

# WIPI 기반의 OMA Client Provisioning Library 설계 및 구현

서상환

고려대학교 컴퓨터정보통신대학원

satanjr@korea.ac.kr

## Design and Implementation of OMA Client Provisioning Library on WIPI Platform

SangHwan Seo,

Graduate School of Computer and Information Technology, Korea University

### 요 약

모바일 단말과 같은 임베디드 시스템은 배포 후 버그를 수정 재 배포하기가 힘들다. 이러한 문제점을 해결 하기 위하여 OMA(Open Mobile Alliance)에서는 OMA Client Provisioning 기반의 OMA Device Management 규격을 정의하고 있다. 본 논문에서는 기반이 되는 OMA Client Provisioning 규격을 지원 하기 위한 Provisioning Library 와 Provisioning Agent를 설계 및 구현 하였다. 본 논문에서 구현된 Provisioning Library는 Standard C를 이용하였으며, Bootstrap 및 Continuous Provisioning을 지원하기 위하여 설계된 Provisioning Agent는 국내 단말 3사가 채택하고 있는 표준 플랫폼인 WIPI 기반으로 제작 되어 단말간 이식성이 높도록 하였다. 본 논문에서 구현한 시스템은 사용자가 새로 추가된 정보 서비스를 받기 위하여 네트워크 정보를 업그레이드를 원할 경우 서비스 센터에 직접 방문하여 단말을 업그레이드 받을 필요 없이 OTA를 통하여 네트워크 정보를 업그레이드 할 수 있는 환경을 제공한다.

### 1. 서 론

모바일 단말과 같은 임베디드 시스템은 일반 PC 프로그램과 같이 버그가 발생할 경우 버그 패치가 배포 된다거나 자동 업데이트를 이용하여 버그 수정이 이루어 지는 것과 달리 버그를 수정하여도 이를 재 배포 하기 힘들다. 이러한 모바일 단말의 문제점 및 전반적인 관리를 위하여 OMA(Open Mobile Alliance)에서는 현재 OMA Client Provisioning 규격을 기반으로 OMA Device Management 규격을 만들고 있는 중이다. 본 논문에서는 OMA Device Management의 기반이 되는 OMA Client Provisioning만을 다루도록 한다. 모바일 단말에서의 OMA Client Provisioning은 무선 네트워크상에서 동작하기 위해 필요한 정보를 클라이언트에게 제공하는 방법을 규정한 것이다. 모바일 단말이 OMA Client Provisioning을 지원하게 되면 얻을 수 있는 가장 큰 이점은 다음과 같다. 우선 신규 서비스가 추가가 되어 새로운 네트워크 설정이 필요하게 되어도 사용자가 별도로 서비스센터 등을 방문하여 클라이언트 단말의 바이너리를 업데이트 받아야 하는 문제점을 해결해준다. 이는 가입된 모든 클라이언트를 이동통신사의 필요에 의해 이미 개통된 사용자를 서비스센터로 유도하지 않고도 네트워크 정보를 일괄적으로 업데이트 시킬 수 있다는 것을 의미 한다. 또, 클라이언트에서 사용되는 네트워크 자원 정보를 한곳으로 통일시켜 자원의 관리 및 인증 받지 못한 사용자의 불법적인 네트워크 사용을

막을 수 있다. 클라이언트의 네트워크를 사용하려는 모든 어플리케이션은 이동통신사가 제공하는 네트워크 정보만을 Client Provisioning 라이브러리에서 제공 받아서 사용해야 하므로 악의적 사용자를 사전에 차단할 수 있게 된다. 본 논문은 OMA Client Provisioning v1.1를 기반으로 OMA Client Provisioning 라이브러리를 설계 및 구현 하였다. 본 논문의 구성은 제2장에서 Client Provisioning, WBXML 및 Provisioning Agent에 관하여 알아보고, 제3장에서 라이브러리와 Agent의 설계를, 제4장에서는 라이브러리와 Agent의 구현에 관하여 논한다. 그리고 제5장에서는 모바일 환경에서 고려해야 할 점 및 문제점에 관하여 논하며 결론을 맺는다.

### 2. 관련 연구

모바일 단말에 적용할 Client Provisioning 라이브러리와 Provisioning Agent 설계를 위해서는 다음과 같은 사항이 고려되어야 한다. 첫째 Client Provisioning 은 OMA(Open Mobile Alliance) Client Provisioning v1.1를 기반으로 설계 하여 표준을 따르도록 한다. 둘째, Provisioning 라이브러리는 클라이언트에서 사용되는 어플리케이션의 모든 네트워크 정보를 가지고 있어야 하고 다양한 단말에 적용가능 해야 하므로 이식성을 고려해야 한다. 셋째, Provisioning Document 는 WBXML 형태로 전달 되므로 WBXML 문서를 디코딩 할 수 있어야 한다. 넷째, Provisioning Agent는 국내

이동통신사가 공통으로 채택하고 있는 WIPi 플랫폼을 기반으로 설계하여 단말 별 중복 개발이 없도록 한다. 본 장에서는 위의 사항을 고려한 Client Provisioning 라이브러리 및 Provisioning Agent에 적용된 기반 기술을 설명한다.

### 2.1 OMA Client Provisioning

OMA에서 제안하고 있는 Client Provisioning에서는 크게 BootStrap과 Continuous Provisioning으로 구분할 수 있다. 첫째 BootStrap 과정은 인증 받지 못한 모바일 단말이 무선 네트워크를 사용하기 위하여 이동통신사로부터 인증을 받는 과정을 말한다.[1] 이때 모바일 단말은 이동통신사로부터 인증 받은 서버(TPS, Trusted Provisioning Server)에 접근하여 인증을 받은 후 사용할 무선 네트워크 정보를 가지고 있는 Provisioning 데이터를 내려 받게 된다. 둘째 Continuous Provisioning은 이미 무선 네트워크에 참여하고 있는 단말이 수정 및 추가되는 데이터 정보를 받는 과정을 말한다.[2]

### 2.2 WBXML

WBXML은 WAP Binary XML의 약자 이며, WBXML은 데이터 크기가 중요성을 가지는 협대역 네트워크를 통해 XML 데이터를 전송할 때 효율을 높여 준다.[3] 클라이언트의 네트워크 정보를 가지는 Provisioning Document는 WBXML 형태로 인코딩 되어 있어 대역폭이 좁은 WAP 환경에서 전송될 데이터의 크기를 줄여준다. 잘 알려져 있으며 사용 빈도가 높은 문자열은 WBXML로 변환 과정에서 단일 Hex 문자로 치환이 된다. 예를 들어 Provisioning Document에서 “Wap-provisioningdoc” 이라는 문자열은 “05”로 치환된다.[4]

### 2.3 WIPi

이 논문에서 제안한 Agent는 국내 모바일 표준 플랫폼 규격인 WIPi 기반의 어플리케이션 이다. WIPi는 SKT, KTF, LGT등의 국내 이동 통신사에서 공통으로 사용하고 있는 플랫폼으로 정보통신부가 추진하고 있는 무선인터넷 플랫폼의 표준이다. 모바일 콘텐츠를 제작하는 콘텐츠 제공자(CP)들은 서로 다른 무선인터넷 플랫폼으로 인한 중복 개발의 부담을 안고 있다. WIPi는 이러한 중복 개발의 부담을 덜고자 표준화된 모바일 플랫폼이다.

### 2.4 Provisioning Agent

모바일 단말은 Bootstrapping 및 Continuous

Provisioning을 진행하기 위하여 이동통신사에서 인증한 서버에 접속을 하기 위하여 서버와 통신을 하는 어떤 방법이 필요하게 된다. 이 논문에서는 이 목적을 위하여 국내 이동통신사가 공통으로 채택하고 있는 WIPi 플랫폼 기반의 Agent를 구현하였다. Agent가 하는 일은 첫째, TPS에 접속하여 모바일 단말의 인증 절차를 진행한다. 둘째, 서버에서 전달에 주는 Provisioning Data를 다운 받아 단말에 저장을 시도한다. 셋째, Continuous Provisioning을 위하여 서버에 접속하여 수정 및 추가된 Provisioning Data를 다운로드 한다.

## 3. 시스템 설계

Client Provisioning은 WBXML로 만들어진 Provisioning Document를 디코딩하여 네트워크정보를 단말의 메모리에 관리하는 Provisioning 라이브러리와 Boot Strap 및 Continuous Provisioning을 지원하기 위한 Provisioning Agent 모듈로 나누어 진다. Provisioning 라이브러리는 단말에 설치되어 있는 모든 어플리케이션이 사용하고 있는 네트워크 정보를 한곳에서 관리 하는 역할을 담당하므로 단말의 라이브러리 형태로 제공되며, Provisioning Agent는 wipi 플랫폼 위에서 작동하는 wipi 어플리케이션이다.

### 3.1 Client Provisioning 라이브러리

Client Provisioning 라이브러리의 구성은 그림1과 같다. Client Provisioning 라이브러리는 크게 세가지 모듈로 구성되어 있다. 전달 받은 WBXML 형태의 Provisioning Document를 디코딩 하여 XML 형태로 바꾸어주는 WBXML 디코더 모듈과 디코딩된 XML 데이터를 Parsing 하는 XML Parser 모듈 그리고 Parsing된 데이터를 테이블로 관리하는 Provisioning Table 관리 모듈을 가진다.

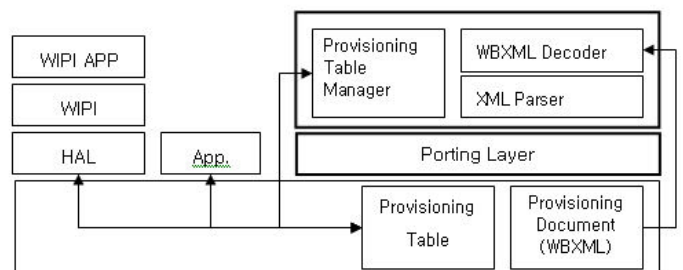


그림 1. Provisioning Library Architecture

### 3.2 Provisioning Agent

Provisioning Agent의 구성은 그림 2와 같다. Provisioning Agent는 OMA Client Provisioning v1.1에서 제시하고 있는 BootStrap 과정과 Continuous

Provisioning 을 지원하기 위한 WIPI 어플리케이션 이다. Agent는 신뢰된 서버로부터 단말을 인증 받고 해당 Provisioning Document를 내려 받아 단말에 저장 하는 역할을 한다.

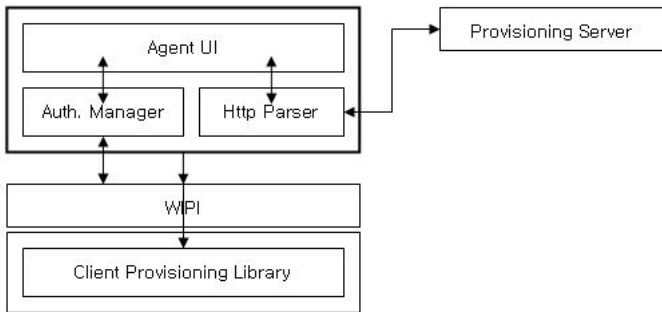


그림 2. Provisioning Agent Architecture

4. 구현

4.1 WBXML Decoder 와 XML Parser의 구현

본 논문에서는 Client Provisioning 라이브러리의 핵심 부분인 WBXML Decoder 및 XML Parser를 구현 하였다. 그리고 WBXML Decoder는 Provisioning 데이터만 디코딩 할 수 있도록 구현하였다.

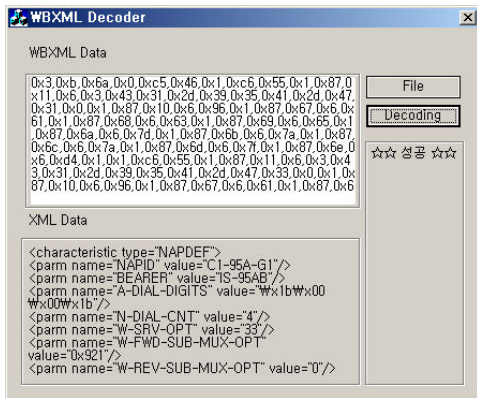


그림 3. WBXML Decoding 성공

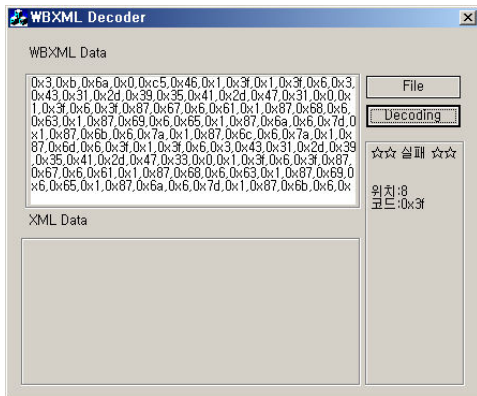


그림 4. WBXML Decoding 실패

그림 4 는 WBXML 파일을 입력한 후 디코딩을 실행하여 WBXML 데이터가 XML 데이터로 변경된 결과를 출력한 화면이다. 그림 4에서 보여주고 있듯이 WBXML Decoder는 잘못된 WBXML 파일이 입력되었을 경우 디코딩 도중 오류가 발생한 위치를 알려주도록 하여 WBXML 파일에 대한 검증을 할 수 있도록 하였다.

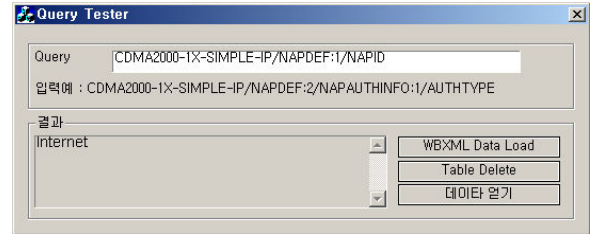


그림 5. Query Tester 실행 화면

또한 그림 5에서 보듯이 Provisioning 라이브러리는 자체 Query 규칙을 가지고 있다. 간단한 Query 규칙은 표 1에 설명되어 있다.

CDMA2000-1X-SIMPLE-IP/NAPDEF:2/NAPAUTHINFO:1/AUTHTYPE	
CDMA2000-1X-SIMPLE-IP	필요한 데이터를 얻어올 Barer Type
NAPDEF:2	선택한 Barer에 존재하는 2번째 NAPDEF 영역을 선택
NAPAUTHINFO:1	선택된 NAPDEF 아래에 있는 첫 번째 NAPAUTHINFO 정보
AUTHTYPE	NAPAUTHINFO 아래에 있는 AUTHTYPE의 값을 얻어 옴.
- 찾으려는 정보는 우선되는 상위 영역의 내부에서만 검색된다. - 찾으려는 정보의 순서가 없다면 첫 번째 정보를 얻어 온다	

표 1. Query 규칙

이렇게 공통된 Query를 사용하게 될 경우 모든 어플리케이션은 동일한 방법으로 데이터를 가져 갈 수 있으며 새로운 네트워크 정보가 추가 되어도 해당 정보를 얻어오기 위하여 새로 API를 추가 할 필요가 없게 된다.

4.2 Provisioning Agent의 구현

클라이언트가 처음 부팅이 되면 우선 단말에 네트워크 정보가 있는 Provisioning Document를 찾게 된다. 만약 Provisioning Document가 없다면 클라이언트는 Provisioning Agent를 구동하여 인증절차를 거친 후 Provisioning Document를 전송

받게 된다. 그림 6는 지정된 서버에 접속하여 WBXML 데이터를 받도록 만들어진 Provisioning Agent의 실행화면이다. Provisioning 초기화가 완료되면 단말은 정상적으로 사용이 가능해 진다.

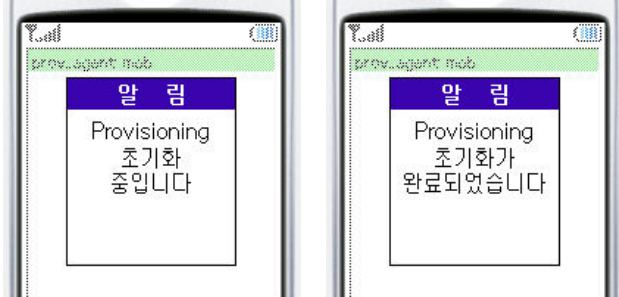


그림 6. Provisioning Agent 실행 화면

## 5. 결론 및 향후 연구

본 논문에서는 OMA Client Provisioning v1.1기반의 Provisioning 라이브러리 및 Bootstrapping 과 Continuous Provisioning을 위한 Provisioning Agent를 제안하여 정상 작동하는 것을 확인 하였다. 또한, 모바일 어플리케이션에서 Provisioning 데이터를 동일한 방법으로 얻어 오기 위하여 Query문을 제안하고 정보를 정상적으로 얻어 오는 것을 보였다. 향후 과제는 현재 구현된 WBXML Decoder 및 XML Parser는 Provisioning 데이터만 다루도록 되어 있으나, 향후 OMA Device Management를 위하여 범용 Decoder 및 Parser를 탑재 할 수 있도록 구현 하는 것이다

## 6. 참고 문헌

- [1] Open Mobile Alliance, "Provisioning Bootstrap", Open Mobile Alliance, 2002
- [2] Open Mobile Alliance "Provisioning Architecture Overview", Open Mobile Alliance, 2002
- [3] Bruce Martin, Bashar Jano, "WAP Binary XML Content Format", Wireless Application Forum, 1999
- [4] Open Mobile Alliance, "Provisioning Content", Open Mobile Alliance, 2002