# OMA DM을 기반으로 한 무선이동통신 단말기 관리 에이전트 설계 및 구현

(Design and Implementation of Wireless Device Management Agent based on OMA DM)

박 주 건 † 박 기 현 †† 장 대 진 ††† 장 명 숙 ††††

(Jugeon Pak) (Keehyun Park) (Daejin Jang) (MyungSook Jang)

**요 약** 무선이동통신 단말기의 기능과 서비스가 고도화됨에 따라 단말기 내의 복잡도가 증가되고 있고 관리 또한 어려워지고 있다. 이에 본 논문에서는 효율적인 단말기 관리를 위해 단말기 관리 표준안인 OMA(Open Mobile Alliance) DM(Device Management) 기술을 기반으로 하는 단말기 관리 에이전트를 설계 및 구현하였다. 제안된 에이전트는 현재 ADD, DELETE, REPLACE, GET 명령을 통해 기본적으로 설정 값을 관리할 수 있다. 추후 오류보고, 소프트웨어 설치 및 배포 등의 추가 기능을 위해 각 모듈을 분리하여 설계 및 구현하였다. 그러므로 관리 명령을 해석 및 생성하는 모듈의 수정 없이 명령어 처리를 담당하는 어댑터 부분만을 수정하여 기능을 추가할 수 있다. 본 논문에서 제시하는 단말기 관리 에이전트는 국내 표준 플랫폼인 위피(Wireless Internet Platform for Interoperability)를 기반으로 구현되었다. 따라서 개발된 에이전트의 이식성을 높일 수 있다. 또한 OMA DM 기술을 위피 플랫폼 상의 에이전트에 적용하여 구현함으로써, 위피 플랫폼의 기능적 적합성 검증 뿐만 아니라, OMA DM 기술의 확산을 위한 실질적인 개발 사례를 제시할 수 있다. 제안된 에이전트의 정상적인 동작여부를 검증하기 위해 SK WIPI SDK v1.2를 사용하였고, OMA DM 에이전트 테스트용 서버인 Funambol 서버와 연동하여 단말기 관리 작업을 수행하였다.

**키워드** : 단말기 관리, OMA, 위피

***Abstract*** Mobile devices become more complex and harder to manage due to the advance of technologies. In this paper, we designed and implemented the DM Agent based on OMA(Open Mobile Alliance) DM(Device Management) which is a standard for mobile device management. The proposed DM Agent basically can manage parameter configuration through the commands ADD, DELETE, REPLACE and GET currently. Each module of the agent was designed and implemented separately for additional functions such as error reporting and software management. So the additional functions can be added by modifying Adaptor part which processes the management commands instead of modifying other modules which analyze or generate management commands. The DM Agent proposed in this paper is implemented based on WIPI(Wireless Internet Platform for Interoperability), therefore portability of the agent can be improved. In addition, this research can give a development example for dissenminating OMA DM as well as verify functional compatibility of WIPI platform. SK WIPI SDK and a Funambol DM test server are used to test operations of the implemented DM Agent.

**Key words** : Device Management, OMA, WIPI

· 본 연구는 산업자원부 지방기술혁신사업(RT104-03-02) 지원으로 수행되었음

† 정 회 원 : 계명대학교 컴퓨터공학과
corea@kmu.ac.kr
†† 종신회원 : 계명대학교 컴퓨터공학과 교수
khp@kmu.ac.kr
††† 정 회 원 : 계명대학교 컴퓨터공학과
djjang@kmu.ac.kr
†††† 정 회 원 : 계명대학교 컴퓨터공학과 강사
msj@kmu.ac.kr

논문접수 : 2007년 9월 6일
심사완료 : 2008년 3월 11일



정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제4호(2008.6)

## 1. 서 론

최근 다양한 서비스를 제공하는 고도화된 무선이동통신 단말기가 등장함에 따라 이를 편리하고 효율적으로 관리할 수 있는 기술이 필요하게 되었다. 단말기를 효율적으로 관리하기 위해서는 자동화되고, 원격 관리가 가능한 기술이 필요하다. 이에 무선 통신 관련 회사들의 연합체인 OMA(Open Mobile Alliance)에서 단말기 관리 표준안인 OMA DM(Device Management)을 제정하였다. 이에 따라 과거 독자적인 단말기 관리 기술로 서비스를해오던 국내외 이동통신사들과 단말기 업체들도 최근 OMA DM 표준안을 적극 반영하고 있다. 또한 국가 간 경계가 무너지는 3세대 이동통신 서비스가 확대되면서 OMA DM은 단말기 관리에 관한 세계적인 표준으로서 앞으로 더욱 확산될 것으로 예상되는 기술이다[1].

OMA DM 기술을 사용하면 다음과 같은 이점이 있다. 첫째 중앙의 관리 서버가 단말기를 관리하기 때문에 많은 수의 단말기를 효율적으로 관리할 수 있다. 둘째, 관리 서버를 통해 관리자가 단말기내의 오류를 진단 및 복구할 수 있으므로, 사용자의 편리성이 증대된다. 셋째, 원격으로 단말기 관리가 이루어지므로 고객 지원 비용을 절감할 수 있다.

본 논문에서는 OMA DM을 기반으로 단말기 관리 에이전트를 설계 및 구현하였다. OMA DM에서 단말기 관리 시스템은 관리 서버와 관리 에이전트로 이루어진다. 관리 서버는 사용자의 단말기 내에 있는 관리 에이전트에게 관리 명령을 전달하는 역할을 한다. 관리 에이전트는 사용자의 단말기에 탑재되는 프로그램으로써, 관리 서버로부터 수신한 관리 명령을 단말기에 반영하는 역할을 한다[2]. 또한, 본 논문에서는 제조사가 다른 단말기간의 이식성을 높이기 위해 위피 플랫폼에 구현하였다. 위피는 C와 자바가 동시에 지원되는 구조를 가지며, 하드웨어나 단말기가 사용하는 운영체제에 관계없이 실행 및 포팅될 수 있게 설계된 표준화된 플랫폼이다. 위피 플랫폼에 단말기 관리 에이전트 구현을 위해서 SK텔레콤에서 제공하는 소프트웨어 개발 키트인 SK WIPI SDK v1.2를 사용하였다. 또한 본 논문에서 구현한 단말기 관리 에이전트의 정상적인 동작여부를 검증하기 위해 Sync4j에서 제공하는 OMA DM 테스트용 서버인 Funambol 서버와 연동하여 단말기 관리 작업을 수행하였다[3].

현재 정보통신부와 이동통신업계는 위피를 국제 표준으로 만들기 위해 OMA에 국제 표준으로 제안하였다. 위피 플랫폼이 국제 표준이 되기 위해서는 다양한 컨텐츠와 응용프로그램의 개발이 선행되어야 한다. 본 논문에서는 위피 플랫폼을 기반으로 한 단말기 관리 에이전트의 개발 사례를 제시함으로써, 위피 플랫폼에 대한 기능 및 성능적인 적합성을 검증할 수 있고 위피 플랫폼의 확산 및 국제 표준화에 기여할 수 있다. 또한 OMA DM 기술이 지속적으로 발전하기 위해서는, 다양한 플랫폼을 기반으로 한 학술연구가 필요하며 위피 플랫폼을 기반으로 하여 구현함으로써 그 기초를 제공할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구 부분으로, OMA DM과 위피에 대해 소개한다. 3장에서는 단말기 관리 에이전트의 설계 및 구현 결과를 제시한다. 4장에서는 구현된 단말기 관리 에이전트의 검증 결과를 제시한다. 5장에서는 본 논문의 결론과 향후 연구 방향에 대해서 제시한다.

## 2. 관련연구

무선이동통신 단말기 관리 에이전트를 구현하기 위해서는 다음과 같은 사항이 고려되어야 한다. 첫째 단말기 관리에 관한 국제 표준안인 OMA DM을 기반으로 구현하여 표준을 따르도록 한다. 둘째 단말기 관리 에이전트는 다양한 단말기에 적용 가능해야 하며, 중복 개발이 없도록 구현되어야 한다. 본 장에서는 위의 요구사항을 만족하는 단말기 관리 에이전트를 구현하기 위해 적용된 기반 기술인 OMA DM과 위피 플랫폼에 대해 소개한다.

### 2.1 OMA DM 표준

OMA는 SyncML DS(Data Synchronization) 표준안을 제시한 후 무선이동통신 단말기 관리의 필요성을 인식하고 SyncML DS를 관리 용도로 확장하여 단말기 관리 표준안을 제시하였다. OMA DM에서 관리 서버는 단말기 내 관리 객체의 정보를 읽고, 새로운 내용을 추가 또는 삭제하기 위해 단말기 관리 에이전트에게 관리 명령을 전달하는 역할을 한다. 관리 명령을 수신한 에이전트는 관리 명령을 관리 객체에 적용함으로써 관리를 수행한다[4]. 관리 명령은 하나의 관리 메시지 안에 포함되고, 이 메시지들은 논리적으로 하나의 패키지를 이룬다. 관리 객체는 관리의 대상이 되는 단말기의 구성 요소이다. OMA DM에서 관리 객체는 트리 형태로 구성되며, 트리에서 각 객체들은 노드로 표현된다[5,6]. 관리 서버는 관리 세션을 통해 에이전트 내의 관리 객체들에 대한 트리를 탐색하여 객체를 추가 또는 삭제함으로써 단말기를 관리할 수 있다. 본 논문에서는 관리 대상이 되는 단말기 구성 파라미터를 단말기 관리 객체로 정의하였고, 트리 구조로 계층화하였다. 또한 관리 트리를 통해 단말기를 관리하는 방식을 적용하였다.

### 2.2 위피

본 논문에서 구현한 단말기 관리 에이전트는 단말기 간의 이식성을 높이기 위해 표준화된 통합 플랫폼인 위피 기반으로 구현되었다. 국내의 이동통신 업체들은 그동안 회사마다 각기 다른 방식의 무선 인터넷 플랫폼을 사용하였는데, 이 때문에 응용프로그램과 서비스를 중복으로 개발하여야 했다. 위피는 이러한 낭비 요소를 줄일 목적으로 제안된 표준 무선인터넷 플랫폼이다. 위피는 C와 자바가 동시에 지원되는 구조를 가지며, 플랫폼과 응용프로그램은 하드웨어나 단말기가 사용하는 OS에 관계없이 실행 및 포팅될 수 있도록 설계되었다[7,8]. 따라서 통합된 플랫폼인 위피에 단말기 관리 에이전트를 구현하면, 각각의 단말기들 간 이식성을 높일 수 있고, 중복개발을 피할 수 있다.

## 3. 단말기 관리 에이전트 설계 및 구현

본 논문에서 제안된 단말기 관리 에이전트는 구조적 설계를 위해 관리 단계 및 메시지 처리 작업별로 각 모듈을 구분하여 엔진을 설계하였다. 또한 에이전트의 확장성을 고려하여 새로운 기능이 추가되더라도 다른 모듈의 수정없이 관리 어댑터만 수정하면 되도록 분리하여 설계하였다[9,10]. 본 논문에서 구현된 단말기 관리 에이전트는 OMA DM을 기반으로 위피 플랫폼에 구현되었다. SyncML 툴킷 4.3과 SK 위피 SDK를 사용하였으며, C언어로 구현되었다. SyncML 툴킷은 관리 메시지를 분석하고, 명령을 생성하는 역할을 하는 공개 소프트웨어이다[11]. SK 위피 SDK는 위피 기반의 응용프로그램을 개발하기 위해 SK 텔레콤에서 제공하는 개발 도구이다. SDK는 응용프로그램의 실행을 위한 위피 에뮬레이터를 제공하고, 응용프로그램 개발을 위한 API를 제공한다. 본 논문에서는 에이전트 구현을 위해 SK 위피 SDK가 제공하는 API를 사용하였고, 에뮬레이터에서 실행되었다. SDK에 의해 제공되는 에뮬레이터는 무선이동통신 단말기와 동일한 실행 환경을 제공한다.

그림 1은 단말기 관리 에이전트의 구조를 나타낸다. 단말기 관리 에이전트는 단말기 관리 엔진, 단말기 관리 어댑터, DDF, SyncML 툴킷으로 구성되어 있다.

OMA DM 표준에 기반하여 설계한 결과가 단말기 관리 엔진과 단말기 관리 어댑터이다. 단말기 관리 엔진 부분은 서버로 전송할 단말기 관리 메시지를 생성하고, 서버로부터 수신된 메시지로부터 관리 명령을 추출한다. 단말기 관리 엔진은 세션, 패키지, 메시지 핸들러로 구성되어 있으며, SyncML 툴킷과 연동하여 동작한다. 세션 핸들러는 관리 서버와의 세션을 담당하며 각 세션에 사용할 세션 ID를 관리한다. 패키지 핸들러는 각 패키지를 처리하고, 생성하는 기능을 담당하며, 메시지 핸들러는 수신한 관리 명령을 처리하고, 송신을 위한 메시지를 생성하는 기능을 담당한다. SyncML 툴킷은 OMA 표준화에 주도적으로 참여하고 있는 기업들이 공동으로 개발한 공개 소프트웨어로써, 메시지의 해석, 명령어 생성, 데이터 송수신 등의 기능을 한다.

그림 2는 단말기 관리 에이전트의 내부 동작을 나타낸다. 서버와의 세션이 시작되면, 세션 핸들러는 관리 서버로부터 관리 메시지를 받아 SyncML 툴킷에 전달한다. SyncML 툴킷은 이 메시지를 분석한 후, 해당하는 메시지 핸들러의 콜백 함수를 호출한다. 호출된 함수에서 명령을 처리하고, 실질적인 관리를 위해 단말기 관리 어댑터 내의 함수를 호출한다. 단말기 관리 어댑터는

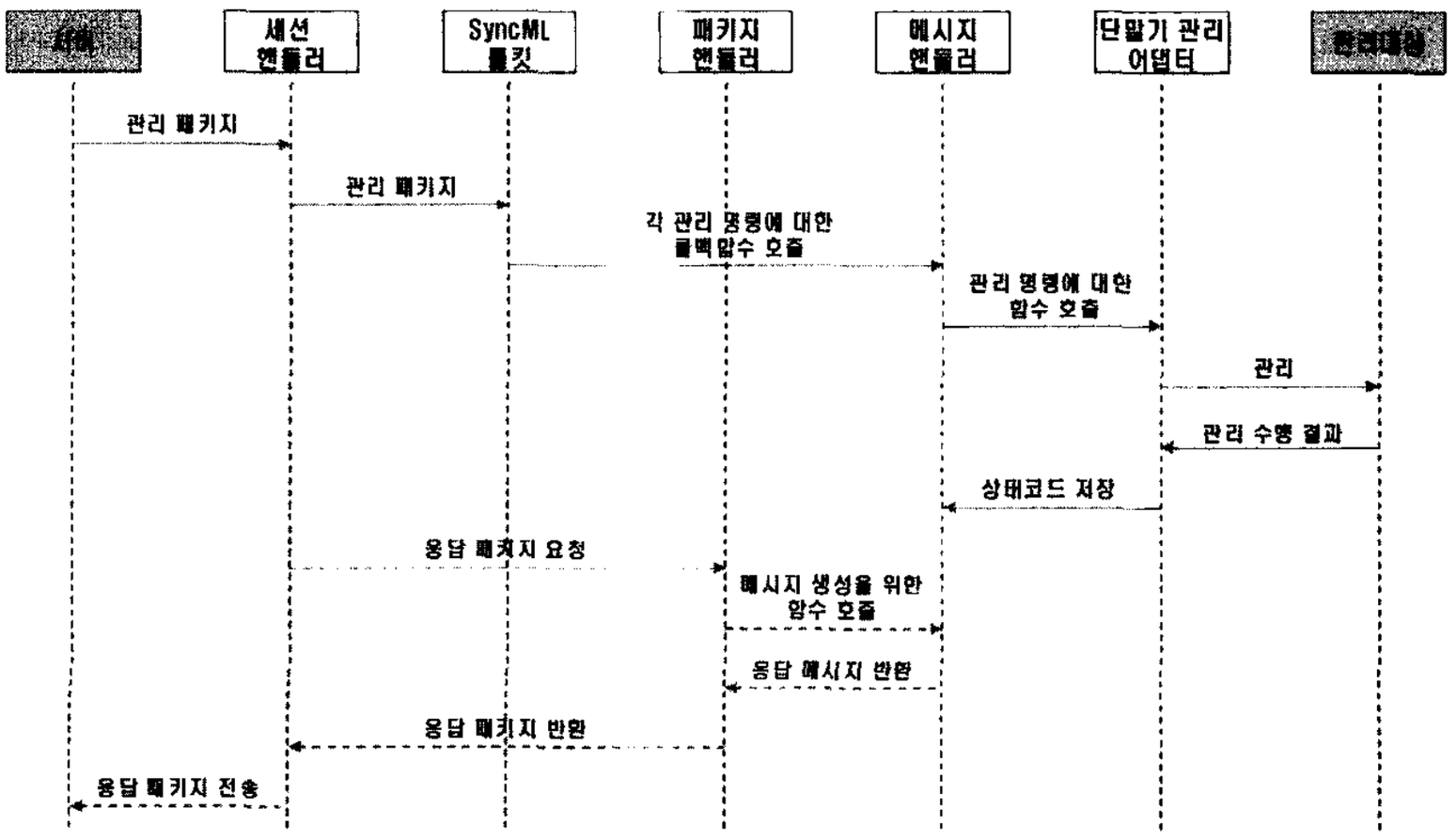



그림 1 단말기 관리 에이전트의 내부 동작

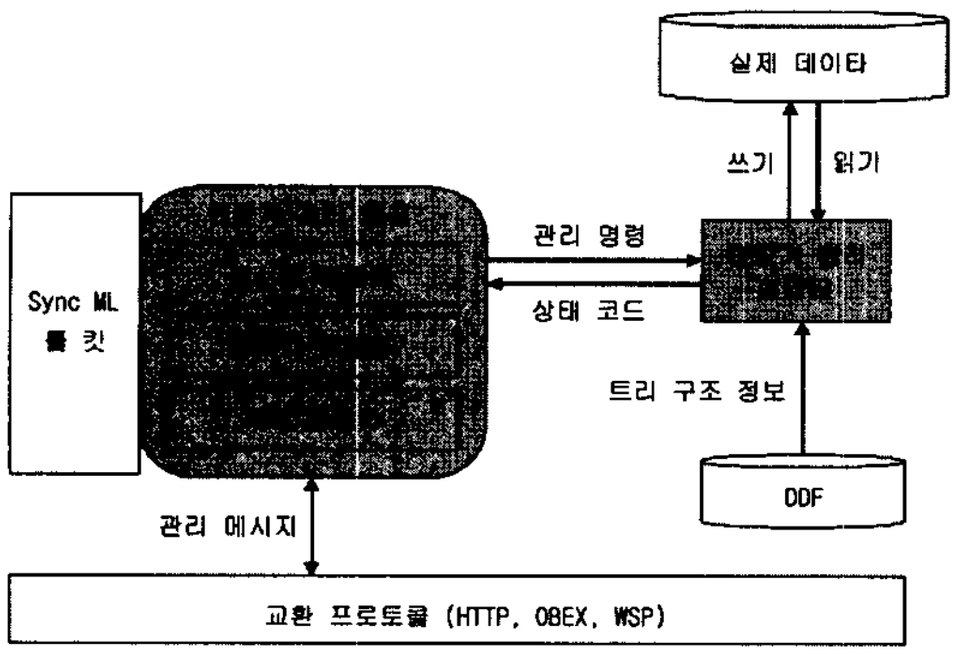


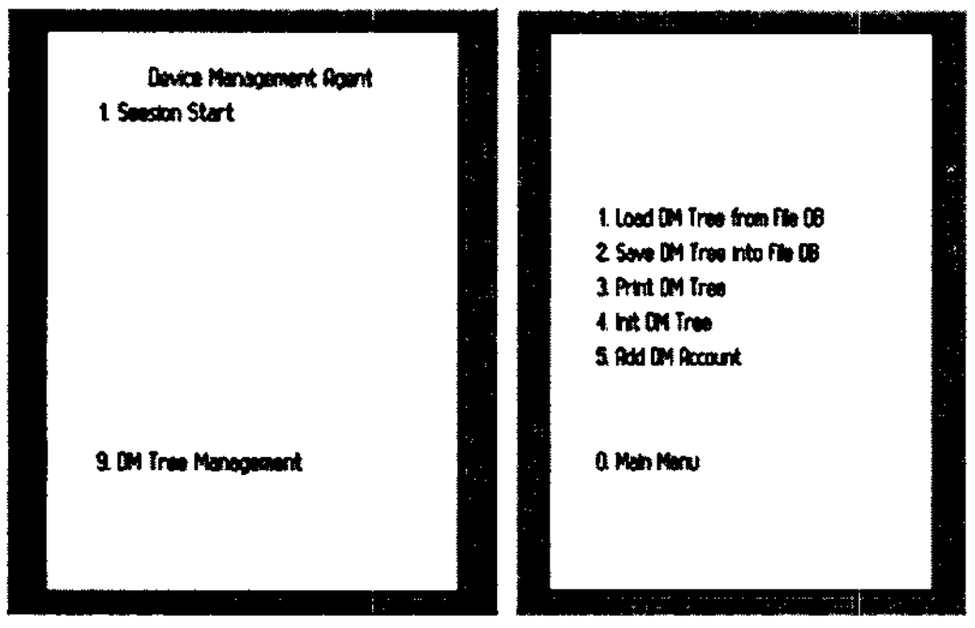

그림 2 단말기 관리 에이전트의 구조

관리 대상에 접근하여 실제 명령어에 대한 처리를 수행하고, 상태 코드를 메시지 핸들러에게 반환한다. 반환된 상태코드는 메시지 핸들러에 의해 상태 코드 리스트에 저장된다. 상태 코드 리스트에 저장된 값은 추후 관리 서버에게 응답 패키지를 보낼 때 사용된다.

에이전트에서 관리 서버로 응답 패키지를 보낼 때, 세션 핸들러는 패키지 핸들러에게 응답 패키지 구성을 요청한다. 패키지 핸들러는 해당 패키지 번호에 알맞은 메시지를 생성하기 위해 메시지 핸들러의 함수들을 호출한다. 메시지 핸들러는 실제 단말기 관리 메시지를 구성하고, 완성된 패키지를 세션 핸들러로 반환하고, 세션 핸들러는 이 패키지를 관리 서버로 전송한다.

그림 3은 단말기 관리 에이전트의 실행 화면을 나타낸다.

그림 3(a)는 단말기 관리 에이전트의 초기 실행 화면으로 서버와의 세션을 시작하거나 트리 조작 화면으로 이동할 수 있다. (b)는 트리 조작화면으로 사용자의 화면에 5개의 메뉴가 나타난다. 이 메뉴를 통해 사용자는 관리 트리를 파일로부터 읽어오거나 파일에 저장할 수 있고, 관리 트리를 화면에 출력할 수 있다. 또한 트리 정보를 초기화 할 수 있고, 서버와의 통신에 사용할 사용자 계정을 추가할 수 있다.




(a) 초기 실행 화면 (b) 트리 조작 화면

그림 3 단말기 관리 에이전트 실행 화면

## 4. 검증

구현된 단말기 관리 에이전트에 대한 정상적인 수행 여부를 검증하기 위해서 Sync4j에서 제공하는 OMA DM테스트용 서버인 Funambol 서버와 연동하여 설정값 관리 시험을 하였다. Funambol 테스트용 서버는 기본적으로 ADD, DELETE, GET, REPLACE 명령어를 테스트할 수 있는 웹 기반의 조작 환경을 제공한다. 서버에서 생성된 관리 메시지는 HTTP 프로토콜을 통해 에이전트로 전달된다. 에이전트는 서버로부터 수신한 관리 메시지를 분석하여 관리 명령을 처리하고 처리 결과에 대한 응답을 서버에게 전송한다.

구현된 단말기 관리 에이전트의 정상적인 동작 여부를 확인하는 방법에는 두 가지가 있다. 첫째 수신한 관리 명령에 따른 에이전트 내 관리 트리의 변화를 확인하는 방법이 있다. 둘째 관리 서버와 관리 에이전트 간 주고받은 관리 메시지를 분석하는 방법이 있다. 우선 관리 명령에 따른 에이전트 내 관리 트리의 변화를 확인한 결과를 제시한다.

그림 4는 Funambol 테스트용 서버의 조작 환경 이용하여 단말기 관리 에이전트로 Add, Replace, Get, Delete 명령을 전송하는 것을 나타낸다. 구현된 단말기 관리 에이전트의 검증을 위해 노드값 Highest를 가지는 노드 Volume을 ./Operator/RingTone/의 자식 노드로 추가하는 Add 명령을 전송하였다. Volume 노드의 값을 Highest에서 Low로 변경하는 Replace 명령을 전송하였고, Volume 노드의 값을 확인하기 위한 Get 명령을

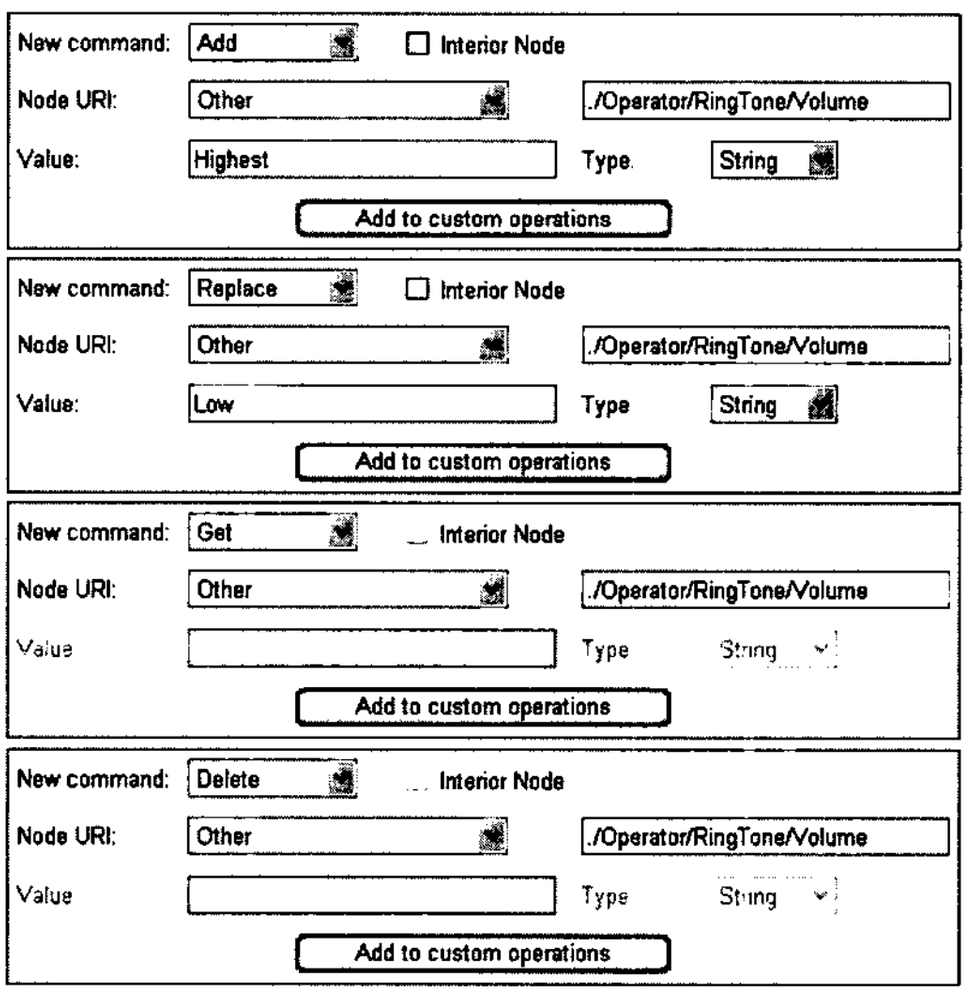



그림 4 Funambol 테스트용 서버의 조작화면

전송하였다. 마지막으로, 추가된 Volume 노드를 삭제하기 위한 Delete 명령을 전송하였다.

그림 5는 관리 명령에 따른 단말기 관리 에이전트 내의 트리 구조 변화를 나타낸다. 그림 5-a는 단말기 관리 명령을 실행하기 전 트리 구조를 보여준다. 그림 5-b는 관리 서버로부터 Add 명령어를 수신하고 처리한 후 변경된 트리 구조를 보여준다. 파선으로 된 부분에서 값이 Highest인 노드 Volume이 추가된 것을 알 수 있다. 그림 5-c는 관리 서버로부터 Volume 노드의 값을 Highest에서 Low로 변경하는 Replace 명령을 수신하고 처리한 후의 트리구조이다. 파선으로 된 부분에서 노드의 값이 Low로 변경된 것을 알 수 있다. 그림 5-d는 서버로부터 Volume 노드를 삭제하는 Delete 명령어를 수신하고 처리한 후의 트리구조이다. Volume 노드가 삭제된 것을 알 수 있다. Get 명령어는 특정 노드의 값을 확인하기 위한 명령어이다. 하지만 Get 명령을 수행하여도 트리 구조의 변화가 없기 때문에 이 방법으로는 올바른 동작 여부를 검증할 수 없고 실제 서버와 주고받은 메시지를 분석하여야 한다.

다음으로 관리 서버와 에이전트 간 주고받은 관리 메시지를 분석한 결과를 제시한다. 그림 6과 7은 Funambol 서버와 단말기 관리 에이전트가 주고받는 관리 메시지의 일부이다. 이 그림을 통해 본 논문에서 구현한 단말기 관리 에이전트의 정상적인 동작 여부를 확인 할 수 있다.

그림 6은 그림 4에서와 같이 Funambol 서버가 단말기 관리 에이전트로 Add, Replace, Get, Delete 명령어를 보낼 때 전달되는 메시지의 일부를 나타낸다. 각 명령어를 나타내는 <Add>, <Replace>, <Get>, <Delete> 태그가 메시지에 포함되어 있다. 대상 노드의 경로를 지정하기 위해 <LocURI> 태그를 사용하였고, <Data> 태그를 사용하여 데이터를 전달하였다.

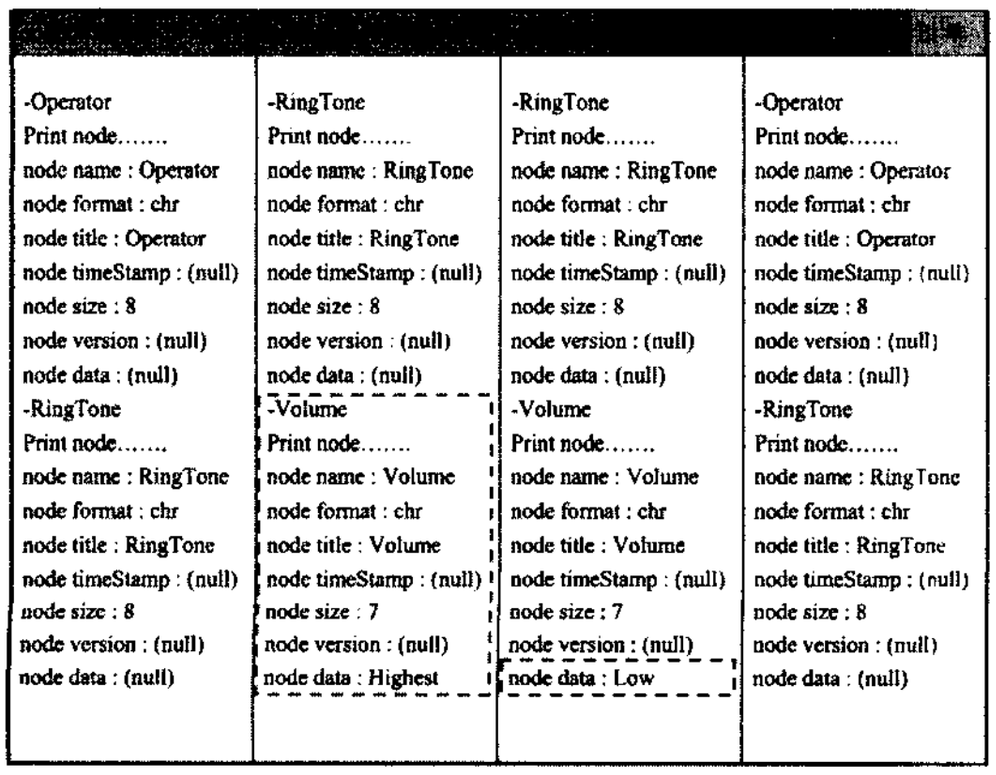



그림 5 단말기 관리 에이전트의 트리 구조


그림 6 Funambol 서버가 보내는 관리 메시지

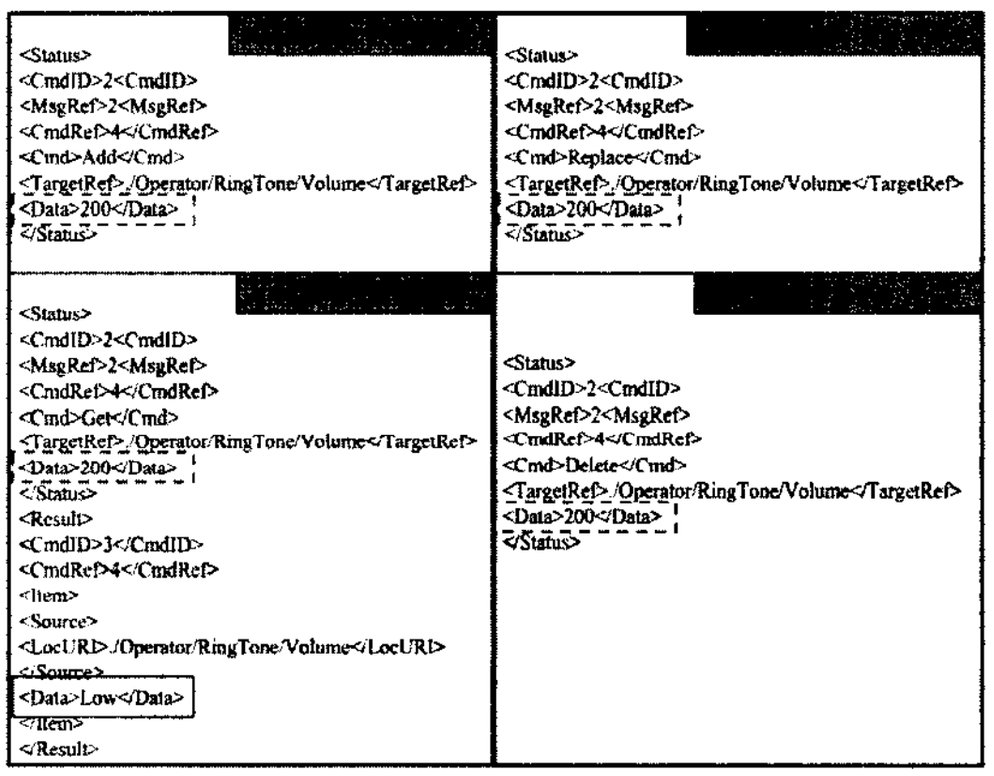



그림 7 단말기 관리 에이전트가 보내는 관리 메시지

그림 7은 단말기 관리 에이전트가 서버로부터 수신한 명령을 처리한 후 반환하는 응답 메시지를 나타내고 있다.

<Status> 태그를 사용하여, 각 명령어에 대한 처리 결과를 서버에게 전달하였다. <Data> 태그 부분에서 각 명령어에 대해 성공적으로 처리했음을 나타내는 '200' 상태 코드를 송신한 것을 알 수 있다. 이 부분은 그림에서 파선으로 나타나 있다. 그림 7-c는 서버로부터 Get 명령어를 수신하여 처리한 결과를 나타내는 그림으로써, Volume 노드의 값인 Low를 성공적으로 반환한다. 이를 통해 본 논문에서 구현한 단말기 관리 에이전트가 서버와의 통신을 올바르게 수행하고 있다는 것을 알 수 있다. 또한 OMA DM 테스트용 서버인 Funambol 서버와의 정상적인 동작을 통해 본 논문에서 구현한 단말기 관리 에이전트가 OMA DM 표준안에 기반하여 구현되었다는 것을 알 수 있다.

## 5. 결 론

본 논문에서는 OMA DM 기반의 무선이동통신 단말기 관리 에이전트의 구현 결과를 제시하였다. 그리고 단

말기 관리 에이전트의 설계 및 구현 구조를 제시하였다. 또한 구현된 단말기 관리 에이전트를 통해 설정 값을 관리 할 수 있음을 보였다. 현재 구현된 단말기 관리 에이전트는 현재 위피 에뮬레이터 위에서 동작하고 있으며 무선 이동통신 단말기의 특성을 고려한 효율적 구현을 위하여 단말기 관리 단위별로 조작 모듈을 구분하였고 전송 효율을 높이기 위해 WBXML 형태로 메시지를 교환하도록 구현하였다. 또한 Funambol 단말기 관리 서버를 이용하여 테스트한 결과 단말기 관리 에이전트가 정상적으로 동작하고 있다는 것을 확인하였다.

현재 단말기 펌웨어 업데이트, 소프트웨어 설치 및 삭제 등을 위한 Exec 명령어를 구현하고 있다. 향후 과제로는 단말기 관리 에이전트의 코드를 점검하여 불필요한 코드를 제거하고 메모리 사용에 대한 최적화 작업을 수행할 것이다. 더 나아가 구현된 에이전트를 이용하여 응답시간에 대한 성능 평가를 수행할 예정이다.

## 참 고 문 헌


[1] M.G. Han, C.G. Lee, and S.W. Lee, "A summary of OMA standardization activity on mobile service platform," Korean Information Science Society, Vol.24, No.7, pp. 44-51, 2006.

[2] Uwe Hansmann, Riku Mettala, Apratim Purakayastha, Peter Thompson, SyncML Synchronizing and Managing Your Mobile Data, p.265, PRENTICE HALL PTR, New Jersey, 2003.

[3] SCTS Server, www.funambol.com/opensource

[4] SyncML Device Management Protocol, OMA(www.openmobileallance.org), 2002.

[5] SyncML Device Management Tree and Description, OMA(www.openmobileallance.org), 2002.

[6] SyncML Device Management Standardised object, OMA(www.openmobileallance.org), 2002.

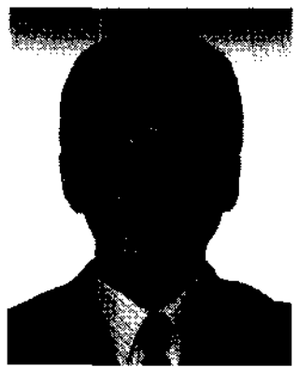
[7] S.J. Kim and H.N. Kim, "The development trend of mobile platform and WIPI," Korean Information Science Society, Vol.24, No.7, pp. 31-37, 2006.

[8] J.W. Lee, M.K. Choi and T.H. Kim. "A technical movement of mobile platform," Korean Content Society, Vol.3, No.1, pp. 452-457, 2005.

[9] D.J. Jang, H.T. Ju, K.H. Park, B.H Ha, M.C. Lee, S.C. Bae, "Design of ThinkSync DM based on SyncML Device Management," The 3rd APIS, pp. 569-574, 2004.

[10] J.G. Park, K.H. Park, D.J. Jang, M.S. Jang, and J.J. Woo, "Design of DM Agent based on the WIPI," Journal of Society of Mobile Technology, Vol.4, No.1, pp. 61-67, 2007.

[11] SyncML Toolkit, http://sourceforge.net/projects/syncml-ctoolkit


박 주 건

2006년 2월 계명대학교 컴퓨터공학 학사. 2006년 3월~현재 계명대학교 컴퓨터공학 석사. 관심분야는 Mobile Device Management, Mobile Data Synchronization, 모바일 소프트웨어

박 기 현

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제 14 권 제 2 호 참조

장 대 진

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제 14 권 제 2 호 참조



장 명 숙

경북대학교 수학과 학사. 미국 밴드빌트 대학교 컴퓨터공학 석사. 관심분야는 병렬처리, 인공지능, 모바일통신 소프트웨어