

WIPI 무선 단말기를 위한 다운로드 시스템의 설계 및 구현

김민정, 박현주, 김민석, 임종태
SK Telecom 플랫폼연구원
e-mail : {kmj, firststrun, minskim, jtihm}@sktelecom.com

The Design and Implementation of a Download System for WIPI Mobile Devices

Minjung Kim, Hyeonju Park, Minseok Kim, Jongtae Ihm
Platform R&D Center, SK Telecom

요 약

1999년부터 상용화된 무선 인터넷 서비스는 국내의 경우 이동통신 3사가 각기 독자적인 통신 기법 및 서비스 체제를 가지고 있었다. 이로 인하여 서로 다른 이동통신사의 무선 단말기 플랫폼들은 상호 호환성(interoperability)이 없었다. 이러한 비호환성은 단말기 제조업체나 콘텐츠 제작 업체에게 기술적인 어려움은 물론이고 비용적인 측면에서 여러 가지 문제점을 낳았다. 이러한 문제점을 해결하고자 2002년부터 WIPI(Wireless Internet Platform for Interoperability)라는 표준화된 무선 인터넷 플랫폼의 개발이 시작되어 현재는 WIPI 2.0을 탑재한 무선 단말기가 출시된 상황이다. 본 논문에서는 WIPI 플랫폼이 탑재된 무선 단말기가 유료화된 콘텐츠를 다운로드 하여 사용할 수 있도록 해주는 다운로드 시스템의 설계 및 구현에 대해서 기술한다. 이를 위해 콘텐츠 다운로드 프로토콜, 최적화된 콘텐츠의 결정기법, 그리고 COD(Compile On Demand)를 이용한 네이티브 코드 적용 기법에 대해 설명한다.

1. 서론

1990년대 말 이후 무선 인터넷 서비스가 시작되면서, 초기에는 무선 단말기(이하 단말기) 내의 WAP 프로토콜을 이용한 브라우저를 기반으로 간단한 형태의 무선 인터넷 서비스가 제공되었다. 이후 서비스가 정착되면서 단말기에 가상머신을 탑재하고 콘텐츠를 다운로드 한 후 실행하는 형태의 비즈니스 모델이 우세를 유지하게 되었다. 이동통신 3사에서는 서로 다른 규격으로 다운로드 서비스를 제공하고 있었고, 이에 대한 문제점을 해결하기 위하여 무선 인터넷 플랫폼을 WIPI라는 단일 환경으로 표준화 하기에 이르렀다 [1, 7].

국내 무선 인터넷 환경은 이동통신 사업자들이 WAP이나 ME 방식의 서로 다른 무선 응용 프로토콜을 채택하고 있었으며, 응용프로그램의 실행 환경인 무선 인터넷 플랫폼과 콘텐츠 개발자를 위한 개발환경이 상이하여 무선 인터넷 활성화의 걸림돌이 되고

있었다. 특히 이동통신 사업자들이 본격적인 무선 인터넷 서비스 시장을 형성하기 위하여 CDMA2000 1X나 W-CDMA와 같은 네트워크 인프라의 도입을 진행하면서 적극적인 무선 인터넷 시장의 표준화 추진이 필요하게 되었다.

국내의 경우 SKT의 GNEX, SKVM과 WITOP, KTF의 BREW와 MAP, LGT의 KVM 등 다수의 무선 인터넷 플랫폼에서 콘텐츠들이 서비스 되고 있었다. 이에 따라 콘텐츠 개발자들은 하나의 콘텐츠를 다수의 무선 인터넷 플랫폼용으로 개발해야 하는 부담이 증가하였으며, 단말기 제조업체 또한 무선 인터넷 플랫폼에 의존된 시스템을 제각각 만들어야 하는 불편함이 있었다 [7].

이러한 문제점을 해결하기 위하여 2001년 5월에 정보통신부, 이동통신 3사, 한국전자통신연구원, 정보통신기술협회가 주관하는 한국무선 인터넷표준화포럼이 창립되었고, 한국무선 인터넷표준화포럼에서 2002

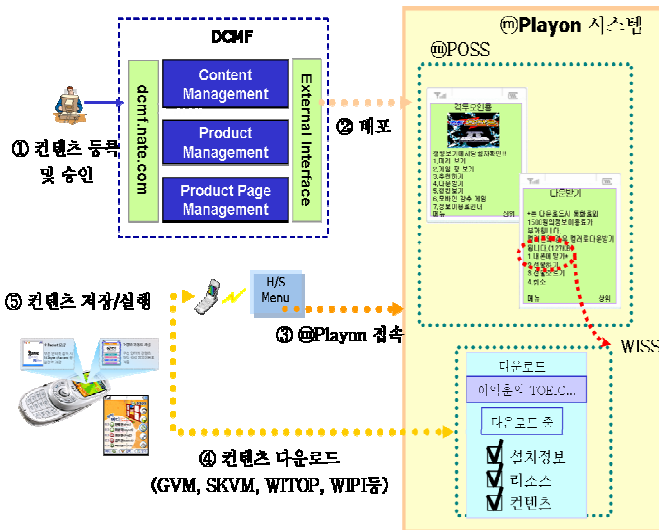
년 3 월에 WIPI 라고 명명된 무선 인터넷 표준 플랫폼 규격 V1.0 을 완성하였다. 무선 인터넷 표준 플랫폼 규격에서는 HAL(Handset Adaptation Layer) 규격, 응용 프로그래밍 인터페이스 규격, 플랫폼의 주요 기능 규격과 같은 내용을 기술하고 있다.

WIPI 규격은 이동통신 단말기에 탑재되어 응용 프로그램을 수행할 수 있는 환경인 모바일 플랫폼 규격을 정의한다. 이 규격을 만족하는 플랫폼은 콘텐츠 개발자에게는 플랫폼간의 콘텐츠 호환성을 보장하고, 단말기 제조업체에게는 플랫폼 이식의 용이성을 제공하며, 일반 이용자에게는 다양하고 풍부한 콘텐츠 서비스의 제공을 목적으로 하고 있다. 또한 이동통신사에는 다양한 콘텐츠의 빠른 공급과 폭 넓은 사용자층의 확대를 위한 기반이 마련된다.

본 논문에서는 이러한 WIPI 플랫폼을 탑재한 단말기를 이용하여 유료 콘텐츠를 매우 빠르고 안정적으로 다운로드 시킬 수 있는 SKT 의 다운로드 시스템에 대해서 기술한다. 이 시스템은 현재 @Playon 이라고 하는 서비스에 이용되고 있다.

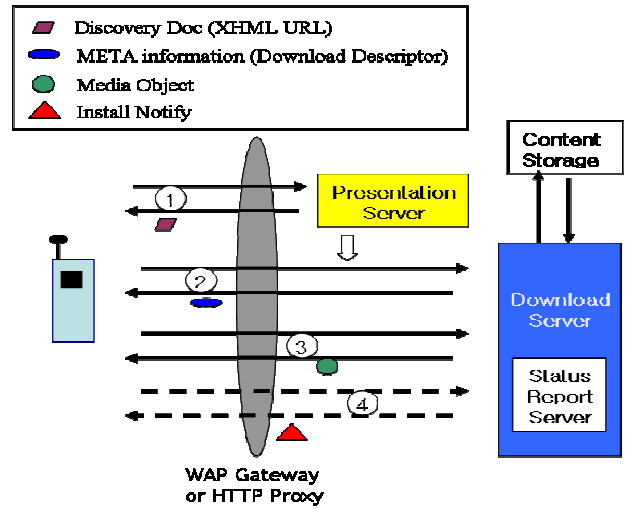
2. 다운로드 서비스 개요

유료 콘텐츠는 2 단계에 걸쳐 시스템 내에 등록된다. 우선 콘텐츠 제공자가 정해진 등록 절차에 따라 DCMF(Digital Contents Management Framework)라고 하는 SKT 의 콘텐츠 등록 서버에 서비스 될 콘텐츠를 등록한다. 이후 콘텐츠는 SKT 사업부서의 승인을 거친 뒤 두 번째 단계로 미들웨어를 통해 콘텐츠 등록 서버에서 서비스 서버인 @Playon 시스템으로 배포된다. 이 과정은 (그림 1)의 1, 2 단계에서 설명하고 있다. 다음 단계로 (그림 1)의 3 과 4 에서와 같이 사용자의 콘텐츠 네비게이션 후 다운로드 요청이 있을 때 단말기로 해당 콘텐츠를 다운로드 해 주게 된다.



(그림 1) 서비스 시나리오

콘텐츠의 종류는 이동통신사마다 차이가 있으나, SKT 의 경우에는 C 기반의 GNEX 와 WIPI-C, Java 기반의 SKVM 과 WIPI-Java 등이 있다.



(그림 2) 서버와 단말기간 다운로드 프로토콜

(그림 2)는 서버와 단말기간의 다운로드 프로토콜에 대하여 설명한다. 이용자가 SKT 의 무선 인터넷 서비스인 NATE WAP 서비스를 진행하다가(flow1) 원하는 콘텐츠 다운로드를 요청하게 되면, WIPI 무선 단말기는 먼저 해당 콘텐츠에 대한 각종 설치정보를 가지고 있는 DD(Download Descriptor)를 다운로드 서버로 요청하게 된다. 해당 DD 안에는 실제 다운로드 받을 콘텐츠의 URL 및 업그레이드 시 접속할 URL, 콘텐츠의 실행횟수 등 관련 설치정보가 담겨져 있다. 다운로드 서버는 단말기로부터 DD 다운로드 요청을 받고 나면 콘텐츠 프로비저닝을 수행하는데, 이는 콘텐츠 등록 서버에 등록되어 있는 콘텐츠 중에서 단말기의 사양에 최적화된 콘텐츠를 찾는 과정을 말한다. 이러한 콘텐츠 프로비저닝 절차가 끝나면 결정된 DD 를 단말기로 다운로드 한다(flow2). 단말기는 다운로드 받은 DD 정보를 파싱하여 실제로 다운로드 받을 실행파일 URL 정보를 추출한 후 해당 URL 로 실행파일의 다운로드를 요청한다. 다운로드 서버는 요청된 콘텐츠를 HTTP 프로토콜을 이용하여 WIPI 단말기의 WAM(WIPI Application Manager) 영역으로 다운로드를 한다(flow3). 단말기는 해당 콘텐츠를 서버로부터 다운로드 완료하게 되었을 때 서버로 설치완료 확인을 보내게 된다 [3, 5]. 설치완료 확인을 받은 서버는 그에 대한 응답을 단말기로 보내고(flow4) 비로소 정보이용료 과금을 수행한다.

3. 시스템 구현 사항

3.1 WIPI 단말기 구조

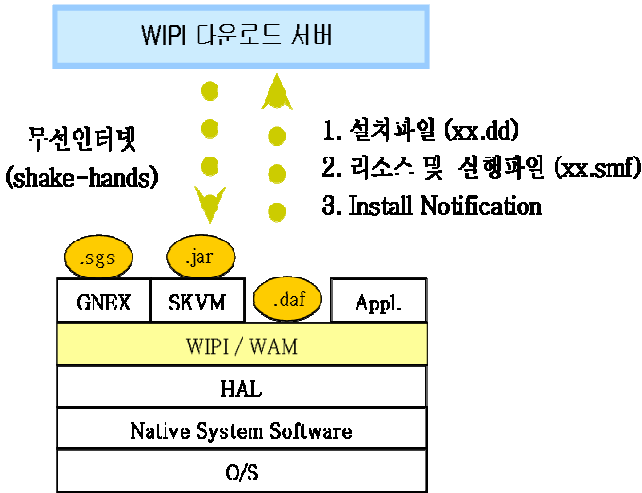
WIPI 단말기의 구조 및 다운로드 서버와의 관계는 (그림 3)과 같다. WIPI 단말기의 O/S 위에는 HAL 이 올라가게 되는데, HAL 은 서로 다른 제조사의 O/S 에 WIPI 를 쉽게 탑재 할 수 있도록 하는 중간 레이어이다. WIPI 다운로드 서버가 직접 통신하는 모듈은 단말기 구조 중 WAM 영역이며 WIPI 위에는 가상머신인 GNEX 와 SKVM 이 탑재된다. GNEX 는 단말기가 출시

될 때 탑재되는 것이 아니라 처음 GNEX 콘텐츠를 다운로드 하고자 할 때 무선인터넷 서비스를 통하여 콘텐츠와 함께 다운로드 된다. 실제로 실행되는 콘텐츠들은 가상머신인 GNEX 와 SKVM 위에서 동작되며, 콘텐츠의 종류로는 C 계열 콘텐츠로 .sgs 의 확장자를 가지는 GNEX 콘텐츠와, Java 계열 콘텐츠로 .jar 의 확장자를 가지는 SKVM 콘텐츠가 있다. .daf 의 확장자를 가지는 WIPI-C 콘텐츠는 가상머신이 아닌 WIPI 플랫폼 위에서 직접 실행된다.

```

dlf_dd_url="http://210.91.42.60:5000/parkyj/WIPIC/00001_0000100010.dlf"/>
</dlf_info>
</add_info>
</dd>
    
```

(그림 4) DD 파일의 예시



(그림 3) WIPI 단말기 구조

3.2 DD 파일 구조

DD 파일은 다운로드 받을 콘텐츠에 대한 설치 정보이다. 콘텐츠 다운로드를 담당하는 다운로드 서버는 콘텐츠 다운로드 전에 설치파일인 DD 를 먼저 다운로드 해주어야 한다. 단말기는 DD 수령 후 DD 파서 API 를 사용하여 DD 버퍼를 전송하며 DD 파서는 구조체에 해당 정보를 넣어 준다. C 계열 콘텐츠인 GNEX 와 WIPI-C 등의 DD 는 XML 구조로 되어 있으며, 미디어 오브젝트 정보, 추가 정보, DRM(Digital Rights Management) 정보 등으로 구성된다 [6]. Java 계열 콘텐츠인 SKVM 과 WIPI-JAVA 의 DD 는 XML 형태가 아니라 표준 Java 에서 사용하는 JAD(Java Application Descriptor) 파일 형식을 따른다 [2].

```

<?xml version="1.0" encoding="euc-kr"?>
<!DOCTYPE media SYSTEM "media.DTD">
<dd dd_version="1.0">
  <mo_info id="1000000001" version="1.0" size="209038">
    down_url="http://210.91.42.60:5000/parkyj/WIPIC/00001_1000000001.daf">
      <title data="WIPIC Test"/>
      <vender data="SK Telecom"/>
      <type data="application/x-wipi-app"/>
      <update_url data="http://210.91.42.60:5000/update"/>
    </mo_info>
    <add_info>
      <dlf_info dlf_num="1">
        <dlf id="0000100010" version="1.0" size="4096"
    
```

3.3 콘텐츠 프로비저닝 절차

다운로드 서버가 DD 를 다운로드 할 때에는 콘텐츠 프로비저닝 이라는 절차를 거친다. 프로비저닝이란 콘텐츠 등록서버에 등록되어 있는 콘텐츠 중에서 단말기 사양에 가장 적합한 콘텐츠를 다운로드 하는 절차를 말한다.

프로비저닝을 수행하기 위하여 필요한 단말기 정보의 취득은 OMA(Open Mobile Alliance) 표준 규격으로 정의되어 있는 UA Profile 규격을 따른다 [4]. 이 규격에 따라 구현된 UA Profile 서버에는 각종 단말기 정보들이 저장되고 관리된다. UA Profile 서버에서 관리되는 단말기 정보 중에서 프로비저닝 시에 주로 이용되는 주요 항목으로는 GIGA(Graphic Instruction Graphic Acceleration) 여부, 가상머신 버전, 해상도, 사운드 플리, LCD 크기, Heap 크기 등이 있다.

(표 1) 다운로드 받을 콘텐츠의 목록 예

Index	VM major	VM minor	Color	Sound	LCD Size
1	1	0	2Color	None	120x110
2	1	0	4Color	Buzzer	120x110
3	1	1	4Color	EvrC	120x110
4	1	0	16Color	MA1(4Poly), Buzzer	120x130
5	1	1	16Color	MA2(16Poly), Buzzer	120x130
6	1	0	256Color	Q4(4Poly)	120x110
7	1	1	256Color	Q15(16Poly)	120x110
8	1	0	4096Color	MA2(16Poly), Buzzer	120x140
9	1	1	4096Color	MA3(40Poly), MA2(16Poly)	120x140
10	1	0	32K 64KColor	MA3, MA2, MA1	120x110

(표 2) 프로비저닝 결과 선택된 콘텐츠 예

Index	VM major	VM minor	Color	Sound	LCD	Index
1	1	0	2	None	120x110	1
2	1	1	4	EVRC	120x130	3
3	1	0	256	MA1(4Poly), Buzzer	120x140	4
4	1	1	256	MA2(16Poly), Buzzer	120x140	5
5	1	1	256	Q15(16Poly)	120x140	7
6	1	1	4096	MA3(40Poly), MA2(16Poly)	120x140	9
7	1	1	4096	MA3(40Poly), MA2(16Poly)	120x130	5
8	1	0	4096	MA2(16Poly), Buzzer	120x130	4
9	1	0	65K	MA3(40Poly), MA2(16Poly)	120x130	10
10	1	1	65K	MA3(40Poly), MA2(16Poly)	120x130	5

프로비저닝을 수행하기 위하여 필요한 단말기 정보의 취득은 OMA(Open Mobile Alliance) 표준 규격으로 정의되어 있는 UA Profile 규격을 따른다 [4]. 이 규격에 따라 구현된 UA Profile 서버에는 각종 단말기 정보들이 저장되고 관리된다. UA Profile 서버에서 관리되는 단말기 정보 중에서 프로비저닝 시에 주로 이용되는 주요 항목으로는 GIGA(Graphic Instruction Graphic

Acceleration) 여부, 가상머신 버전, 해상도, 사운드 폴리, LCD 크기, Heap 크기 등이 있다.

앞 페이지의 (표 1)과 (표 2)는 다운로드를 요청한 사용자의 단말기 사양에 따라 콘텐츠 등록서버에 등록된 콘텐츠 중에서 어떤 콘텐츠를 프로비저닝 하여 다운로드 해야 하는지를 보여주는 예이다.

콘텐츠를 서비스 하기 위해서는, 다양한 사양의 단말기를 위한 여러 종류의 콘텐츠를 콘텐츠 등록서버에 미리 등록해 놓아야 한다. (표 1)은 가상머신의 상위와 하위 버전, 해상도, 사운드 폴리, LCD 크기의 종류에 따라 색인 1 부터 10 까지의 10 개의 콘텐츠가 등록되어 있는 모습이다. (표 2)중 색인 5 번의 사양을 가진 단말기가 콘텐츠 다운로드 요청을 하게 되면, 색인 5 번의 단말기 사양보다 같거나 낮은 사양의 콘텐츠를 다운로드 하기 위하여 차례로 10 개의 콘텐츠 검색을 거치게 된다. 콘텐츠 검색 결과로 여러 개의 콘텐츠가 추출된 경우에는 그 중에서 가장 사양이 높은 콘텐츠를 다운로드 하게 된다. 이러한 프로비저닝의 절차에 따라 (표 2)의 색인 5 번 단말기는 (표 1)의 10 가지 콘텐츠 중에서 색인 7 번의 콘텐츠를 다운로드 하게 된다.

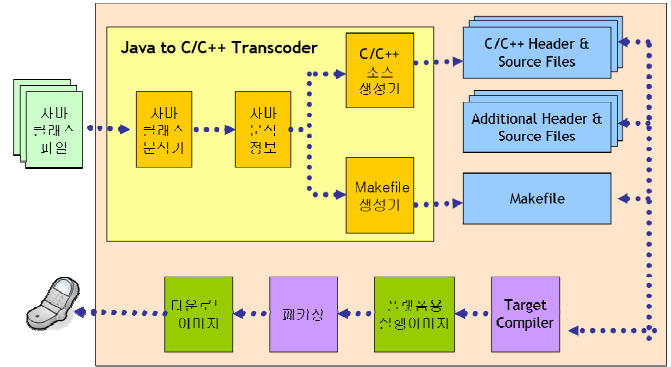
3.4 COD(Compile On Demand)

가상머신 기반의 플랫폼은 가상머신을 이용하여 중간코드를 실행과정에 번역하는 부담이 있어 CPU 성능과 자원이 제한되는 휴대 단말기에서 상대적으로 느린 속도를 나타낸다. 그러나 상대적으로 시스템의 안정성과 완벽한 보안을 제공하는 장점을 가지고 있다. 반면 네이티브 바이너리 기반의 플랫폼은 가상머신 기술과 달리 탑재된 CPU의 성능을 최대로 활용할 수 있으므로 매우 빠른 실행속도를 제공한다. 그러나 개발언어로 C 나 C++을 사용하고 있어서 프로그램 개발자에 의한 악의적인 메모리 접근 등으로 시스템 안정성을 해칠 수 있는 가능성을 가지고 있다.

WIPI 는 이런 두 플랫폼의 장점을 모두 가지도록 설계되었다. 즉 자바 언어의 장점을 수용함과 동시에 네이티브 바이너리의 실행성능을 갖고자 한 것이다. C 나 C++과 자바를 모두 지원하여 프로그램을 개발하도록 하였지만, 실제 단말기에 다운로드 되는 형태는 네이티브 바이너리로 규정하여 실행성능을 보장하도록 하였다. 이때 C 나 C++은 네이티브 바이너리의 생성에 별다른 문제가 없지만, 가상머신에서 실행되는 바이트 코드를 생성하는 자바의 경우에는 바이트 코드를 단말기에서 실행할 수 있는 바이너리로 변환하는 단계가 필요하다. 이에 따라 서버형 컴파일러인 COD 시스템이 개발 되었다. (그림 5) 는 이러한 COD 시스템의 동작 절차를 보인다.

COD 시스템 중에서 트랜스코더라는 모듈을 거쳐 자바 클래스 파일은 C 소스로 변환이 되고, 트랜스코더의 결과물인 C 소스는 컴파일러 모듈을 거쳐 실행 이미지로 컴파일 된다. 해당 이미지는 이동통신사에서 서비스 될 수 있도록 패키징 절차를 거쳐 최종 결과

물인 다운로드 이미지가 된다.



(그림 5) COD 시스템의 동작 절차

콘텐츠 개발자가 콘텐츠 등록서버에 WIPI-Java 바이트코드를 등록하면 COD 시스템은 단말기에 탑재 가능한 바이너리 이미지를 생성하여 관리하고, 사용자가 다운로드를 요구할 때 이미 컴파일 된 실행 바이너리를 선택하여 다운로드 하게 하는 것이다. 따라서 WIPI 플랫폼은 바이트 검증 부분이 필요 없게 되고 대신 COD 서버에서 이 기능을 수행하게 된다.

4. 결론 및 향후 과제

본 논문에서는 무선 인터넷 표준화 플랫폼인 WIPI가 탑재된 단말기를 위한 다운로드 시스템의 설계 및 구현에 대하여 기술하였다. 이 중에서 콘텐츠 다운로드 프로토콜, 최적화된 콘텐츠의 결정기법, 그리고 COD를 이용한 네이티브 코드 적용 기법에 대하여 상술하였다. 향후 연구 과제는 프로비저닝 로직을 룰 기반으로 재구축하는 것이다. 시스템에 산재한 프로비저닝 업무 로직을 소스코드에서 분리하여 룰로 독립 관리한다면 소스 코드의 알고리즘으로 표현된 업무로직을 자동화 할 수 있을 것이다. 또한 프로그래밍 언어에 의존하지 않고 업무용어 기반의 서술적 방식으로 룰을 변경하는 것도 가능할 것이다. 이를 통해 보다 다양화 되는 서비스와 단말기 종류에 따라 늘어나는 업무부담을 줄일 수 있을 것이라 판단한다.

참고문헌

- [1] "Generic Content Download Over The Air Specification", <http://www.openmobilealliance.org>
- [2] "Over The Air User Initiated Provisioning Specification (MIDP2.0 OTA)", <http://java.sun.com>
- [3] "HTTP Authentication: Basic and Digest Access Authentication", <http://www.ietf.org/rfc/rfc2617.txt>
- [4] "User Agent Profile 1.1", <http://www.openmobilealliance.org>
- [5] "Hypertext Transfer Protocol HTTP/1.1", RFC2616, R. Fielding, et al., June 1999.
- [6] "Extensible Markup Language (XML) 1.0 (Second Edition)", WWW Consortium Recommendation, <http://www.w3.org/TR/2000/REC-xml-20001006>
- [7] 한국 무선인터넷 표준화 포럼 <http://www.kwisforum.org>