

자바플랫폼을 기반으로 하는 마이크로브라우저설계

최 원 호*, 황 옥 철*, 김도우*, 정 민 수*

*경남대학교 컴퓨터공학과

e-mail : wonho@hawk.com.kyungnam.ac.kr

Design of MicroBrowser Based on Java Platform

Won-Ho Choi*, Uk-chul Hwang*, Do-woo, Kim*, Min-Soo Jung*

*Dept. of Computer Engineering, Kyungnam University

요 약

향후 IMT2000 에서도 사용가능한 자바 플랫폼을 기반으로 하는 마이크로브라우저를 설계하기 위해서는 썬마이크로시스템즈에서 제공하는 J2ME 의 기술인 CLDC 와 이를 좀 더 구체적으로 하는 MIDP 를 이용해서 플랫폼을 설계한다. 그리고 무선 인터넷 환경에 적합한 WAP 를 마이크로브라우저로 사용한다. MIDP 를 사용하기 위해서는 JAM 을 이용해 애플리케이션을 관리할 수 있다.

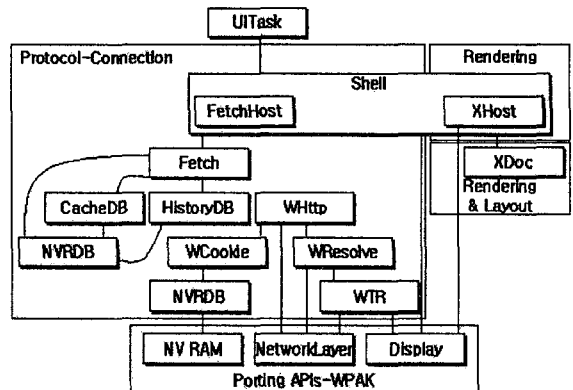
1. 서 론

우리는 지금 메인프레임의 시대에서 PC 의 시대를 지나, 어디에나 컴퓨터가 존재하는 컴퓨팅의 시대, 포스트 PC 의 시대로 가고 있다. 그리고 그 중심에는 모바일이라는 패러다임이 존재한다. 양방향 페이지, 셀룰러폰, 무선기능이 탑재된 PDA 등으로 대표되는 다양한 무선단말기들은 기존의 데스크탑과 노트북으로 대표되는 컴퓨팅 환경에 커다란 혁명을 불러올 것이다. 현재 사용되고 있는 마이크로소프트 모바일 익스플로러(MME)와 WAP 브라우저를 장착한 모바일 단말기기 시판되고 있지만 그 사용 기능들은 아직까지 미흡하다. 하지만 자바가 이러한 모바일 솔루션을 대표하는 기술로서 주목받고 있다. 세계적으로 발표되는 IT 관련 신제품의 60%가 자바를 기반으로 하고 있다고 한다. 그 중에서 인터넷 관련 제품이라는 단서를 달게 되면 그 비율은 아마 훨씬 더 높아질 것이다. 자바는 이미 시장 지배적인 기술일 뿐만 아니라, 그 변화의 속도가 엄청 빠르다. 그 중에서도 가장 변화의 속도가 빠른 분야가 무선 분야이다. 무선망 사업자와 단말기 제조 업체를 중심으로 한 무선 시장의 주요 플레이어들이 자바를 선택하는 이유는 동적인 애플리케이션의 다운로드, 크로스 플랫폼 호환성, 향상된 사용자 경험과 역동성, 비연결성, 보안 문제등에서 현재 모바일 단말기에서 이용되는 마이크로 소프트웨어 모바일 익스플로러(MME)나 유럽을 중심으로한 WAP 브라우저에 제약된 사항을 해결할 수 있다. 앞으로 상용화될 IMT2000 단말기에서도 자바 플랫폼을 이용하는 많은 단말기들을 만들어 질 것이다.

본 논문의 구성은 2 장에서는 현재 사용되고 있는 ME 와 WAP 를 중심으로 한 브라우저 기반의 모바일 단말기의 특징과 기능을 비교하였고, 3 장에서는 본 논문에서 설계할 자바 플랫폼을 위한 기술인 CLDC(Conn-ected Limited Device Configuration)과 MIDP(Mobile Infor-mation Device Profile) 에 대한 기본적인 개념들을 설명한다. 4 장에서는 결론 및 향후 연구 방향과 5 장에서는 참고문헌에 관해 서술하였다.

2. ME 와 WAP 브라우저

2.1 모바일 익스플로러(Mobile Explorer)



[그림 1] ME 컴포넌트 아키텍처

ME 는 기존 HTML 과의 호환성을 무기로 HTML

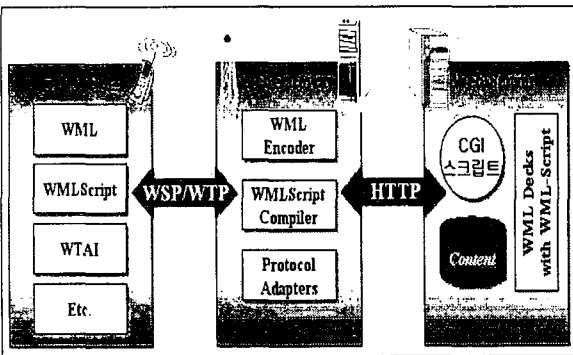
3.2의 부분집합으로 mHTML을 사용하고 1.0 버전에서는 이미지가 사용되지 않고, alt 텍스트가 표시된다. ME의 파싱(Parsing)은 텍스트는 고정 간격 폰트로 표시되고, 문자 포매팅/텍스트 마크업 태그는 지원하지 않거나 무시된다. 또 LCD 창에 4x12 문자를 표시하고 한 개의 프레임만을 표시할 수 있다.

그리고, ME의 컴포넌트 아키텍처는 [그림 1]과 같이 네비게이션 컴포넌트, 프로토콜 컴포넌트, 파싱과 레이아웃, 렌더링 컴포넌트로 구성되어 있다. 사용자로부터 입력 받은 URL을 히스토리 DB에 저장하고, 캐쉬 DB와 URL 스트링을 비교한다. 캐쉬 DB에 값이 없다면 WHttp를 실행하고 Wresolver를 수행한다. WCookie 객체를 통해 쿠키를 저장하고 복귀한 다음 WTR에 의해 구현된 큐 시스템을 통해서 http 세션이 생성된다. Shell은 UI 이벤트를 처리하고 사용자가 요청한 페이지로부터 데이터를 받으면 WHttp 객체를 폐치하고, Xdoc를 파싱과 레이아웃 컴포넌트에서 렌더링 컴포넌트에 전달하여 화면에 보여준다.

2.2 WAP(Wireless Application Protocol)브라우저

이동 통신망에서 인터넷서비스를 제공하기 위해 폰닷컴에서는 HDTP(Handheld Device Transport Portocol)와 HDML(Handheld Device Markup Language)을, 노키아에서는 TTML(Tagged Text Markup Language)를 각각 내놓았다. 또한 에릭슨에서는 ITTP(Intelligent Terminal Transfer Protocol)를 개발했다. 이에 따라 모바일폰 분야의 양대 산맥에서 각기 나름의 기술을 내놓음에 따라 서로 호환되지 않는 문제가 발생했다. 이에 에릭슨, 모토로라, 노키아, 폰닷컴이 이 공통 규격을 제정하기로 하고 WAP 포럼을 결성했고, 현재 세계 수백여 업체가 참여하고 있다.

WAP의 목적은 디지털 셀룰러 전화와 무선 터미널에서의 인터넷 서비스 및 다른 종류의 무선 통신망 기술에서 운용될 수 있는 무선 프로토콜 규격과 콘텐츠, 응용 기술을 개발하는 데 있다.



[그림 2] WAP 모델

WAP 모델은 [그림 2]와 같이 휴대단말기(클라이언트)와 인터넷 서버 사이에 WAP 프록시(Proxy)라는 WAP 게이트웨이(Gateway)를 두고 있다. WAP 게이트웨이는 WAP 프로토콜과 인터넷 TCP/IP 프로토콜을 중간에서 변화하는 기능을 한다. 즉, 모든 휴대 단말기의 인터넷 서비스 요구는 WAP 게이트웨이를 거쳐도통 되어 있고, 게이트웨이는 WAP 프로토콜에 따라 요청받은 서비스를 기존 인터넷 유선망을 통해 서버에 요청한다. 이어 게이트웨이가 인터넷 서버로부터 응답을 받고 다시 서비스를 최초 요청했던 휴대 단말기에게 WAP 프로토콜로 전송함으로써 모든 과정이 이루어진다. [그림 3]는 WAP 프로토콜의 구조와 기존 인터넷 프로토콜을 비교해 놓은 것으로 각 계층의 역할은 [표 1]과 같다.

인터넷	이더넷
HTML 자바 스크립트	Wireless Application Environment(WAE)
HTTP	Wireless Session Protocol(WSP)
TLS-SSL	Wireless Transaction Protocol(WTP)
TCP/IP UDP/IP	Wireless Transport Layer Security(WAE)
	Wireless Datagram Protocol(WAE)
	SMS USSD GPRS CSD CDPD IR-DATA

[그림 3] WAP 프로토콜 구조

[표 1] WAP 프로토콜 구조

프로토콜 계층	설명
WAE	<ul style="list-style-type: none"> ● 일반적, 다목적 응용을 개발하기 위한 응용 환경에 대한 규격 정의 ● WML, WML 스크립트 등을 정의해 휴대 단말기에 적용될 수 있는 소규모 브라우저를 개발할 수 있도록 함
WSP	<ul style="list-style-type: none"> ● HTTP1.1에 상응하는 기능을 정의 ● 장시간 활용의 세션을 정의하고, 세션 관리를 위해 서스펜드/리쥬 기능도 제공 ● 프로토콜 기능에 대한 협상도 가능하게 함
WTP	<ul style="list-style-type: none"> ● 트랜잭션 형태의 데이터 전송 기능 제공 ● 신뢰성과 비신뢰성 전송 기능을 제공하고 오류복구를 위해 재전송 기능 담당
WTLS	<ul style="list-style-type: none"> ● 인터넷의 TLS를 근간으로 작성된 보안 프로토콜

	<ul style="list-style-type: none"> ● 인증, 무결성, 기밀성 등 보안 서비스 제공
WDP	<ul style="list-style-type: none"> ● 종단간 전송을 위해 포트 어드레스를 제공 ● 인터넷 UDP 와 같은 전송 기능을 담당

그리고, WAP 브라우저는 무선 전화기, 페이지, PDA 와 같은 이동 단말기에서 입력과 표시를 목적으로 XML(eXtensible Markup Language)에 기반을 둔 WML(Wireless Markup Language)를 사용한다. WML 언어는 작은 화면과 제한된 메모리, CPU, 낮은 대역, 긴 접속 대기 시간을 갖고 있는 이동 단말기에 적합한 언어이다.

2.3 ME 와 WAP 의 비교

WAP 은 전통적인 웹 기반 프로토콜에 비해 더 효율적이다. WAP 은 무선 환경을 위해 디자인되었으므로, 최소한의 대역으로 클라이언트와 서버 사이의 데이터 교환이 가능하다. 즉, ME 솔루션은 IP 스택을 그대로 사용하므로 게이트웨이 가 필요없다. WAP 이 새로운 무선 프로토콜을 주장하는 이유는 무선의 특성상 대용량의 데이터를 주고받는 것은 그만큼 많은 비용을 발생시키므로 전송 패킷의 오버헤드를 최소화해 비용을 최소화하기 위한 차원이다.

또, 북마크의 경우 WAP 폰은 해당 통신 사업자의 서버에 저장되어 있어 북마크를 사용하고자 할 때 해당 서버에 접속해야만 가능하다. 반면 ME 폰은 북마크, 쿠키 등을 단말기에 저장할 수 있어 특정 서버나 프록시 지원이 필요없다.

3. 자바 플랫폼 상의 마이크로브라우저

본 논문에서 설계된 자바 플랫폼을 기반으로 하는 마이크로브라우저는 선마이크로시스템즈에서 개발한 소형 디지털 디바이스를 위한 J2ME(Java 2 Micro Edition)를 기반으로 해서 설계되었다. 또한 마이크로브라우저는 주로 모바일 단말기와 같이 소용량의 메모리를 사용하는 디바이스를 위해 J2ME 의 CLDC 과 이를 기반으로 하는 MIDP 를 이용해서 설계했다.

그리고, 사용될 마이크로브라우저는 무선 인터넷에 적합한 프로토콜인 WAP 를 이용하여 자바 플랫폼과 연결했다.

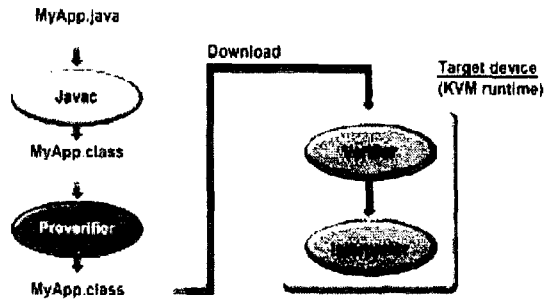
3.1 CLDC(Connecte Limited Device Configuration)

CLDC 는 128~512K 의 메모리 여유 공간과 16~32 비트 프로세서, 저전력 소모, 네트워크 연결성을 가

진 디바이스들 목표로 하고 있다.

CLDC 의 특징은 부동소수점을 지원하지 않는 오버헤드를 발생하기 때문에 지원하지 않고, finalization 과 가비지 콜렉션을 지원하지 않는다. 예외 처리를 지원하지 않지만 임베디드 환경에 의해 제한된 에러 처리만 하고, 네이티브 함수를 호출하는 JNI 는 지원하지 않고, 또한 리플렉션을 지원하지 않는다. 보안 문제에 있어서는 보안 모듈 자체가 CLDC 구현보다 크기 때문에 저수준의 가상머신 보안과 애플리케이션 레벨의 보안을 제공한다.

클래스 검증과정은 사전검증과 실행시검증이라는 새로운 방법을 도입해서 사용하는데, 이를 통해 오프 디바이스에서 공간 절약적인 처리와 검증시간을 줄일 수 있다. CLDC 에서는 JAR 파일을 사전로딩과 사전링킹을 통한 로마이징을 구현한다. CLDC 라이브러리는 lang, util, io 패키지만 사용한다.



[그림 4] 클래스 파일 검증 과정

3.2 MIDP(Mobile Information Device Profile)

MIDP 는 모바일 인포메이션 디바이스를 목표로 CLDC 컴피규레이션을 기반으로 설계된 자바 클래스 라이브러리이다. MIDP 에는 애플리케이션 모델, 유저 인터페이스와 이벤트 핸들링, 영속적 저장공간, 네트워킹, 타이머 지원이 정의되어 있다.

JAM(Java Application Manager)은 CLDC/MIDP 플랫폼의 새로운 애플리케이션 모델을 지원하기 위한 애플리케이션 관리 소프트웨어이고, 그 역할은 MIDP 애플리케이션인 MIDlet 를 다운로드해 설치, 업그레이드, 실행, 삭제하는 것이다. [표 2]는 JAM 의 기본적인 기능이다.

[표 2] JAM 의 기본 기능

기능	설명
추출(retrieval)	웹 사이트의 소스로부터 MIDlet 추출
설치(installation)	추출한 MIDlet 을 디바이스에 설치
실행(launching)	MIDlet 을 호출
버전 관리	설치된 MIDlet 을 새로운 버전으로 업그레이드

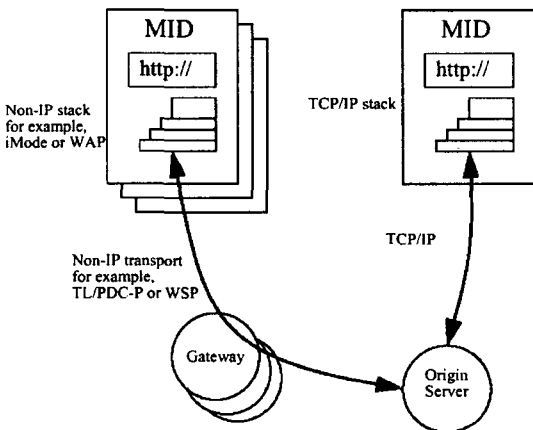
삭제(removal)	설치된 프로그램을 삭제
-------------	--------------

MIDlet 은 JAR 파일 형태로 배포되어야 하고, 이 JAR 파일에는 MIDlet 스위트의 패키지가 들어갈 수 있는데 MIDlet 스위트의 패키지는 JAR 콘텐츠에 대한 설명을 담고 있는 매니페스트, MIDlet 클래스를 상속한 클래스와 다른 공유 클래스, MIDlet 이 사용하는 리소스 파일로 구성되어 있다. 마이크로브라우저로부터 웹사이트에 접속해서 JAM 이 애플리케이션을 다운로드 해 설치, 업그레이드, 실행, 삭제하기 위해서는 애플리케이션의 이름, 버전, 제작자 등의 정보가 반드시 필요하다. 이러한 MIDlet 애트리뷰트는 애플리케이션 디스크립터에서도 정의될 수 있으며, 만약 애플리케이션 디스크립터와 매니페스트에서 동시 정의되었을 경우에는 애플리케이션 디스크립터가 우선한다.

JAR 파일을 추출하는 방법은 여러가지가 있지만, 가장 쉬운 방법이 브라우저를 이용하는 것이다. ME 든지 WAP 브라우저든 상관없이 브라우저는 콘텐츠 혹은 파일을 네트워크에서 찾아다니는 가장 보편적이고 일반적인 방법이다. 그러므로 WAP 브라우저를 통해 다운로드할 JAR 파일을 찾고, JAM 이 그 파일을 다운로드해 설치하는 것은 가장 생각하기 쉽고 구현하기가 용이하다.

네트워크를 통해 JAM 에게 MIDP 애플리케이션과 JAR 파일의 정보, 즉 MIDlet 애트리뷰트를 전송할 수 있는 효율적인 방법이 필요한데, 이에 대한 가장 좋은 솔루션이 바로 JAD(Java Application Descriptor)라는 애플리케이션 디스크립터이다. JAD 파일은 MIME 타입을 갖고 있는 파일이다.

그리고, MIDP 에서는 CLDC 의 커넥션 프레임워크에 대해 확장된 네트워킹에 대한 정의와 구현을 제공하고, TCP/IP 네트워크의 IP 프로토콜 구현과 WAP 게이트웨이를 통한 non-IP 프로토콜 구현을 포함하고 있다.



[그림 5] HTTP 네트워크 연결

[그림 5]에서 보는 것과 같이 한쪽은 WAP 게이트웨이를 통해서, 다른 한쪽은 TCP/IP 로 직접 웹 서버에 접속하고 있다.

3.3 자바 플랫폼 기반의 마이크로브라우저 설계

자바 플랫폼을 기반으로 하는 마이크로브라우저는 모바일 인포메이션 디바이스(MID) 위에 실시간 운영체제(RTOS)가 올라가고, 그 위에 KVM 을 포함하는 CLDC 가, CLDC 위에 MIDP 가 배치된다.

자바 플랫폼위의 마이크로브라우저는 WAP 를 사용하고, 사용자로부터 요청 웹 사이트에 접속하여 JAR 파일을 다운로드하여, JAM 을 이용해서 모바일 단말기에 애플리케이션을 설치하고 수행한다.

4. 결론 및 향후 연구 과제

자바 플랫폼 기반의 마이크로브라우저의 설계는 썬 마이크로시스템에서 제공하는 CLDC/MIDP 를 이용해서 쉽게 설계할 수 있었다. 그리고 이 마이크로브라우저가 가지는 장점은 현재 온라인에서 사용자가 요청한 애플리케이션을 다운받아서 사용할 수 있고, 또 오프라인에서도 애플리케이션 이용이 가능하다. 향후에 연구해야 할 과제는 이 마이크로브라우저를 구현하는 것이고, KVM 이 보다 빨리 클래스 파일을 수행되도록 가상 머신에 대한 연구가 필요하다.

5. 참고 문헌

- [1] CLDC 1.0 Specification, <http://java.sun.com/products/cldc>
- [2] MIDP 0.9 Specification, <http://java.sun.com/products/midp>
- [3] KVM WhitePaper, <http://java.sun.com/products/kvm>
- [4] J2ME, <http://java.sun.com/j2me>
- [5] WAP 포럼, <http://www.wapforum.com>
- [6] 무선 인터넷 & WAP 개발포탈, <http://mobile.daesang.co.kr>
- [7] WIPS, <http://wap.sejong.edu>
- [8] B.Venners, Inside Java Virtual Machine, McGraw-Hill, 1997
- [9] 박용재, 자바가상머신 프로그래밍, 인포북, 1999
- [10] Charles Arehart, Professional WAP, Wrox Press, 2000
- [11] William R. Stanek, Netscape Mozilla Source code guide, IDG Books, 1999