

# 모바일 표준 플랫폼 WIPI를 위한 WAP2.0 마이크로 브라우저의 설계 및 구현

최우영<sup>0</sup> 허 신  
한양대학교 컴퓨터공학과  
{wychoi<sup>0</sup>, shinheu}@cse.hanyang.ac.kr

## Design and Implementation of WAP2.0 Micro Browser for Mobile Standard Platform, WIPI

Woo-Young Choi<sup>0</sup> Shin Heu  
Dept. of Computer Science and Engineering, Hanyang University

### 요 약

모바일 표준 플랫폼 WIPI(Wireless Internet Platform for Interoperability)는 하드웨어 독립적으로 설계되어 어떠한 무선단말기에도 사용가능하며 개발언어인 C와 자바를 선택적으로 사용할 수 있는 국내 무선인터넷 표준 플랫폼이고 WAP2.0(Wireless Application Protocol) 규격은 XHTML의 사용으로 유.무선 인터넷의 통합 콘텐츠 제공이 가능한 무선 응용프로토콜이다. 본 논문에서는 WIPI 표준 플랫폼에서 단말기의 무선인터넷 활용을 위해 XHTML을 마크업 언어로 채택한 WAP2.0 규격의 마이크로 브라우저 원형(Prototype)을 설계하여 구현하였고 WIPI 시뮬레이터를 사용하여 응용프로그램으로 구현된 마이크로 브라우저의 동작실험을 함으로써 WIPI 표준 플랫폼을 위한 WAP2.0 마이크로 브라우저의 활용 가능성을 제시한다.

### 1. 서 론

세계 최초 CDMA 상용화 국가인 한국의 이동통신과 무선인터넷 기술은 세계최고라고 인정 받고 있으며, 무선인터넷 플랫폼의 표준규격인 WIPI(Wireless Internet Platform for Interoperability)는 또 하나의 세계적인 성공사례가 될 것이다. WIPI 규격은 단말기 미들웨어의 표준규격을 제정한 것으로 하드웨어 독립적이며 무선응용프로그램의 상호 운용성이 보장되고 개발언어는 C와 자바를 선택적으로 사용할 수 있다.

1세대 무선인터넷은 무선응용프로토콜 방식(WAP, ME)을 채택하고 있으며 사용자들은 무선 단말기의 내장된 마이크로 브라우저를 사용해서 무선인터넷 서비스를 제공 받는다. 2세대 무선인터넷은 무선 단말기에 미들웨어 플랫폼인 VM 방식을 사용하므로 더욱 다양한 응용프로그램들을 개발하여 배포할 수 있다. 2세대 무선인터넷은 플랫폼 방식이 적용되면서 각종 응용프로그램이 무선 단말기에 내장될 필요가 없으며 사용자는 무선인터넷을 통해 사용하고자 하는 응용프로그램을 무선 단말기에 다운로드하여 사용하면 된다.

1세대와 2세대 무선인터넷 방식은 서로 다르지만 동일하게 필요한 응용프로그램이 있는데 이것이 바로 마이크로 브라우저다. 마이크로 브라우저는 무선인터넷 서비스를 위한 기본적인 중요 응용프로그램이다. 무선 인터넷에 접속하여 사용자에게 다양한 콘텐츠를 단말기 화면에 최적화하여 보여주며 단말기의 하드웨어적으로 제한된 사용자 인터페이스(User Interface)를 극복하여 사용하기 편리한 UI를 제공할 수 있다.

WAP2.0의 경우 XHTML을 마크업 언어로 채택하여 유.무선 인터넷에 동일한 콘텐츠의 제공이 가능하며 WAP2.0 규격 문서에서는 기존에 사용되고 있는 무선 마크업 언어인 WML과 XHTML이 동시에 지원되는 이중형태(Dual-Mode) 마이크로 브라우저의 개발을 권고한다.

본 논문의 2장에서는 국내 무선인터넷 플랫폼 동향과 WIPI 표준규격 그리고 WAP2.0과 XHTML의 기술적 개요를 관련연구로 간략히 기술하였고 3장에서는 WIPI기반의 마이크로 브라우저의 프로토타입을 설계하고 구현하였으며 구현된 마이크로 브라우저를 WIPI 플랫폼에 이식하여 시뮬레이션함으로써 동작 검증실험을 하였다.

### 2. 관련 연구

#### 2.1 국내 무선인터넷 플랫폼

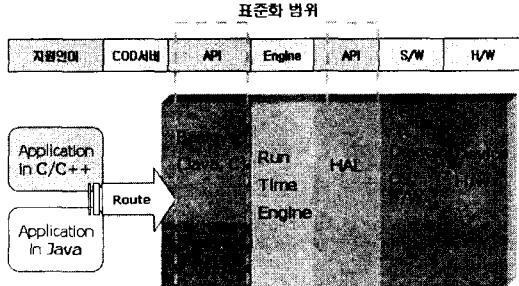
현재 국내 이동통신사의 무선인터넷 플랫폼은 [표 1]과 같이 총 7가지가 제공되고 있다. 가장 최근에 서비스되는 SKT의 WITOP 플랫폼을 제외한 모든 플랫폼의 개발언어는 C 또는 자바 중 한가지 언어만을 지원하며 개발언어가 같더라도 무선인터넷 플랫폼이 다르면 콘텐츠 간의 상호 운용성은 없다. 이와 같은 문제의 해결을 위해 모바일 플랫폼 표준화 연구가 진행되고 있으며 WIPI라는 통합 플랫폼이 개발되었다.

[표 1] 국내 이동통신사의 무선인터넷 플랫폼 현황

플랫폼	개발언어	수행방식	이동통신사
KVM	자바	인터프리터	LGT
SK-VM	자바	인터프리터	SKT
GVM	C/C++	인터프리터	SKT
MAP	C/C++	바이너리	KTF
BREW	C/C++	바이너리	KTF
WITOP	자바, C/C++	인터프리터	SKT

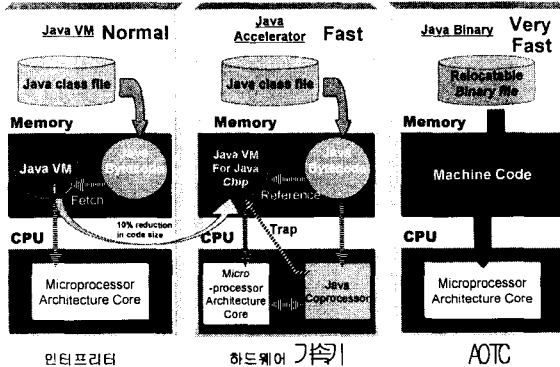
2.2 WIPI 기술 개요

WIPI 표준규격은 무선 단말기에 탑재되어 응용프로그램을 수행할 수 있는 실행환경을 제공하는 모바일 표준 플랫폼(Mobile Standard Platform)을 정의하고 있다.



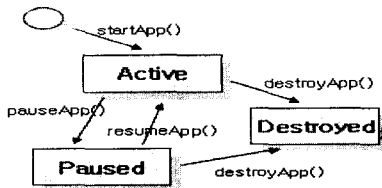
[그림 1] WIPI 플랫폼 개념적 구조와 표준화 범위

[그림 1] 과 같이 WIPI 규격에서 정의하고자 하는 모바일 표준 플랫폼은 플랫폼 이식성을 높이기 위한 표준화된 하드웨어 추상화 계층인 HAL(Handset Adaptation Layer) 을 적용하여 플랫폼 호환성을 제공하며 다양한 응용 프로그램 개발을 촉진하기 위해 기본 응용 프로그래밍 인터페이스(Basic Application Programming Interface 또는 Basic API)로 구성된다. 플랫폼 개발언어로는 C 와 자바를 모두 지원하며, 자바의 경우 응용 프로그램의 실행속도 향상을 위해 [그림 2]의 AOTC (Ahead Of Time Compile) 방식을 사용한다.



[그림 2] 자바 응용프로그램 Jlet의 성능 개선방법

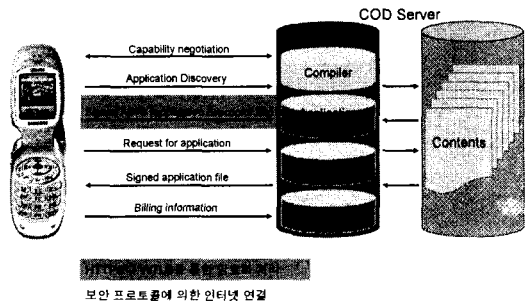
AOTC방식은 바이트 코드 형태로 되어 있는 자바 어플리케이션을 수행하기 전에 컴파일해서 단말기의 CPU에 최적화된 바이너리 코드를 생성하는 방식이다. AOTC 방식을 통해서 고속화 수행이 가능한 WIPI의 자바 응용프로그램인 Jlet은 [그림 3]와 같은 상태변화를 가지게 되어 [그림4]와 같은 절차로 실행된다.



[그림 3] Jlet 응용프로그램의 3가지 상태

```
protected void startApp(String args[]){ // App. 시작
display= Display.getDefaultDisplay(); //display 객체
myCard = new MyCard(); //card카드 객체 생성
display.pushCard(myCard); //card 객체 보여줌
}
protected void pauseApp(){...} // App. 정지
protected void resumeApp(){...} // App. 재시작
protected void destroyApp(boolean ){...} // App. 중단
```

[그림 4] Jlet 응용프로그램 실행절차

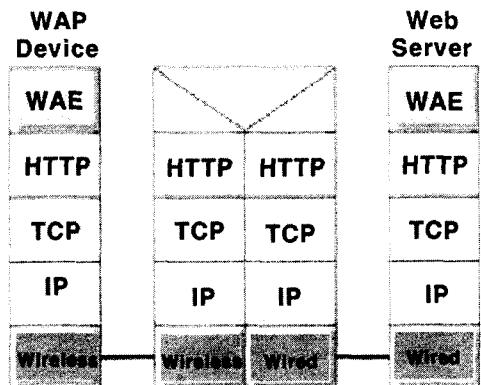


[그림 5] WIPI 서비스 과정

WIPI 응용프로그램은 [그림 5]와 같은 과정을 통해서 사용자에게 제공된다. 신뢰할 수 있는 COD 서버는 기존 자바 실행환경에서 요구되는 바이트코드 검증과정을 통해서 WIPI Jlet 응용 프로그램에 대한 안전한 서비스를 제공한다.

2.3 무선인터넷 응용프로토콜: WAP2.0

WAP2.0은 다양한 무선 단말기에 적용 가능하며 [그림 6]과 같이 유·무선간의 상호 호환성이 지원되는 규격을 가지고 있다. 특히, 사용되는 프로토콜과 마크업 언어의 경우 HTTP1.1와 XHTML 기술을 채용함으로써 유·무선 인터넷의 통합 서비스가 가능하다.



[그림 6] WAP2.0 프로토콜

WAP2.0에서 사용되는 마크업 언어는 무선 단말기 프로파일(Profile)이 정의된 XHTML Basic을 사용하며, 이 언어는 문서의 구조가 매우 잘 정의되어 있다.

[표 2] XHTML과 HTML 특성 비교

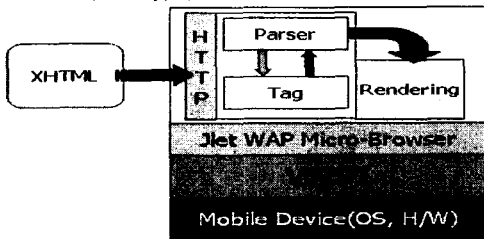
영역	XHTML (WML 2.0)	HTML
표준 규격	XHTML Basic base	HTML 4.01
문서구조	Head&Body,Deck&Card	Head&Body
문서표준	표준화된 문서	비표준 허용
대소문자	구분	구분하지 않음
Text & Image	b, big, br, p, code, dfn, div, em, h1, h2, h3, h4, h5, h6, 외 16가지	+12
Table	caption, table, tr, td, th	+5
Form	fieldset, form, input, select, option, 외 4가지	+3
List	dd, dl, dt, li, ol, ul	+2

XHTML Basic은 어떠한 플랫폼에서도 적용 가능하도록 표준화된 문법을 허용하며, [표 2]와 같이 무선 단말기에 적용 가능한 태그(tag)만을 정의해서 지원하고 있다.

### 3. 마이크로 브라우저의 설계 및 구현

#### 3.1 설계

WAP2.0 규격문서에서 정의한 마이크로 브라우저는 WAP1.x의 마크업 언어인 WML과 WAP2.0 마크업 언어인 XHTML Basic을 모두 지원하는 이중모드(Dual-Mode) 브라우저를 정의하고 있으나, 본 논문에서는 WAP2.0 마크업 언어인 XHTML Basic의 Text 관련 일부 태그(Tag)만을 지원하는 마이크로 브라우저의 원형(Prototype)을 개발하였다.



[그림 7] 마이크로 브라우저 원형(Prototype) 구조

서버로부터 XHTML 파일을 HTTP Get방식으로 가져오게 되며 마이크로 브라우저의 Parser, Tag를 통해서 미리 정의된 XHTML 태그요소(Tag Element)를 해석하여 단말기의 LCD에 표현하게 된다. [그림 7] 참조.

#### 3.2 구현

WIPI 응용 프로그램은 MIDP의 midlet과 유사한 Jlet으로 구현하였고 WIPI규격에서 지원하는 기본 JAVA API와 함께 MIDP의 java.net.\*의 API를 포함하여 개발하였다.

Parser 클래스는 지정된 URL의 XHTML파일을 전송 받아 고정 문자집합(Static set of Escape)을 분류한 후 Tag 클래스의 Vector 스트림으로 전송하여 태그값 비교를 통해 사용 가능한 태그일 경우 WapMB 클래스는 전송받은 데이터를 시뮬레이터의 LCD에 표시하게 된다.[표 3] 참조.

[표 3] 소스구성 및 API

Class and API	Description
Parser.class	특정 URL의 데이터를 가져와서 Tag 분류
Tag.class	전송 받은 Tag 데이터의 사용유무판단.
WapMB.class	DTD를 통한 Tag 해석 및 LCD표시
org.kwis.msf.io	WIPI 표준 입출력 클래스
org.kwis.msp.lcdui	WIPI 표준 LCD화면의 표현 클래스

### 3.3 동작 검증

동작검증을 위해 Aroma-WIPI 시뮬레이터를 사용하였으며 [그림 8]은 브라우저 실행을 통해 XHTML을 인식한 화면이다.



[그림 8] WIPI 시뮬레이터를 사용한 마이크로 브라우저 실행화면

### 4. 결과 및 향후 연구과제

본 논문에서는 국내 모바일 표준 플랫폼 WIPI에 적용 가능한 WAP2.0 마이크로 브라우저를 Jlet 응용프로그램으로 구현하였다. 본 연구의 마이크로 브라우저는 모바일 표준 플랫폼 WIPI 규격에 맞추어 구현되었으며, 이를 통해서 WIPI기반의 마이크로 브라우저 개발과 활용 가능성을 제시한다.

향후 연구 과제로 WAP2.0 마크업 언어인 XHTML Basic의 모든 태그 요소(Tag Element)의 지원과 동적인 무선 페이지 구현을 위한 스크립트(Script) 엔진 장착 그리고 즉각적인 무선 페이지의 접속과 응답 속도의 향상을 위한 캐쉬관리(Cache-Management) 시스템 등을 마이크로 브라우저에서 제공할 수 있도록 연구를 진행 중이다.

### 5. 참고 문헌

- [1] ETRI, "WIPI 표준화 현황 및 규격", Korea Mobile Conference, June 26,2002
- [2] KWISF, "KWISFS-K.05.001 모바일 표준 플랫폼 규격", 한국무선인터넷 표준화 포럼, April 16,2002
- [3] TTA, "TTA.KO-06.0036 Mobile Standard Platform Specification", 한국정보통신기술협회, May 7,2002
- [4] <http://www.w3.org/TR/xhtml-basic/>
- [5] WAPFORUM, "WAP Architecture Version 12-July-2001", WAPFORUM, July 12,2001
- [6] WAPFORUM, "XHTML Mobile Profile Version 29-Oct-2001", WAPFORUM, October 29,2001
- [7] <http://www.openmobilealliance.org/>