

SyncML의 디바이스 관리 표준화 동향

The Survey of SyncML Device Management Specification

박지은(J.E. Park)

휴대클라이언트연구팀 선임연구원

박준석(J.S. Park)

휴대클라이언트연구팀 선임연구원

한동원(D.W. Han)

휴대클라이언트연구팀 책임연구원, 팀장

현재, 이동 디바이스를 조작하고 관리하는 방법은 개별 기기마다 공급 업체에서 제공하는 고유 방식에 의존하고 있다. 따라서, 일관된 방법으로 원격에서 사용자를 대신하여 디바이스의 하드웨어 및 소프트웨어 상태를 점검하거나 응용 소프트웨어를 설치하는 서비스를 현재의 환경 하에서는 제공할 수 없다. 이는 일반 사용자 뿐만 아니라 제조업체와 서비스 업체에서도 디바이스 관리를 위하여 많은 비용을 지출하게 한다. 이러한 문제를 해결하기 위하여 SyncML은 표준화된 방법으로 원격에서 이동 디바이스를 관리할 수 있는 Device Management 스펙 1.1을 발표하였다. 본 고에서는 SyncML 디바이스 관리의 기본 개념과 스펙의 구성 및 주요 스펙을 분석하고자 한다.

1. 서론

네트워크 기능을 탑재한 이동 디바이스의 사용이 보편화되고, 서비스 영역이 확대됨에 따라 이들의 하드웨어 및 소프트웨어 기능이 점점 복잡하고, 다양해지고 있다[1],[2]. 이러한 기능들을 비전문가인 일반인들이 쉽게 이용하기에는 어려운 점이 있다. 예를 들어, 새롭게 개발된 소프트웨어를 설치하고자 하는 경우, 대부분의 일반 사용자들은 자신의 디바이스를 취급하는 서비스 업체를 직접 방문하여 원하는 소프트웨어의 설치를 의뢰한다. 뿐만 아니라, 디바이스에 이상이 발견된 경우에도 일반 사용자들은 원인 파악을 위해 직접 해당 서비스 업체를 방문하여야 하며, 수리를 의뢰하여야 한다. 이러한 문제의 근본 원인은 이동 디바이스를 조작하고 관리하는 방법이 개별 기기마다 공급 업체에서 제공하는 고유 방식에 의존하고 있기 때문이다. 따라서 일관된 방법으로 원격에서 사용자를 대신하여 디바이스의 하

드웨어 및 소프트웨어 상태를 점검하거나 응용 소프트웨어를 설치하는 서비스를 현재 환경 하에서는 제공할 수 없다. 이는 일반 사용자 뿐만 아니라 제조업체와 서비스 업체에서도 디바이스 관리를 위하여 많은 비용을 지출하게 한다.

SyncML(Synchronization Markup Language)은 2002년 2월에 독립적이고, 표준화된 방법으로 디바이스를 관리함으로써 원격 관리 서비스가 가능한 디바이스 관리(device management) 스펙 1.1을 발표하였다.

SyncML은 그 동안 주소록, 일정 관리 등과 같은 PIMS(Personal Information Management System) 데이터를 표준화 방법으로 동기화하기 위하여 XML 기반의 데이터 동기화(data synchronization) 스펙을 발표하였다[3]. 현재, SyncML은 두 영역 즉, 데이터 동기화 부분과 디바이스 관리 부분으로 나누어 활동하고 있다. SyncML의 디바이스 관리 스펙은 대부분 데이터 동기화 스펙의 DTD(Docu-

ment Type Definition)를 기반으로 하고 있으며, 특히 디바이스 자체의 하드웨어 및 소프트웨어적인 특성을 보다 상세하게 기술하기 위한 엘리먼트들이 추가 정의되었다.

본 고에서는 SyncML의 디바이스 관리에 대한 기본 개념과 발표된 스펙 1.1을 분석하고자 한다.

II. SyncML 디바이스 관리의 개요

1. 기본 개념

디바이스 관리란 용어는 디바이스 서비스 업체에서 사용자를 대신하여 탑재된 소프트웨어를 새롭게 구성(configuration)하거나 네트워크나 기타 내부 기기들의 설정 값들을 변경하는 등과 같은 일들을 말한다. 이와 같은 의미를 기반으로 디바이스 관리 영역은 다음과 같이 세 부분으로 나눈다[2].

- 프로토콜
디바이스 관리 서비스를 제공하는 서버와 클라이언트 사이에 사용하는 프로토콜
- 데이터 모델
브라우저나 메일 응용에서 사용자 환경 설정 기능 등과 같이 원격에서 조작이 가능한 데이터 정의
- 방식
디바이스의 설정 값 혹은 디바이스 내의 특별한 객체를 변경할 수 있는 권한 결정

이상의 영역을 다루는 SyncML 디바이스 관리 스펙(이하 SyncML DM)은 기존에 발표한 데이터 동기화 스펙에서 정의한 핵심 DTD를 재사용하고 있으며, 데이터 동기화 방식 중 하나로 정의한 단방향 서버 요구 동기화(one-way server-alerted sync)와 유사한 형태로 SyncML DM 프로토콜을 디자인하였다. SyncML DM 스펙의 응용 영역은 다음과 같다.

- 디바이스의 구성 관리
디바이스 동작에 필요한 파라미터 값을 변경하

거나 검색

- 소프트웨어 관리
디바이스에 설치된 소프트웨어의 설치, 삭제, 갱신
- 저장소 관리
디바이스 내의 하드웨어의 구성 정보나 설치된 소프트웨어의 목록 정보 관리
- 고장 진단
디바이스가 진단을 요구하는 메시지를 서버로부터 받거나 디바이스 내에 설치된 고장 진단 모듈을 서버에서 원격 호출

2. 스펙의 구성

SyncML DM은 다음과 같이 총 7개 스펙으로 구성된다.

- SyncML Device Management Bootstrap
- SyncML Device Management Security
- SyncML Notification Initiated Session
- SyncML Device Management Tree and Description
- SyncML Device Management Protocol
- SyncML Representation Protocol, Device Management Usage
- SyncML Device Management Standardized Objects

Device Management Bootstrap 스펙[4]은 클라이언트 디바이스가 서버와 디바이스 관리 세션을 시작할 수 있는 상태가 되기까지의 과정에 대해 기술한다. 이는 두 형태, 즉 주문형(customized bootstrap)과 서버 초기화(server initiated bootstrap) 형태로 나눈다. 주문형은 제조업체에서 부트스트랩 과정에 필요한 모든 정보들을 디바이스에 적재한 형태로 사용자에게 공급함으로써 사용자의 추가 작업 없이 세션을 시작할 수 있는 경우이다. 후자는 사용자가 먼저 단말의 주소를 특정 서버에 알린 후, 이 서버로부터 부트스트랩에 필요한 정보들을 획득함으로써 세션을 시작하는 경우이다.

Device Management Security 스펙[5]은 서버와 클라이언트 디바이스의 상호 인증 과정과 방법을 기술한다. Notification Initiated Session 스펙[6]은 디바이스 관리 서비스를 제공하기 위하여 서버가 클라이언트 디바이스에 세션 시작을 요구하는 과정과 이때, 서버가 송신하는 메시지를 정의한다.

Device Management Tree and Description 스펙[7]은 디바이스 관리 서비스의 대상이 되는 객체(management objects)들을 트리 구조로 구성한 Management Tree를 정의한다. 이때, 각 객체는 URI(Uniform Resource Identifier)를 통해 식별되며, 각각의 속성들을 가진다. 디바이스 관리 서비스를 제공하는 서버는 디바이스 관리 세션을 통해 클라이언트 디바이스 내의 관리 객체들의 트리 구조를 탐색하거나, 새로운 객체를 추가 혹은 변경할 수 있다.

Representation Protocol 스펙, Device Management Protocol 스펙, Device Management Standardized Objects 스펙은 디바이스 관리 세션 과정의 메시지 흐름과 메시지 표현 방법, 실제 관리할 객체들을 정의하는 SyncML의 주요 스펙이다. 이 스펙들은 다음 장에서 상세히 분석하고자 한다.

III. SyncML 디바이스 관리의 주요 스펙 분석

1. 표현 프로토콜 스펙

표현 프로토콜은 디바이스 관리 서비스를 실행하기 위해 서버와 클라이언트가 상호 교환하는 XML 메시지(well-defined)를 정의한 문서이다[8]. 이는 기존에 발표한 SyncML의 데이터 동기화 표현 스펙 1.1[3]을 기반으로 하고 있다.

본 고에서는 SyncML의 엘리먼트들을 주 사용(common use), 메시지 컨테이너(message container), 데이터 기술(data description), 프로토콜 관리(protocol management), 프로토콜 명령어(protocol command)로 나누어 기술한다.

가. 주 사용 엘리먼트

데이터 동기화 표준 스펙 1.1에서 정의한 주 사용 엘리먼트들 중 SyncML DM에서 사용하는 엘리먼트들은 다음과 같이 요약할 수 있다.

사용되지 않음	Archive, Lang, NoResp, NoResults, NumberOfChanges, SoftDel
동일한 의미로 사용됨	Chal, Cmd, CmdID, CmdRef, Cred, Final, LocURI, MoreData, MsgID, MsgRef, RespURI, SessionID, Source, SourceRef, Target, TargetRef, VerDTD, VerProto
추가/변경됨	LocName

Chal은 인증을 요구하기 위한 엘리먼트이며, Cmd는 명령어, CmdID는 Cmd에 대한 식별자, CmdRef는 수신한 Cmd를 나타내는 식별자이다. Cred는 인증 정보를 보낼 때 사용하며, Final은 메시지의 종료를 알리며, LocURI는 관리 객체의 주소를 나타낸다. SessionID는 세션을 구별하기 위한 식별자, Source는 송신측 객체의 라우팅 정보, SourceRef는 Source를 지시하는 식별자, Target은 수신측 객체의 라우팅 정보, TargetRef는 Target에 대한 지시자이다. VerDTD는 메시지에서 사용하는 DTD 버전 정보, VerProto는 프로토콜 버전 정보를 나타낸다. LocName은 데이터 동기화 서비스에도 이용되나, DM에서 이용될 때는 반드시 MD5로 인증을 하는 과정에서 사용자 ID를 보낼 때 사용되어야 한다는 제약 조건을 가진다.

나. 메시지 컨테이너 엘리먼트

SyncML의 메시지 컨테이너로 사용되는 엘리먼트들은 SyncML, SyncHdr, SyncBody가 있다. SyncML은 최상위 엘리먼트로서 MIME 콘텐츠 타입, 즉 application/vnd.syncml.dm+xml 혹은 application/vnd.syncml.dm+xml의 값을 나타낸다. SyncHdr는 DM 서비스를 위해 필요한 인증 정보, 프로토콜 유형 등을 명세하기 위한 엘리먼트로서 데이터 동기화 서비스와 동일한 의미로 사용된다. SyncBody는 실제 수행이 필요한 명령어들을 담고

있는 엘리먼트이다.

다. 자료 기술 엘리먼트

자료 기술에 사용되는 엘리먼트들은 Data, Item, Meta가 있으며, 데이터 동기화 스펙에서 정의한 의미와 동일하게 DM에서 사용된다. Data는 SyncML 메시지로 전달되어야 할 실제 값을 나타내며, Item은 Data를 포함하는 상위 엘리먼트이다. Meta는 Data의 기술 형식 즉, 메타 정보를 나타낸다.

라. 프로토콜 관리 엘리먼트

프로토콜 관리 엘리먼트는 송신한 명령어의 실행 여부를 나타내기 위한 Status가 있다. DM에서 이의 의미는 데이터 동기화 스펙과 동일하게 사용된다.

마. 프로토콜 명령 엘리먼트

SyncML 고유 프로토콜에서 데이터 동기화 혹은 디바이스 관리 서비스를 위해 정의하여 사용하는 엘리먼트들로 데이터 동기화와 비교하여 다음과 같이 요약할 수 있다.

데이터 동기화 프로토콜에서 서버와 클라이언트간 동일 데이터의 매핑 정보를 나타내기 위한 Map, MapItem, 정보 전달/검색/동기화를 위한 Put, Search, Sync는 SyncML DM에서 사용되지 않는다. Results 즉, 수신측 정보를 요구하는 Get 명령의 수행 결과를 송신하기 위한 명령어는 동일한 의미로 DM에서 사용한다.

사용되지 않음	Map, MapItem, Put, Search, Sync
동일한 의미로 사용됨	Results
변경 혹은 추가됨	Add, Alert, Atomic, Copy, Delete, Exec, Get, Replace, Sequence

Add는 새로운 관리 대상, 즉 SyncML DM 서비스를 적용할 객체를 추가하기 위한 엘리먼트이다. Alert는 알림(notification) 메시지를 표현하기 위해 사용하는 엘리먼트로서 SyncML DM 세션을 시작

혹은 종료하거나 사용자와 상호 작용할 형태 즉, 전달할 메시지를 단순히 표시만 할 것인지 혹은 사용자의 텍스트 입력을 요구할지 등의 형태를 명시한다.

Atomic은 여러 개의 프로토콜 명령어들을 포함하고, 이들은 명시된 순서대로 실행되어야 하며, 이의 수행 결과는 하나의 Status로 응답한다. Copy, Delete, Replace는 관리 대상 객체의 복사, 삭제, 변경을 위한 엘리먼트이다. Get은 SyncML DM에서 정의한 객체의 정보를 얻기 위한 엘리먼트이며, Sequence는 여러 개의 프로토콜 명령어들을 포함하는데 이때 각 명령어들은 명시한 순서대로 실행되어야 함을 나타낸다.

2. 디바이스 관리 프로토콜 스펙

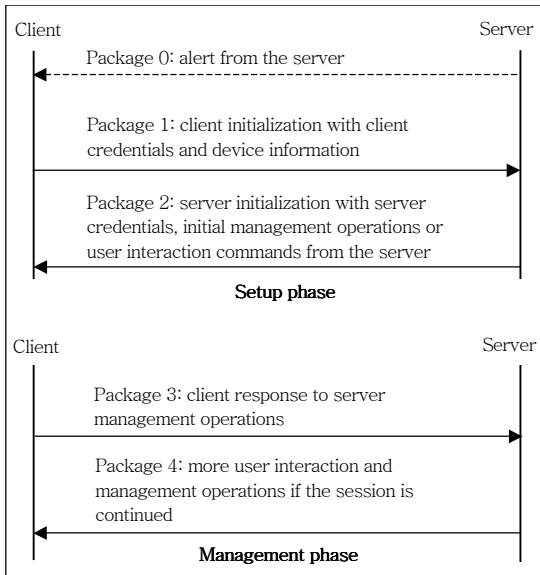
디바이스 관리 프로토콜(이하 DM 프로토콜 명칭) 스펙은 데이터 동기화 프로토콜과 유사한 형태로 실제 디바이스를 관리하기 위해 서버와 클라이언트가 상호 송수신하는 패키지들의 흐름을 정의한다[9].

DM 프로토콜은 (그림 1)과 같이 두 단계 즉, 인증과 디바이스 자체 정보를 교환하기 위한 설정(setup) 단계와 관리(management) 단계로 구성된다.

설정 세션은 서버로부터 Alert 명령을 받는 패키지 #0에서 시작할 수 있지만, 이 부분은 응용 환경에 따라 의존도가 있으므로 DM 프로토콜 스펙에서는 기술하지 않는다.¹⁾

패키지 #1은 Alert 명령을 받는 클라이언트가 자신의 디바이스 정보와 인증 정보를 서버로 전달하기 위한 명령어들로 구성된다. 서버는 이 패키지를 수신하여 DM 서비스를 수행할 적절한 대상인지를 인증하며, 또한 수신한 디바이스 정보를 분석하여 해당 디바이스의 하드웨어 및 소프트웨어 사양 정보를 획득한다. 패키지 #2는 서버가 자신의 인증 정보와 현 프로토콜에서 정의한 명령어 혹은 사용자와 상호 작용하는 형태를 클라이언트에게 알린다.

1) Trigger 관련 내용은 SyncML Notification Initiated Session에 자세히 기술된다.



(그림 1) SyncML DM 구성

예를 들어, PDA에 자동으로 바이러스 차단 소프트웨어를 관리하는 경우, 서버는 다음과 같은 엘리먼트들로 구성된 메시지를 클라이언트로 송신한다.

```

...
<Alert>
  <CmdID>2</CmdID>
  <Data>1100</Data> <!-- User displayable notification -->
  <Item>
    <Data>Your antivirus software is being updated</Data>
  </Item>
</Alert>
<Get>
  <CmdID>4</CmdID>
  <Item>
    <Target><LocURI>./antivirus_data/version</LocURI></Target>
  </Item>
</Get>
...
    
```

Alert는 사용자에게 바이러스 차단 소프트웨어가 갱신되고 있음을 알리는 메시지를 디스플레이 하기 위해 1100이라는 데이터와 메시지 내용을 전달한다. 또한, 사용자가 현재 보유하고 있는 소프트웨어의 버전 정보를 알기 위하여 Get 엘리먼트를 보낸다.

패키지 #3은 클라이언트가 서버로 패키지 #2의 명령어들을 수행한 결과를 전달한다. 이 결과에 따라, 예를 들어 바이러스 차단 소프트웨어 관리 서비스인 경우, 클라이언트의 소프트웨어 버전 정보가 최신이면 패키지 #4는 서비스 종료를 위한 명령을 클라이언트로 보내게 된다. 만약 버전 갱신이 필요한 경우, 서버는 패키지 #4를 통해 해당 소프트웨어를 클라이언트로 전달하게 되고, 다시 패키지 #3를 통해 수행결과를 전달 받게 된다.

DM 프로토콜의 관리 단계는 실제 디바이스를 관리하기 위한 명령어의 수행을 요구하고 그 결과를 알려주는 패키지들로 구성되며, 또한 이러한 패키지 내의 내용에 따라 반복적으로 수행될 수 있다.

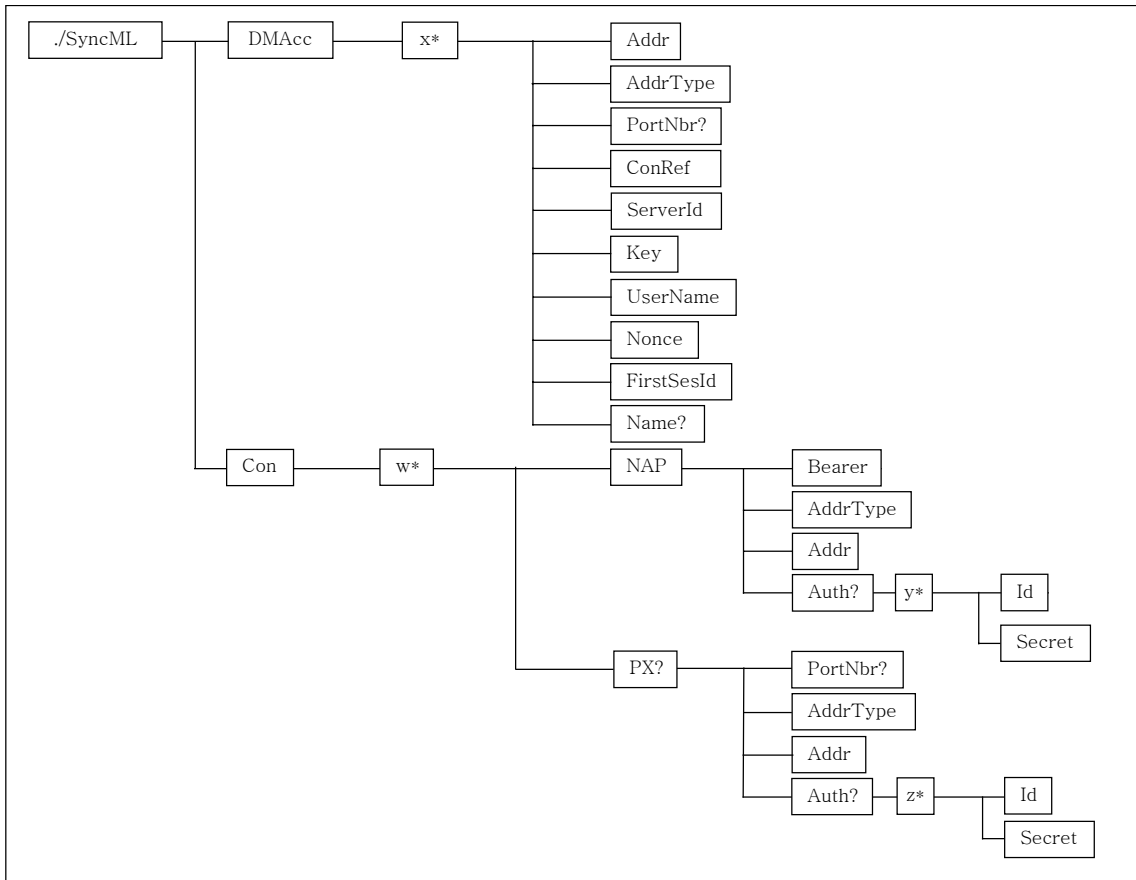
3. 디바이스 관리 표준 객체 스펙

디바이스 관리 표준 객체 스펙은 SyncML DM 명령어를 수행할 대상인 디바이스 객체들 중 일부를 필수 객체로 지정하고 이를 정의한다[10]. SyncML DM의 필수 객체는 SyncML DM, DevInfo, Dev-Detail이다.

가. SyncML DM 객체

SyncML DM 객체는 사용자 계정을 나타내는 DMAcc 객체와 네트워크와 관련된 Con 객체로 구성된다.

DMAcc/x는 부트스트랩 시점에 서버가 이름을 부여하는 컨테이너 객체로서 하나 이상의 DM Account를 나타낸다. DMAcc/x/Addr는 주소를 저장하며, DMAcc/x/AddrType는 DMAcc/x/Addr의 포맷과 해석 방법을 명세한다. DMAcc/x/PortNbr는 현재 이용할 베어러의 포트 번호를 지정한다. DMAcc/x/ConRef는 Con 객체의 참조 값을 나타



(그림 2) SyncML DM 객체

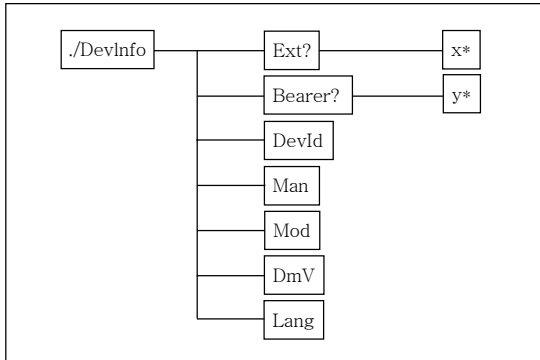
낸다. DMAcc/x/ServerId는 서버 Id, DMAcc/x/UserName은 인증에 이용할 사용자 이름, DMAcc/x/Key는 인증에 이용하게 될 키 정보를 나타낸다. DMAcc/x/Nonce는 인증에 사용할 Nonce값을 저장한다. DMAcc/x/FirstSesId는 클라이언트가 사용할 첫번째 세션 ID를 나타내며, DMAcc/x/Name은 사용자의 이름을 나타낸다.

Con/w는 하나 이상의 Connectivity 객체를 위한 컨테이너 객체이다. 이 객체의 이름은 서버에 의해 지정되며, DMAcc/x/ConRef 객체의 값과 동일하여야 한다. Con/w/NAP/Bearer는 베어러 타입을 명세한다. Con/w/NAP/AddrType는 Address 유형을 명세한다. Con/w/NAP/Addr는 원격 엔티티와 통신하기 위해 필요한 정보 즉, 접근할 라우터의 폰 번호를 나타낸다. Con/w/NAP/Auth/y는 인증 방법

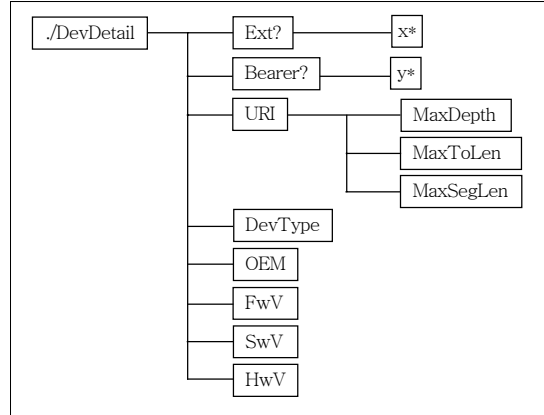
을 명세하기 위한 객체로 인증 파라미터, 즉 Id와 Secret을 연관짓기 위해 사용한다. Con/w/NAP/Auth/y/Id는 인증 ID 즉, 사용자 이름을 명세한다. Con/w/NAP/Auth/y/Secret password를 명세한다. Con/w/PX/PortNbr는 사용할 포트 번호를 명세하며, Con/w/PX/AddrType은 Con/w/PX/Addr 객체의 유형을 명세한다. Con/w/PX/Addr는 주소를 명세하며, Con/w/PX/Auth/z는 PX의 인증 방법을 명세한다. Con/w/PX/Auth/z/Id는 사용자 이름을 명세하며, Con/w/PX/Auth/z/Secret는 패스워드를 명세한다(그림 2) 참조.

나. DevInfo 객체

DevInfo는 SyncML DM 세션 중에 서버로 보내야 하는 디바이스 정보를 정의하기 위한 객체로 다



(그림 3) DevInfo 객체



(그림 4) DevDetail 객체

음과 같이 구성된다.

Ext는 서브 트리를 확장할 수 있다는 것을 나타내며, Bearer는 네트워크 베어러와 관련된 아이템들을 저장하기 위한 객체이다. DevId는 디바이스를 유일하게 식별하기 위한 식별자이며, Man은 디바이스의 제조자, Mod는 디바이스 모델 ID를 나타낸다. DmV는 디바이스가 지원하는 DM 버전 정보를 나타내며, Lang은 디바이스에 나타나는 사용자 인터페이스에서 사용하는 언어를 나타내는 객체이다 ((그림 3) 참조).

다. DevDetail 객체

DevDetail 객체는 디바이스의 상세 정보를 정의하는 객체로서, 이 객체는 DM 세션중에 서버로 보내지지는 않는다. 다만, 서버가 전달한 명령어에 의해 관리되는 클라이언트 객체이다. DevDetail은 다음과 같이 구성된다.

Bearer는 네트워크 베어러와 관련된 아이템들을 저장하며, URI/MaxDepth는 디바이스가 지원하는 관리 트리의 깊이를 나타낸다. URI/MaxToLen는 관리 객체를 지정하기 위해 사용하는 URI 최대 길이이며, URI/MaxSegLen는 URI 단위 길이를 나타낸다. DevType은 디바이스의 유형을 지정한다. OEM은 주문자 회사를 명시하며, FwV는 디바이스의 펌웨어 버전을 나타낸다. SwV는 디바이스의 소프트웨어 버전을 나타내며, HwV는 하드웨어 버전이다. HwV, SwV, FwV, Man, Mod, OEM의 결합

은 특정 소프트웨어를 지정할 수 있는 유일한 식별자가 된다((그림 4) 참조).

IV. 결론

본 고에서는 SyncML의 새로운 기술 영역인 디바이스 관리의 표준화 동향에 대해 기술하기 위하여 SyncML에서 의미하는 디바이스 관리의 기본 개념과 이를 구성하는 스펙에 대해 설명하였다. 또한, 디바이스 관리 서비스를 위한 주요 스펙의 내용을 분석하였다.

디바이스 관리를 위하여 이러한 인프라가 구축이 된다면, 일반 사용자들은 점점 많은 영역으로 확대되어 복잡하게 제공되는 서비스들을 자신이 직접 조작하지 않고도 자동화된 방법으로 보다 편리하게 제공받을 수 있다. 또한, 디바이스 제조업체에서도 호환성 있는 디바이스를 개발함으로써 개별 디바이스마다 별도로 운영해온 관리 서비스 구축에 따른 비용을 최소화 할 수 있다.

지금까지 공개된 SyncML 디바이스 관리 서버와 클라이언트 제품은 없다. 그러나, 현재, SyncML 데이터 동기화 스펙 인증을 받은 제품들은 80여 개가 있으며, 이들이 이용하는 SyncML 프레임워크의 많은 부분들이 디바이스 관리를 위해 재 사용될 수 있다. 따라서 SyncML 디바이스 관리 스펙을 따르는 제품이 시장에 나오는 데 걸리는 시간은 데이터 동

기화 스펙보다 더 이를 것으로 생각된다.

향후 디바이스 관리 스펙을 기반으로 하는 다양한 디바이스들과 응용 서비스들이 개발되기를 기대해 본다.

참 고 문 헌

- [1] Starfish co., "Device Management Overview," www.starfish.com/solutions/datasheets/ds_dm.pdf.
- [2] SyncML Initiative Ltd., "The Business Case for Device Management," White Paper, www.syncml.org, 13 Nov. 2001.
- [3] SyncML Initiative Ltd., "Building an Industry-Wide Mobile Data Synchronization Protocol," White Paper, www.syncml.org, 2000.
- [4] SyncML Device Management Bootstrap, Version 1.1, www.syncml.org, 2002.
- [5] SyncML Device Management Security, Version 1.1, www.syncml.org, 2002.
- [6] SyncML Notification Initiated Session, Version 1.1, www.syncml.org, 2002.
- [7] SyncML Device Management Tree and Description, Version 1.1, www.syncml.org, 2002.
- [8] SyncML Representation Protocol, Device Management Usage, Version 1.1, www.syncml.org, 2002.
- [9] SyncML Device Management Protocol, Version 1.1, www.syncml.org, 2002.
- [10] SyncML Device Management Standardised Objects, Version 1.1, www.syncml.org, 2002.