이동 단말기를 위한 OMA DM 기반 관리 에이전트 설계 및 구현

신 재 영[†] · 정 용 우[††] · 고 광 선[†††] · 엄 영 익[††††]

요 약

이동 단말기들은 이동통신 서비스의 발달과 더불어 다양한 멀티미디어 기능을 포함하는 컨버전스 단말기로 진화하고 있으며, 사용자들로 하여금 다양한 서비스를 원활히 이용할 수 있도록 고성능화되어 가고 있다. 이러한 이동 단말기의 고성능화는 기기 자체에 대한 복잡한 설정을 요구하며, 잠재적인 H/W 또는 S/W 오류 발생 가능성과 설치 프로그램 간 충돌로 인하여 시스템 성능 저하 등의 문제를 가질 수 있다. 이러한 문제를 해결하기 위하여 단말기 관리에 대한 필요성이 대두되어, 이를 위한 기구로서 OMA(Open Mobile Alliance)가 결성되었다. OMA는 이동 단말기 관리의 표준으로 OMA DM(Device Management)을 제시하였다. 최근에 공개된 OMA DM v1.2는 이전 스펙보다 개선된 이동 단말기 관리 방법을 제공한다. 본 논문에서는 이동 단말기를 위한 OMA DM v1.2 기반 관리 에이전트에 대한 설계 및 구현 내용을 보인다. 이 에이전트는 C언어로 구현되었으며, 이동 단말기의 관리 객체를 획득, 변경 그리고 추가하기 위해 TNDS(Tree and Description Serialization)를 사용함으로써 적은 양의 네트워크 트래픽으로 효과적인 단말기 관리가 가능하다.

키워드 : 이동 단말, 관리 에이전트, OMA DM

Design and Implementation of the Management Agent for Mobile Devices based on OMA DM

Jae Young shin[†] · Youngwoo Jung[††] · Kwang Sun Ko[†††] · Young Ik Eom[††††]

ABSTRACT

With the rapid advancement of mobile communication, mobile devices are evolving into convergence devices with various multimedia capabilities. But, high performance of devices demands complicated settings, and thus contains latent error possibilities, and poor system performance caused by the collision of different softwares. To solve these problems, device management becomes an important issue. Open Mobile Alliance(OMA) developed OMA DM(Device Management), which is a device management standard. Recently released OMA DM v1.2 provides improved mobile device management methods compared to earlier versions. This paper introduces design and implementation of OMA DM v1.2 based management agent. By using Tree and Description Serialization(TNDS) for acquiring, adding, and editing objects in the mobile devices, it provides more effective device management with small amount of network traffic.

Key Words : Mobile Device, Management Agent, OMA DM

1. 서 론

이동 통신 서비스의 발달과 더불어 이동 단말기들은 카메라, DMB(Digital Multimedia Brodcating), 영상전화와 같은 다양한 멀티 미디어 기능들을 포함할 뿐만 아니라, 최근에는 이러한 여러 가지 기능들과 컨텐츠가 융합되어 제공되는 컨버전스 단말기로 진화하고 있다. 그러나, 이러한 단말기들의 고기능화는 복합한 설정으로 인한 잠재적인 오류 가능성과 컨텐츠 및 서비스 제공에 필요한 다수의 소프트웨어간의 충돌로 인한 시스템의 성능 저하 등의 문제점들을 내포한다. 이로 인해, 이동 단말기 관리의 필요성은 계속적으로 증가하고 있다 [1, 2].

이동 단말기의 관리는 단말기의 초기 구성 정보의 설정, 단말기 정보의 지속적인 업데이트와 부가적인 서비스 설치를 비롯한 단말기로부터의 관리 정보 수집 및 단말기에서 생성된 이벤트의 처리 등을 포함한다[3, 4].

기존의 단말기 관리 방법으로는 네트워크 단말기 관리를 위해 많이 사용되는 SNMP(Simple Network Management Protocol)[5], 마이크로 소프트 윈도우 기반의 시스템을 관리하는데 사용되는 WEBM(Web Based Enterprise Management) [6] 그리고 이동 단말기의 관리를 위해 사용되는 WAP(Wireless Application Protocol) 프로비저닝[7]이 존재한다. 이러한 방

※본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기술개발사업의 지원에 의한 것임 (2007-0266-0100).

† 준 회 원 : 성균관대학교 이동통신공학과 석사과정
†† 준 회 원 : 성균관대학교 전자전기컴퓨터공학과 석사과정
††† 정 회 원 : 성균관대학교 이동통신공학과 연구교수
†††† 종신회원 : 성균관대학교 정보통신공학부 교수

논문접수 : 2007년 11월 22일, 심사완료 : 2008년 1월 7일

DOI: 10.3745/KIPSTC.2008.15-C.1.61

법들은 복잡성, 크기 그리고 기반 프로토콜에 의존적이라는 점에서 이동 단말기의 관리 방법으로 효과적이지 못하다.

OMA(Open Mobile Alliance)는 단말기 관리를 위한 표준으로 OMA DM(Device Management)를 제공하고 있으며, 최근에는 OMA DM v1.2가 배포되었다. OMA DM은 XML 기반으로 언어 중립적이고 기반 프로토콜에 중립적이며 경량이다. 또한 실시간에 관리 대상을 증가시키거나 줄일 수 있는 명령을 통하여 관리 객체에 대한 확장성을 제공한다[8, 9].

본 논문에서는 이동 단말기 관리 에이전트인 MDM(Mobile Device Management) Agent를 보인다. MDM 에이전트는 OMA DM v1.2를 기반으로 설계되었으며, C언어로 구현되었다. 또한, MDM 에이전트는 OMA DM v1.2에서 제공하는 TNDS(Tree and Description Serialization)를 사용함으로써 적은 양의 네트워크 트래픽을 통해 보다 효과적인 단말기 관리가 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 OMA DM 표준에 대해서 살펴 보고, 3장에서는 본 논문에서 제안하는 MDM 에이전트에 대해 설명한다. 4장에서는 실험을 통해 MDM 에이전트의 적합성을 검증한 뒤, 마지막으로 5장을 통해 결론을 맺는다.

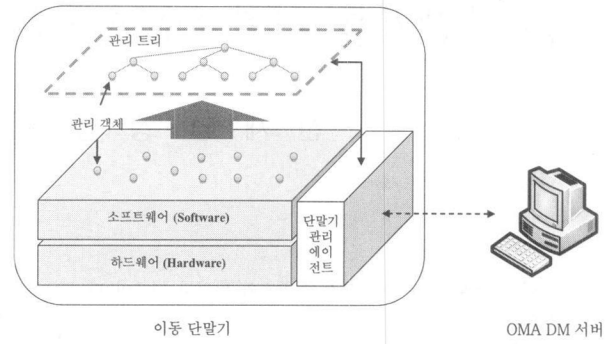
2. OMA DM (Open Mobile Alliance Device Management) 표준

본 장에서는 이동 단말기 관리를 위한 OMA DM 기반 구조를 살펴보고, OMA DM v1.1.2와 OMA DM v1.2를 비교 분석한다[8].

2.1 OMA DM 기반 구조

단말기는 물리적인 특징을 담당하는 하드웨어와 하드웨어를 관리하는 다수의 소프트웨어 그리고 OMA DM 서버와 상호작용을 통해 소프트웨어의 유지, 보수, 관리 및 펌웨어 업데이트 등의 작업을 수행하는 단말기 관리 에이전트로 구성된다[9]. OMA DM 서버는 관리 에이전트와 주기적으로 통신하며 관리 에이전트의 요구 사항에 응답 또는 처리를 담당하는 원격지에 위치한 단말기 관리 서버를 의미한다.

단말기 관리 에이전트를 통한 소프트웨어의 관리는 각 소프트웨어의 관리 객체를 추가, 수정, 삭제함으로써 이루어진다. 관리 객체는 서비스를 실행하기 위한 각각의 소프트웨어의 설정 값들을 의미하는 것으로써 단순한 소프트웨어의 환경 변수부터 펌웨어 이미지까지 다양한 형태로 존재한다. 이러한 관리 객체는 OMA DM 서버와 관리 에이전트 간의 상호작용을 통해 관리된다[10, 11].



(그림 1) OMA DM 기반 구조

관리 객체는 가상의 트리 구조를 가짐으로써 효과적인 관리 객체의 접근을 가능하게 한다. 다시 말해, 관리 서버는 관리 에이전트를 통해 관리 트리의 각각의 노드에 해당하는 관리 객체를 개별적으로 접근할 수 있을 뿐만 아니라 상위 노드에 위치한 관리 객체의 접근을 통해 그 노드의 하위에 위치한 모든 관리 객체까지 접근할 수 있다[2]. (그림 1)은 단말기 관리를 위한 기반 구조를 도시화한 것이다.

OMA DM 서버는 단말기 관리 에이전트에서 제공하는 프로비저닝(Provisioning), 펌웨어 업데이트 등과 같은 다양한 작업을 통해 관리 트리에 위치한 각각의 관리 객체를 수정함으로써 단말기의 구성 정보에 대한 관리를 비롯하여 소프트웨어 관리, 단말기 오류진단 등을 수행한다.

2.2 OMA DM v1.2

OMA DM v1.2는 관리 에이전트가 반드시 갖춰야 할 8가지 요구 사항을 제시한다. <표 1>은 OMA DM v1.2에서 제시하고 있는 8가지 요구사항을 보인다.

2002년에 배포된 OMA DM v1.1.2는 표 1의 1~7까지의 요구사항을 포함한다. 최근에 배포된 OMA DM v1.2에서는 이러한 요구사항에 추가적으로 XML 또는 WBXML 형태의 관리 트리를 전송하기 위한 방법에 대해 정의하고 있는 TNDS(Tree and Description Serialization)가 추가되었다.

〈표 1〉 OMA DM v1.2에서 제시하는 8가지 요구사항

요구 사항	내 용
Bootstrap	OMA DM 서버와 단말기의 관리 세션 초기화 절차 및 이에 필요한 프로파일에 대한 정의
Notification Initiated Session	단말기의 에이전트에게 세션 시작 요청을 보내기 위한 절차와 메시지에 대한 정의
Protocol	SyncML Representation Protocol 을 사용하는 관리 프로토콜과 관리 절차에 대한 정의
Representation Protocol	OMA DM 서버와 에이전트가 서로 상호 교환하는데 필요한 모든 XML 메시지에 대한 정의
Security	일반적인 보안, 전송 계층 그리고 어플리케이션 계층에 대한 보안 요구 사항에 대한 정의
Standardized Objects	모든 OMA DM에 적합한 단말기들에게 필수적인 관리 객체에 대한 정의
Tree and Description	관리 객체를 트리 구조로 구성한 관리 트리와 관리 트리의 노드 속성을 기술하기 위해 필요한 방법에 대한 정의
Tree and Description Serialization	관리 트리의 부분 또는 전체를 XML 또는 WBXML 형태의 바이트 스트림을 전송하기 위한 방법에 대한 정의

다음은 OMA DM v1.2의 특징을 나타낸다. 요구사항은 OMA DM v1.2를 만족하기 위해 요구되는 각 스펙의 내용이며 특징은 기존 OMA DM v1.1.2와 비교하여 변경되거나 추가된 사항이다.

- OMA DM v1.2 프로토콜에서는 에이전트와 OMA DM 서버가 주고 받는 메시지에 프로토콜 버전을 명시한다. 이는 서로 다른 버전으로 작성된 메시지에 대한 지원이 가능하다.
- Generic alert 명령을 사용하여 비동기적으로 OMA DM 서버에게 관리 객체에 관련된 단말기의 정보를 보낼 수 있는 기능을 지원한다.
- OMA DM v1.2 부트스트랩은 ACL(Access Control List)과 Access Type과 같은 관리 객체의 모든 속성을 포함할 수 있다.
- Exec 명령에 correlator를 포함할 수 있다. 이는 관리 명령과 비동기 명령 결과에 대한 식별자로 사용된다.
- 관리 트리의 전체 또는 부분을 XML 또는 WBXML을 통하여 TNDS(Tree and Description Serialization) 오브젝트로 인코딩 할 수 있다.
- 이동 단말기의 오류 또는 기능 추가를 위해 변경될 수 있는 펌웨어에 대한 업데이트를 제공하기 위하여 OMA DM의 확장 스펙을 정의하고 있다.

본 논문에서는 OMA DM v1.2를 만족하는 이동 단말기 관리 에이전트인 MDM(Mobile Device Management) 에이전트의 설계와 구현을 보인다. MDM 에이전트는 단말기의 초기 권한 설정, 단말기 구성 정보에 대한 관리, 단말기 펌웨어 다운로드 등의 기능을 수행 함으로써 단말기상의 소프트웨어 버그나 사용자의 잘못된 설정으로부터 발생하는 오작동을 방지 혹은 복구 한다. MDM 에이전트는 3장을 통해 자세히 다루도록 한다.

3. MDM(Mobile Device Management) 에이전트

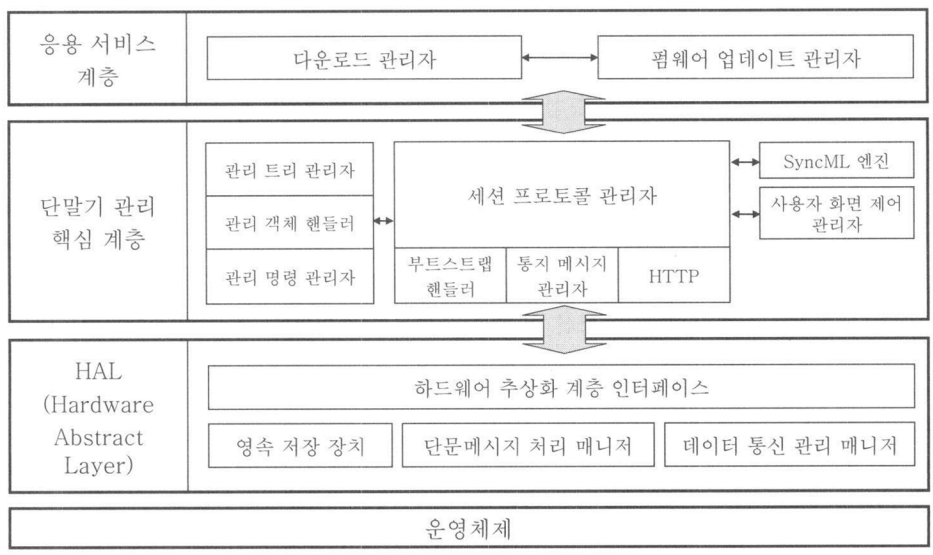
MDM 에이전트는 프로그램의 크기와 실행 속도 등을 고려하여 C언어로 구현하였다. 본 장에서는 MDM 에이전트의 논리적 계층과 물리적 설계에 대해서 설명하고, MDM 에이전트의 프로비저닝 기법을 소개한다.

3.1 계층 구조

MDM 에이전트는 다양한 이동 단말기 플랫폼의 최소한 변경을 통해 사용 가능하고 OMA DM v1.2를 기반으로 한 서비스에 대해 확장성을 가질 수 있는 구조를 갖는다. MDM 에이전트는 단말기 서비스와 같은 계층에 위치하며 내부의 계층 구조는 OMA DM 기반의 확장된 서비스를 포함하는 응용 서비스 계층과 OMA DM 스펙의 구현부인 단말기 관리 핵심 계층, 단말기의 파일 입출력, 네트워크, 유저 인터페이스와 결합될 수 있도록 정의한 하드웨어 추상화 계층 그리고 단말기의 하드웨어와 시스템 소프트웨어로 나누어진다. 또한 MDM 에이전트는 휴대폰, PDA와 같은 이동 단말기 내에 적재되어 이동 단말기를 관리하는 OMA DM 서버와 주기적으로 통신함으로써 이동 단말기를 관리한다. MDM 에이전트는 물리적으로 하나의 컴포넌트로 구성되며, 논리적인 계층 구조를 기반으로 각 기능별 모듈이 구성된다. (그림 2)는 각 계층별 MDM 에이전트의 세부 구조를 나타낸다.

<표 2>는 MDM 에이전트를 구성하는 주요 모듈과 그 기능을 나타낸다.

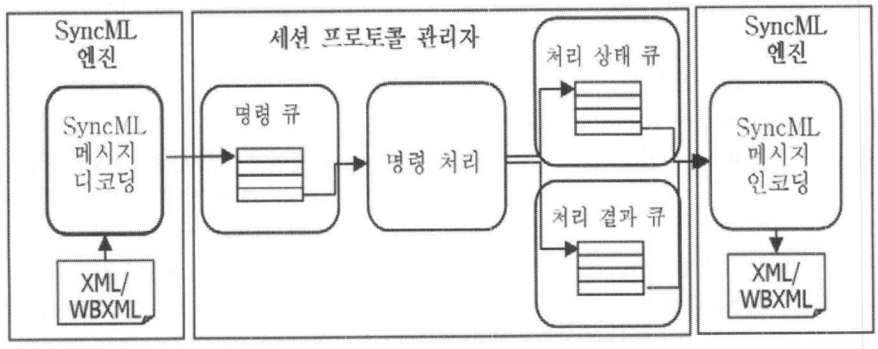
MDM 에이전트는 OMA DM 서버와 통신을 통해 단말기의 초기 구성 정보 설정, 단말기 정보의 지속적인 업데이트와 부가적인 서비스 설치, 단말기로부터 관리 정보 수집, 그리고 단말기에서 생성된 이벤트의 처리를 수행한다. 본 논문에서는 MDM 에이전트의 주요 모듈의 구현에 대해 자세히 설명한다.



(그림 2) 각 계층별 MDM 에이전트의 세부 구조

〈표 2〉 MDM 에이전트의 주요 모듈 및 그 기능

계 층	모 듈	기 능
응용 서비스 계층	다운로드 관리자	HTTP 또는 HTTPS를 통하여 다운로드 디스크립터를 해석하고 펌웨어를 이동 단말기로 다운로드 한다.
	펌웨어 업데이트 관리자	단말기 관리 트리의 펌웨어 업데이트에 관련된 노드들의 정보를 관리한다.
단말기 관리 핵심 계층	관리 트리 관리자	단말기 설정 정보 저장에 사용되는 추상적인 관리 트리를 관리한다.
	관리 객체 핸들러	관리 에이전트의 관리 트리 관리자와 하드웨어 추상화 계층 인터페이스간의 중계 역할을 한다.
	관리 명령 관리자	관리 객체의 추가, 삭제, 변경, 탐색 그리고 실행 등의 모든 관리 명령들을 다룬다.
	세션 프로토콜 관리자	OMA DM 표준에 따라 순차적으로 전송되는 패키지를 포함하여 서버와의 세션을 관리하며 서버와 에이전트 사이의 Basic 또는 MD5 상호 인증을 관리한다
	부트스트랩 핸들러	부트스트랩 메시지를 해석하고 관리 트리에 등록한다.
	통지 메시지 관리자	OMA DM 서버의 통지 메시지를 디코딩하고, 메시지에 대한 인증을 수행한다.
	SyncML 엔진	수신된 메시지를 파싱하여 각각의 관리 명령에 대한 외부 콜백 기능을 수행하며, 에이전트의 처리 결과를 인코딩한다.
	사용자 화면 제어 관리자	UI Alert 메시지를 분석하여 단말기의 사용자 인터페이스를 제어한다.
하드웨어 추상화 계층	하드웨어 추상화 계층 인터페이스	네트워크, 사용자 인터페이스, 파일 입출력, 단말기 정보와 같이 단말기의 플랫폼에 의존적인 부분에 대한 스텁을 정의한다.
	영속 저장 장치 관리 매니저	단말기의 저장장치를 관리하며 MDM 에이전트의 관리 트리를 영속적으로 보관한다.
	단문메시지 처리 매니저	단말기로 전달된 단문 메시지를 관리하며 부트스트랩 메시지와 관리 서버 통지 메시지를 받아 MDM 에이전트에 전달한다.
	데이터 통신 관리 매니저	단말기의 데이터 통신을 관리하며 MDM 에이전트의 데이터 통신 부분을 지원한다.



(그림 3) MDM 에이전트의 큐를 사용한 세션 처리 흐름

3.2 MDM 에이전트의 구현

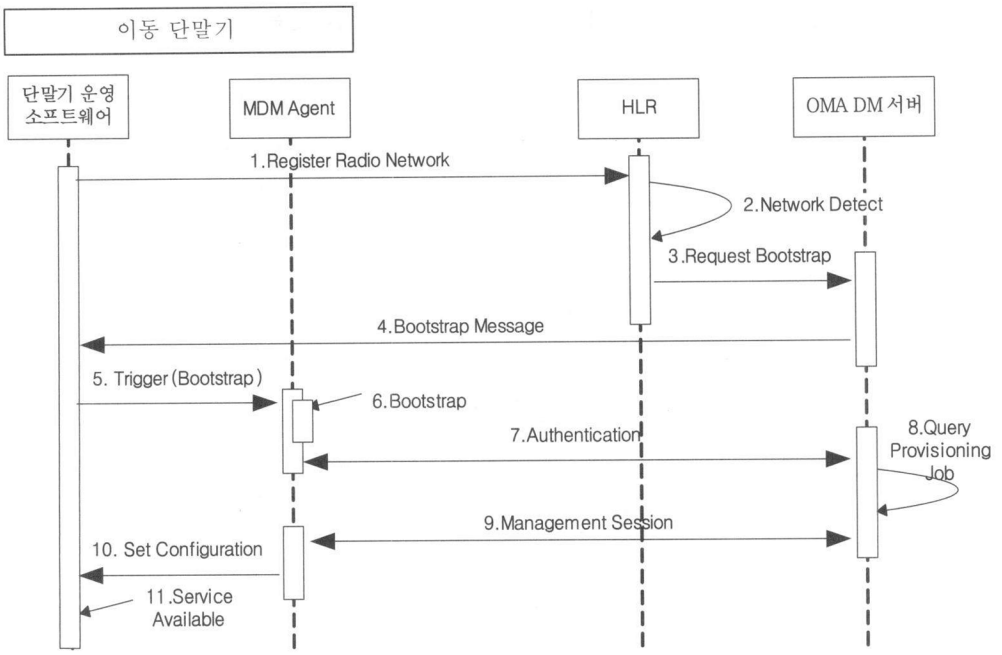
MDM 에이전트는 위의 계층 구조에서 설명한 응용 서비스 계층과 단말기 관리 핵심 계층의 모든 기능을 포함하는 하나의 컴포넌트 형태로 구성하고 하드웨어 추상화 계층은 마이크로소프트 윈도우 운영체제를 기반으로 구현하였다.

OMA DM 서버와의 관리 세션의 유지와 처리 그리고 처리 결과를 효율적으로 관리 하기 위해서 세션 프로토콜 관리자는 세 가지의 큐(Queue)를 사용하여 관리 세션을 처리한다. (그림 3)은 MDM 에이전트의 큐를 사용한 세션 처리 흐름을 나타낸다. 세 가지의 큐는 OMA DM 서버의 명령을 넣는 명령 큐와 명령 처리에 대한 결과를 넣는 처리 상태 큐와 처리 결과 큐로 구성된다.

관리 트리 관리자는 단말기 관리 설정 정보를 추상화된 트리 형태로 관리를 하며 관리 트리를 영속적으로 유지하기 위해서 두 가지의 방법을 제공한다. 첫째는 존재하는 관리 설정 정보를 XML형태의 파일로 관리하고 MDM 에이전트의 관리 세션이 시작되면 메모리에 정보를 위치시킨다. 둘째는 관리 설정 정보를 트리형태의 디렉토리로 구성하여 각 디렉토리에 관리 설정 정보 파일을 위치시킨다. 이는 MDM 에이전트가 관리 세션이 시작되면 관리에 필요한 설정 정보 파일만을 메모리에 위치시키므로 관리 세션 중 메모리 사용에 대해서 효율적이다. MDM 에이전트에서 OMA DM 서버의 관리 명령의 처리는 관리 명령 관리자를 통하여 접근 권한을 확인한 후 관리 트리 트리 매니저에서 관리되는 설정 정보를 처리한다. 본 논문에서는 단말기의 초기 구성 정보 설정을 위한 MDM 에이전트의 프로비저닝 기능에 대해 자세히 설명한다.

3.3 MDM 에이전트의 프로비저닝 프로토콜

프로비저닝은 단말기 제조사에 의해 제작된 단말기를 무선 네트워크 망에서 사용할 수 있도록 단말기의 설정 정보를 변경하여 초기 서비스가 가능하도록 하는 일련의 작업을



(그림 4) MDM 에이전트와 OMA DM 서버간의 관리 세션 프로토콜

말한다. (그림 4)은 이동 단말기의 프로비저닝을 위한 MDM 에이전트와 OMA DM 서버간의 관리 세션을 보인다. HLR는 이동통신 단말기의 위치 정보, 인증정보, 서비스 정보, 권한 및 부가 정보를 실시간으로 관리하는 시스템이며 새롭게 등록된 이동통신 단말기 정보를 획득한다. 획득된 정보를 바탕으로 HLR은 이동통신 단말기의 프로비저닝 시작을 OMA DM서버에게 알린다.

단말기가 프로비저닝 절차를 진행하기 위해서는 단말기의 MDM 에이전트는 부트스트랩 과정을 통하여 OMA DM 서버와 세션을 수행할 수 있는 상태가 되어야 한다. 따라서 단말기가 무선 네트워크에 등록되면, HLR은 OMA DM 서버에게 부트스트랩을 요청 한다. OMA DM 서버는 부트스랩 프로파일을 작성하여 정해진 SMS(Short Message Service)의 채널을 통하여 단말기에 보낸다. 단말기는 OMA DM 서버로부터 받은 메시지를 MDM 에이전트에게 전송한다. MDM 에이전트는 부트스트랩에 포함된 OMA DM 서버 주소, OMA DM 서버 아이디와 패스워드, 에이전트 아이디와 패스워드 그리고 네트워크 설정을 관리 트리에 등록한 후, OMA DM 서버에 접속한다. 이 때, OMA DM 서버는 접속을 요청한 MDM 에이전트를 대상으로 인증을 수행한다. 인증에 성공하면, OMA DM 서버는 접속된 MDM 에이전트의 단말기 정보와 가입자 정보를 확인하고 서비스 제공을 위하여 프로비저닝을 수행한다. MDM 에이전트는 프로비저닝 정보를 OMA DM 서버로부터 받아 단말기의 설정을 변경한 후 정상적인 서비스의 이용이 가능한 상태가 되도록 초기화한다.

4. 실험 결과 및 평가

본 장에서는 MDM 에이전트의 구현이 OMA DM v1.2를 기반하여 개발되었음을 검증하고 이를 바탕으로 관리 객체를 제어하기 위해 필요한 OMA DM 서버와 MDM 에이전트간의 데이터 전송량을 분석한다.

4.1 MDM 에이전트의 적합성 검증

본 절에서는 논문에서 소개된 MDM 에이전트가 OMA DM v1.2를 기반으로 올바르게 구현되었음을 보인다. 이를 위하여 OMA에서 배포하는 SCTS(SyncML Conformance Test Suite) 1.2 적합성 검증 툴을 사용한다[5]. 본 적합성 검증 방법은 OMA DM 서버의 역할을 하는 SCTS가 MDM 에이전트에게 명령을 전송하고 이에 대한 결과값을 돌려받아 확인 하는 방식으로 진행된다. 본 검증은 SCTS가 제공하는 24개의 그룹으로 구성된 48개의 검증 항목에 대해 수행되었다. <표 3>은 SCTS가 제공하는 24개의 그룹의 검증 항목을 나타낸다.

MDM 에이전트는 SCTS의 24개 그룹으로 구성된 48개의 모든 검증 항목에 대해서 올바른 결과값을 전송하였다. 따라서 MDM 에이전트는 OMA DM v1.2를 지원하는 다른 관리 서버와 연동되어 이동 단말기의 관리를 수행할 수 있다. (그림 5)는 SCTS를 이용한 검증 화면을 보인다.

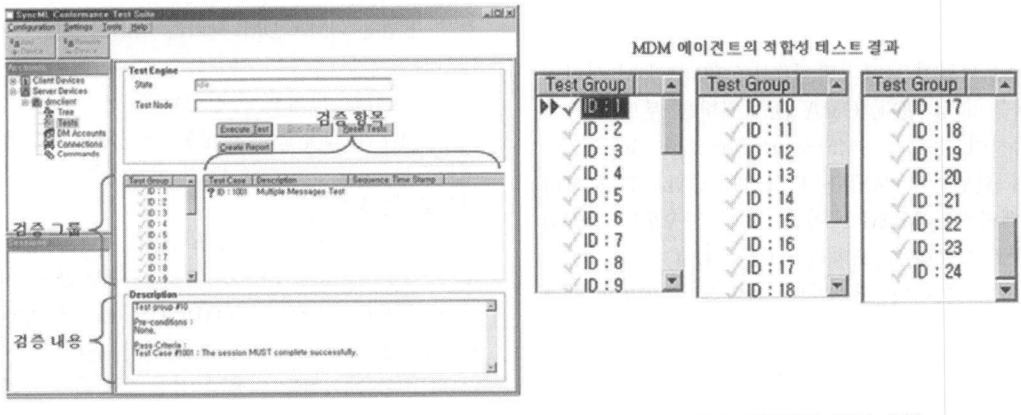
4.2 관리 객체 제어를 위한 메시지 전송량 비교

OMA DM v1.2의 TNDS(Tree and Description Serialization)는 XML 또는 WBXML 형태로 구성된 관리 트리의 전송을 지원함으로써 다량의 관리 객체에 대한 일괄적인 제어가 가능하다. 본 MDM 에이전트는 TNDS(Tree and Description Serialization)를 지원함으로써 관리 객체의 정보를 획득, 변경, 추가 시 필요한 메시지 전송량을 감소시킨다. 이는 다음과 같은 메시지 전송량 비교 실험을 통해 검증한다.

본 절에서는 3.3절에서 설명한 MDM 에이전트의 프로비

〈표 3〉 SCTS에서 제공하는 24개 그룹의 검증항목

검증 그룹	검증 항목	검증 그룹	검증 항목
ID 1	유효한 Alert 명령의 전송	ID 13	루트 노드(Root node)의 접근 권한 확인
	단말기 기본 정보의 전송		인테리어노드(Interior node)의 Format속성의 Get명령 지원
	소스(Source)의 LocURI의 확인		인테리어노드(Interior node)의 Type속성의 Get명령 지원
ID 2	MD5로 인증 방법 변경		인테리어노드(Interior node)의 Size 속성의 Get명령 지원
ID 3	MD5인증 방법 지원		인테리어노드(Interior node)의 Name 속성의 Get명령 지원
	루트 노드(Root node)에 대한 Get명령 결과의 응답		리프 노드(leaf node)의 Size 속성의 Get명령 지원
	리프 노드(Leaf node)의 Get명령 결과의 응답		영속적인 노드의 Name 속성 변경
	존재하지 않는 노느의 Get명령 결과의 응답		인테리어노드(Interior node)의 접근 권한 변경 지원
ID 4	HMAC의 지원과 불안정한 전송의 사용	ID 14	접근 권한의 강제성 확인
ID 5	인테리어 노드(Interior node) 의 추가	ID 15	접근 권한 설정
	리프 노드(Leaf node)의 추가	ID 16	리프 노드(Leaf Node)의 삭제
	존재하는 리프 노드(Leaf node)의 추가	ID 17	큰 객체 전송 방법의 지원
ID 6	Replace명령의 처리		큰 객체 전송 규칙
	존재하지 않는 노드의 Replace명령 거절		큰 객체 전송 결과
ID 7	Sequence명령의 처리		MaxObjSize 관리 객체의 확인
ID 8	사용자 상호작용 Alert 명령으로 사용자 허락 처리	ID 18	Get명령의 'list=Struct'속성 처리
	사용자 상호작용 Alert 명령으로 사용자 거절 처리	ID 19	Get명령의 'list=StructData'속성 처리
ID 9	인테리어 노드(Interior node)의 삭제	ID 20	통지 메시지 초기화 세션(Session)의 지원
	존재하지 않는 노드의 삭제	ID 21	제한된 AccessType속성을 가진 노드의 Exec명령 처리
	영속적인 노드의 삭제		접근 권한이 보호된 노드의 Exec명령 처리
ID 10	다중 메시지 처리	ID 22	인테리어 노드(Interior node)의 함축적인 추가 확인
ID 11	Atomic명령의 처리	ID 23	Get명령의 'list=TNDS'속성 처리
ID 12	DevInfo 표준 관리 객체의 구조	ID 24	Copy명령의 처리
	DevDetail 표준 관리 객체의 구조		
	DMAcc 표준 관리 객체의 구조		

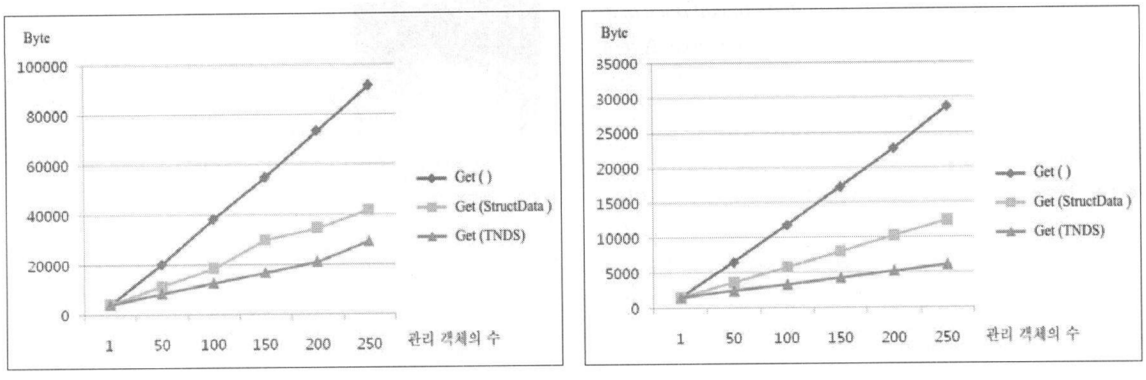


(그림 5) SCTS (SyncML Conformance Test Suite)를 이용한 MDM 에이전트 검증 화면

저닝 과정에서 발생하는 MDM 에이전트와 OMA DM 서버 간의 메시지 전송량을 다음과 같이 세 가지의 경우로 나누어 비교·분석 한다.

- 각각의 관리 객체의 정보를 개별적으로 전송하는 경우
- OMA DM v1.1.2에서 지원하는 structData을 사용하여 관리 객체의 정보를 전송하는 경우
- OMA DM v1.2에서 지원하는 TNDS(Tree and Description Serialization)를 이용하여 관리 객체의 정보를 전송하는 경우

본 실험의 환경 구성은 Intel Pentium 4.3GHz의 CPU와 2GB의 RAM을 사용한다. MDM 에이전트설치를 위한 운영체제는 Microsoft Window XP를 사용하며, OMA DM 서버는 SCTS 1.2를 이용한다. 또한 관리 객체를 제어하기 위한 메시지의 최대 패킷 사이즈는 20480byte로 제한한다.



(가) SCTS와 MDM Agent가 XML을 이용하여 통신할 경우 (나) SCTS와 MDM Agent가 WBXML을 이용하여 통신할 경우

(그림 6) 프로비저닝 과정에서 Get 명령에 관련된 메시지 전송량 측정 결과

본 실험은 다음과 같이 수행된다.

(1) SCTS는 관리 객체를 제어하기 위해 관리 명령(Get, Replace, Add, Delete, Exec, Copy)을 전송한다. 이 때, 관리 명령의 인자값 (none, StructData, TNDS 등)을 통해 리턴되는 결과값의 형태를 지정 할 수 있다.

(2) MDM 에이전트는 전송 받은 관리 명령을 관리 트리에 접근하여 처리하고, 그 결과값을 요청한 형태에 맞도록 작성하여 SCTS에게 전송한다.

프로비저닝 과정은 (1), (2)의 과정을 반복적으로 수행하며 이루어진다. 본 실험은 이러한 프로비저닝 과정에서 발생하는 MDM 에이전트와 OMA DM 서버 간의 메시지 전송량을 측정한다.

(그림 6)는 프로비저닝 과정에서 발생하는 Get 명령에 관련된 메시지 전송량을 측정한 결과이다.

(그림 6)의 실험 결과를 통해 다음과 같은 결과를 알 수 있다. 첫째 XML을 이용한 경우보다 WBXML을 이용하여 통신을 할 때 메시지 양이 감소한다. 이러한 결과는 WBXML은 XML문서를 압축한 형태로 전송하기 때문에 나타난다. 둘째 TNDS(Tree and Description Serialization) 사용함으로써 관리 객체를 제어하기 위해 필요한 OMA DM 서버와 MDM 에이전트간의 메시지 양을 감소 시킴을 알 수 있다. 이러한 전송량의 차이는 Get명령의 경우 각각의 관리 객체에 대하여 명령과 명령 처리 결과 그리고 결과값을 메시지에 포함하게 되며 TNDS(Tree and Description Serialization)명령의 경우 관리 객체 그룹을 대표하는 객체에 대하여 하나의 명령과 명령 처리결과를 포함하고 각 관리 객체의 결과값을 TNDS(Tree and Description Serialization)로 인코딩하여 이를 결과값으로 메시지에 포함하기 때문에 나타난다. MDM 에이전트는 OMA DM v1.2에서 제공하는 WBXML을 이용하여 통신을 하고 TNDS(Tree and Description Serialization)를 사용함으로써 다량의 관리 객체에 대한 제어가 가능하다.

5. 결 론

본 논문에서는 단말기 관리 표준인 OMA DM을 살펴보고, OMA DM v1.1.2와 OMA DM v1.2를 비교·분석함으로써 OMA DM v1.2의 개선된 특징을 서술하였다. 또한 본 논문에서는 이동 단말기 관리 에이전트인 MDM 에이전트를 보였다. MDM 에이전트는 OMA DM v1.2를 기반으로 설계되었으며, 프로그램의 크기와 실행 속도등을 고려하여 C언어로 구현되었다.

본 논문에서는 SCTS 1.2 적합성 검증 툴을 사용하여 구현한 MDM 에이전트가 OMA DM v1.2에 준하여 개발되었음을 보였다. 또한 프로비저닝 과정에서 발생하는 메시지 전송률을 분석하여 MDM 에이전트의 성능을 평가 하였다. 특히, OMA DM v1.2에서 제공하는 TNDS를 사용함으로써 적은 양의 네트워크 트래픽을 통해 보다 효과적인 단말기 관리가 가능함을 실험을 통하여 보였다.

참 고 문 헌

[1] Y. Hong, S. Son, and S Cho, "Device Management for Mobile WiMAX Services," *IEEE Trans on Consumer Electronics, Digest of Technical Papers*, 2007

[2] P. Oommen "A framework for integrated management of mobile-stations over-the-air," *Proc. of the 2001 IEEE/IFIP Int'l Symposium on Integrated Network Management*, 2001.

[3] S. Adwankar, S. Mohan, and V. Vasudevan, "Universal Manager: Seamless Management of Enterprise Mobile and Non mobile Devices," *Proc. of the 2004 IEEE Int'l Conf. on Mobile Data Management (MDM'04)*, 2004.

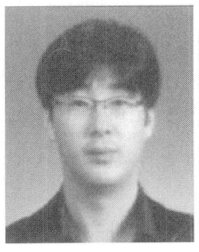
[4] R. Chakravorty and H. Ottevanger, "Architecture and implementation of a remote management framework for dynamically reconfigurable device," *Proc of the 2002 IEEE International Conf. on Networks*, Aug. 2002, pp.375 380.

[5] J. Case, M. Fedor, M. Schoffstall, J. Davin. A Simple Network Management Protocol. RFC 1157, Internet Engineering Task Force, May 1990.

[6] WBEM Initiative, *http://www.dmtf.org/standards/standard_wbem.php*

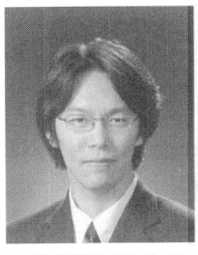
[7] Provisioning Architecture Overview Version 1.1, 20002, *http://www.openmobilealliance.org*

[8] Open Mobile Alliance, *http://www.openmobilealliance.org*
[9] R. State, O. Festor, and B. Zores, "An Extensible Agent Toolkit for Device Management," *Proc. of the 2004 IEEE/IFIP Int'l Symposium on Network Operations and Management*, 2004.
[10] SyncML Toolkit, *http://sourceforge.net/projects/syncmlctoolkit*
[11] Device Management Server, *http://www.funambol.com/opensource*



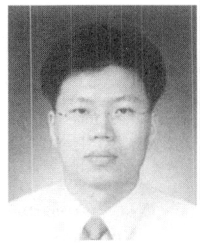
신 재 영

e-mail : jyoungshin@skku.edu
2001년 순천향대학교 제어계측공학과 (학사)
2001년~2004년 ㈜코베콤 부설기술연구소
2004년~현재 ㈜휴미트 부설연구소 선임연구원
2007년~현재 성균관대학교 이동통신공학과 석사과정
관심분야 : 모바일 미들웨어, 이동 에이전트, 운영체제 등



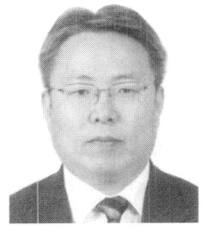
정 용 우

e-mail : withdubu@ece.skku.ac.kr
2006년 성균관대학교 정보통신공학부 (학사)
2006년~현 재 성균관대학교 전자전기컴퓨터공학과 석사과정
관심분야 : 이동 에이전트, 이동 에이전트 시스템 보안, 유비쿼터스 컴퓨팅 등



고 광 선

e-mail : rilla91@ece.skku.ac.kr
1998년 성균관대학교 정보공학과(학사)
2004년 성균관대학교 전기전자및컴퓨터공학부 (공학석사)
2007년 성균관대학교 전자전기컴퓨터공학과 (공학박사)
2007년~현재 성균관대학교 이동통신공학과 연구교수
관심분야 : 정보보호, 리눅스, 네트워크 등



엄 영 익

e-mail : yieom@ece.skku.ac.kr
1983년 서울대학교 계산통계학과 (학사)
1985년 서울대학교 전산과학과 (이학석사)
1991년 서울대학교 전산과학과 (이학박사)
2000년~2001년 Dept. of Info. and Comm. Science at UCI 방문교수
2004년~2005년 한국정보보호학회 논문지 편집이사
2005년~2006년 한국정보처리학회 학회지 편집위원장(상임이사)
1993년~현재 성균관대학교 정보통신공학부 교수
2007년~현재 성균관대학교 정보통신처 처장
관심분야 : 분산 컴퓨팅, 이동 컴퓨팅, 이동 에이전트, 시스템 보안, 운영체제, 내장형 시스템 등