체를 중심으로 모바일 환경에서 데이터 동기화를 위한 표준프로토콜인 SyncML을 정의하였으며, 국내에서는 팬

택앤큐리텔과 같은 단말기 제조사에서 도입하여 그 사용영역이 점차 확장되고 있다[2].

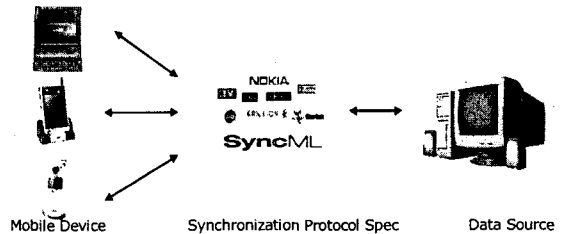


그림1 SyncML을 이용한 동기화 개념

본 연구에서는 동기화에 사용될 공간데이터 메시지 형태를 XML 기반의 공간데이터 표준 명세인 GML을 사용한다. GML 표준화 연구는 2000년 5월 1.0버전을 시작으로 3년동안 많은 노력을 통해 오늘날에 이르렀다. GML 표준화 연구 초기에는 지리정보의 XML 인코딩이라는 동일한 목적하에 OpenGIS Consortium, ISO TC/211, 일본 세 개의 기관에서 서로 다른 이름으로 진행되었다 [3][4][5]. 2001년도에는 통합된 지리정보 인코딩 표준화의 요구로 연합된 형태로 GML 3.0 제정을 추진하여, 2003년 1월 GML 3.0이 공표되었으며, 이와 관련된 연구가 국내에서도 진행되고 있다[6][7].

3. SyncML 동기화 메시지 구조 및 시나리오

동기화 메시지는 동기화 대상들 간의 일종의 약속이며, SyncML을 사용하여 동기화 메시지를 정의할 경우 이것의 결과는 SyncML Representation 프로토콜 명세를 따르는 XML 문서로 나타난다.

application/vnd.syncml+xml,application/vnd.syncml+xml MIME 유형을 가지는 이 문서는 GML과 같이 XML 기반의 데이터를 사용하는 어떠한 Application에서도 인식 및

전송이 가능하므로 그 활용 분야가 점차적으로 확대될 것으로 기대된다. 서버와 클라이언트 사이에 전송될 동기화 메시지 구조는 그림 2와 같다.

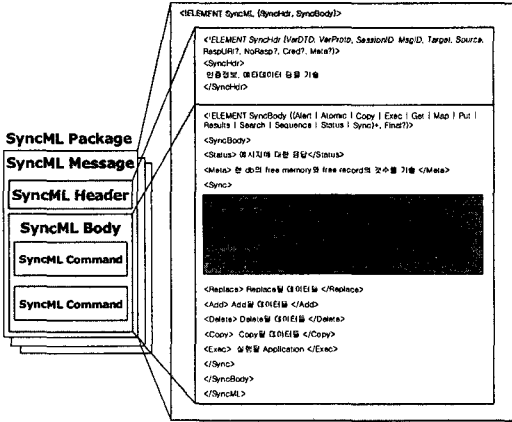


그림 2 SyncML 기반 동기화 메시지 구조

SyncML Command 메시지는 크게 Request와 Response로 나누어진다. Request는 동기화에 필요한 디바이스 정보(Put/Get), Map Table 유지(Map), 동기화 대상 데이터 라우팅(Sync), 모든 Command 처리 결과(Status)에 대한 메시지가 포함된다. 동기화를 위한 주요 SyncML 메시지 구조는 다음과 같다.

① MetaData Namespace 식별

비공간 데이터의 Namespace 정의는 다음과 같다.

```
<Type xmlns='syncml:metinf'>text/x-vcard</Type>
<Type xmlns='syncml:metinf'>application/vnd.syncml-devinf+xml</Type>
```

공간데이터의 Namespace 정의는 다음과 같이 별도의 Namespace를 정의하여 사용할 수도 있으며, 표준화된 Namespace의 정의로만으로도 데이터 연동이 가능하다.

```
<schema targetNamespace="http://www.myorg.com/parcels"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml"
xmlns:gp="http://www.myorg.com/geodeticPoints"
xmlns:pcl="http://www.myorg.com/parcels"
xmlns:iso19115="http://www.iso211.org/iso19115/"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified"
version="2003-07-20">
<include schemaLocation="Buildings.xsd"/>
<import namespace="http://www.myorg.com/geodeticPoints"
schemaLocation="GeodeticPoints.xsd"/>
<import namespace="http://www.opengis.net/gml"
schemaLocation="base/gml3.xsd"/>

<Type xmlns='gml:featureinfo'>text/xml</Type>
<Type xmlns='csw:QueryType'>text/xml</Type>
```

② Insert

```
<xsd:complexType name="InsertType">
<xsd:sequence>
<xsd:element ref="csw:AbstractRecord" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="handle" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

③ Update

```
<xsd:complexType name="UpdateType" id="UpdateType">
<xsd:sequence>
<xsd:choice>
<xsd:element ref="csw:AbstractRecord"/>
<xsd:element ref="csw:RecordProperty" maxOccurs="unbounded"/>
</xsd:choice>
<xsd:element ref="csw:Constraint" minOccurs="1" maxOccurs="1"/>
</xsd:sequence>
<xsd:attribute name="typeName" type="xsd:anyURI" use="optional"/>
<xsd:attribute name="handle" type="xsd:string" use="optional"/>
</xsd:complexType>
<xsd:element name="RecordProperty"
type="csw:RecordPropertyType"/>
<xsd:complexType name="RecordPropertyType">
<xsd:sequence>
<xsd:element name="Name" type="xsd:string"/>
<xsd:element name="Value" type="xsd:anyType" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
```

④ Delete

```
<xsd:complexType name="DeleteElementType"
id="DeleteElementType">
<xsd:sequence>
<xsd:element ref="csw:Constraint" minOccurs="1" maxOccurs="1"/>
</xsd:sequence>
<xsd:attribute name="typeName" type="xsd:anyURI" use="optional"/>
<xsd:attribute name="handle" type="xsd:string" use="optional"/>
</xsd:complexType>
```

⑤ Search

Search는 특정 조건의 데이터만을 동기화할 수 있게 해 주는 기능을 제공하며 Catalog Service의 Filter 메커니즘을 사용한다.

```
<Meta><Type xmlns='CSW:Filter'>text/xml</Type></Meta>
<Filter>
<ClassifiedAs>
<EntryPoint>RECORD</EntryPoint>
<KeyName>/GeoClass/Continent/Country/State</KeyName>
<KeyValue>/GeoClass/NorthAmerica/%/Ontario</KeyValue>
</ClassifiedAs>
</Filter>
```

4가지 정도의 동기화 주요 메시지 구조 정의를 살펴보았다. 이 이외에도 SyncML 명세 포함되어 있는 충돌 처리 (Conflict Resolution), 인증, Busy Signal에 대한 부분과 이에 따른 메시지를 포함해야 한다. 그러나 실제로는 이러한 메시지 정의보다는 충돌 처리의 정책이 필요하며, 일반적인 충돌처리 정책이 SyncML을 사용하였을 때도 유효한가에 대해서는 실제적인 구현을 통한 검증이 필요할 것으로 보인다.

SyncML 명세에는 모두 7 가지 동기화 유형을 정의하고 있다. 가장 기본이 되는 동기화 유형은 Two-Way 방식이라고 할 수 있으며, Server Alerted 방식을 제외하고는 모두 서버가 동기화의 주체가 된다. 서버가 동기화의 주체가 되기 위해서는 Push 모델을 사용한 전송이 필요하며, 본 연구에서도 그림 3과 같이 Two-Way 동기화 방식에 따른 시나리오를 적용한다. Two-Way 동기화 방식에서는 일반적으로 클라이언트가 먼저 동기화를 시작하며, 각 메시지에 대한 응답을 Status 메시지 형태로 보내게 된다.

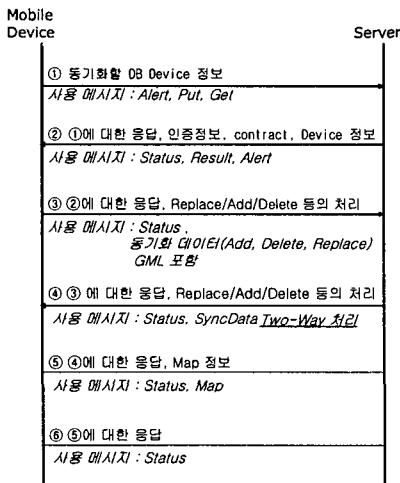


그림 3 SyncML을 이용한 동기화 시나리오

5. SyncML기반 동기화 시스템 구조

본 연구에서는 분산된 데이터의 처리를 위하여 OpenGIS Catalog Service를 이용한다. OpenGIS Catalog Service는 공간, 비공간데이터에 대한 분산 처리를 지원하며, 각 세부 서비스는 그림 4와 같다.

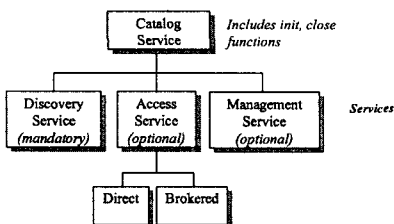


그림 4. OpenGIS Catalog Service

OpenGIS Catalog Service는 분산 검색을 위한 Discovery Service, 분산 데이터에 대한 접근을 위한 Access Service, 분산 데이터 관리를 위한 Management Service의 3가지를 제공한다[8].

동기화 시스템은 그림 5와 같이 Application, Service, Data의 3-tier로 구성되며, 데이터 소스로부터 확장성 및 상호운용성을 지원한다. 동기화 엔진에서는 공간데이터, 비공간 데이터에 대한 관계성을 고려한 연동을 위하여 Metadata를 이용한 동기화 부분을 설계하였으며, 분산된 데이터의 표준화된 인터페이스 제공 및 소스 관리를 위한 Catalog Service 엔진이 포함되어 있다[9]. SyncML 메시지 내부에서 공간데이터 표현을 위해 사용되는 GML은 공간데이터와 비공간데이터의 저장과 전송을 위한 XML 인코딩 표준이므로, 데이터 공유를 위한 확장성 및 상호운용성을 확보할 수 있다. 그림 5는 SyncML을 이용한 동기화 시스템 구조를 나타낸다.

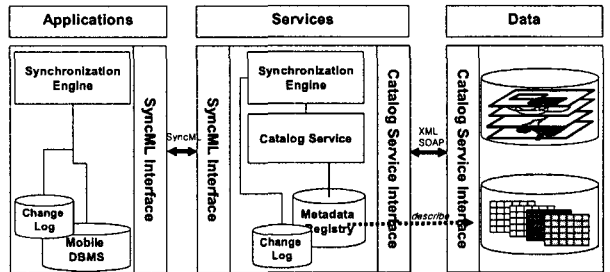


그림 5. SyncML을 이용한 동기화 시스템 구조

6. 결론

본 연구는 SyncML, GML 등의 표준을 기반으로 한 동기화 방안을 제시함으로써 다양한 환경으로 확장하여 상호운용성을 확보할 수 있으며, 분산된 공간데이터와 비공간 데이터의 관계성을 기반으로 동기화함으로써 다양한 데이터 소스에 대한 개방된 접근을 가능하게 한다. 향후 공간데이터 동기화를 위한 충돌 처리 방안 및 GML 압축방안을 적용한 모바일 장비와의 동기화를 확보한 시스템을 구현할 계획이다.

[참고문헌]

- [1]이지연, 최훈, "SyncML 서버 응용 개발을 위한 통합 개발 환경", 정보과학회논문지, 제 11권, 제 1호, p.37, 2004년
- [2]SyncML.org, SyncML Sync Protocol, version 1.1, 2002
- [3]OpenGIS Consortium, Inc., Geography Markup Language(GML) 2.1.2, 2002
- [4]ISO/TC 211, <http://www.isotc211.org/>
- [5]G-XML, <http://www.dpc.or.jp/>
- [6]OpenGIS Consortium, Inc., Geography Markup Language(GML) 3, 2003
- [7]이혜진 외, "GML을 적용한 이동체 관리 시스템의 설계 및 구현", 2003년 정보과학회 가을 학술발표논문집, 제 30권, 제2호, 2003.10
- [8]OpenGIS Consortium, Inc., OpenGIS Catalogue Services Specification, 2002
- [9]ISO/TC 211, ISO 19115 Metadata, 2004