

무선 인터넷의 WAP 기반 서비스와 자바 기반 서비스간의 성능 비교

오기욱*

A Performance Comparative between WAP Service and JAVA Wireless Internet Service

Gi - oug Oh *

요 약

무선인터넷을 개발하는 방법은 여러가지 개발방법이 있다. 그 중 대표적인 것으로 WAP(Wireless Application Protocol) 기반 서비스와 자바 기반 서비스가 있다. WAP 기반은 사용자에게 한정된 편의성을 제공하지만 자바 기반의 무선 인터넷은 사용자에게 많은 편의성과 GUI(Graphic User Interface)를 지원하여 쉽게 접근 할수 있게 한다. 본 논문에서는 WAP 기반 애플리케이션 개발방법과 자바 기반 애플리케이션 개발 방법을 제시하며, 실험을 통해 두 시스템간의 성능을 비교한다.

Abstract

There are many development methods to implement wireless internet service. These methods are WAP(Wireless Application Protocol) based services and Java based service. WAP based service has been proposed to support limited convenience for user. But Java based service can support many convenience and GUI(Graphic User Interface) and easy to use to access for user. This paper will be propose a WAP based application development method and Java based application development method. As an illustrative experiment, WAP based service and Java based service are used to show performance for each systems.

* 강원관광대학 컴퓨터정보계열 전임강사

I. 서론

무선 인터넷을 개발하는 방법은 여러 가지 방법이 있다. 그중 WAP을 기반으로 하는 개발방법과 자바를 기반으로 하는 개발 방법이 있다. WAP을 기반으로 하는 기술에서 사용하는 기술의 대표적인 것이 바로 마크업 언어이다. 마크업 언어로는 WML(Wireless Markup Language), MML(Mobile Markup Language), HDML(Handheld Device Markup Language) 등이 있다. 그러나 이러한 언어의 특징은 GUI(Graphic User Interface)의 형태를 지원하지 못한다는 것이 단점이다 [1].

WAP은 이동통신 단말기(Handheld Device)나 페이지(Pager) 및 PDA 등 소형 무선 단말기 상에서 인터넷 데이터를 이용하는 것에 대한 규약이며, WAP은 표준 네이밍 모델(Standard Naming Model)과 콘텐츠 타이핑(Content Typing), 표준 콘텐츠 형식(Standard Content Type) 및 표준 통신 프로토콜(Standard Communication Protocol)을 제공한다[2].

표준 네이밍 모델은 웹서버상에 있는 WAP 콘텐츠를 인식하는 것을 말하며, 콘텐츠 타이핑은 WAP 내용이 웹타입과 호환되는 특별한 타입을 갖는 것이다. 표준 콘텐츠 형식은 WAP 콘텐츠 형식이 웹 기술을 기반으로 하기에 마크업 언어, 이미지, 달력정보, 스크립 언어를 포함하는 것을 말한다. 마지막으로 표준 통신 프로토콜은 이동통신 단말기가 네트워크에 연결된 웹서버와 통신을 가능하게 만드는 것을 말한다[3].

그러나 이러한 점에도 불구하고 WAP을 기반으로 하는 무선 인터넷 서비스는 대부분이 메뉴방식에 기반을 두고 있으며, 텍스트위주의 서비스와 항상 네트워크가 연결되어 있어야만 서비스를 받을 수 있다는 단점을 가지고 있다. 이러한 것은 사용자가 무선 인터넷을 사용하길 꺼리는 가장 근본적인 원인이며, 사용에 불편한 점이다. 그리고 WAP을 기반으로 하는 서비스의 두 번째 단점으로는 플랫폼에 종속적이며, 사용자 인터페이스에 한계가 있다는 점이다.

자바를 기반으로 하는 개발 방법은 이동통신 단말기의 개발 환경을 발전시킨 최적의 기술이지만, 처음 자바의 개발 목적은 전자제품의 업그레이드 시 새로운 구성요소를 바꾸는 것에 관계없이 기존과 새로 구성될 아키텍처에 독립적으로 진행하기 위해서 개발하였다. 자바 기반 무선 인터넷 핵심기술은 자바 애플리케이션을 무선 인터넷 망을 이용하여 동적으로 다운로드하여 설치하도록 하였으며, 객체 지향적이면서 간결한 언어구조를 가지고 있으며, 플랫폼에 독립적이다 즉, 플랫폼에 독립적이며, 아키텍처 중립성이 뛰어나 어떠한 운영체제에서도 다운로드 혹은 이식하여 사용할 수 있는 WORA(Write Once, Run Anywhere)에 적합한 언어이다[4].

자바기반의 무선 인터넷 서비스를 개발 시에 유용한 특징으로는 첫째, 동적 애플리케이션 다운로드가 가능하며, 클로스 플랫폼을 제공한다. 세 번째로, 사용자의 만족성을 높일 수 있고, 역동성을 제공하며, 네 번째로는 네트워크에 대해 WAP과는 다른 비연결성을 갖으며 보안에 유연하다는 것이다.

본 논문에서는 WAP 기반의 인터넷 서비스의 문제점을 해결하기 위한 방법으로 자바 기반의 무선인터넷 서비스를 개발하며, 두 시스템간의 성능을 비교 분석한다.

II. 관련연구

1. WAP 플랫폼

WAP은 응용 통신 프로토콜이며, 디지털 무선 전화와 같은 무선 단말기를 이용하여 정보와 서비스들을 액세스 하는데 사용한다. WAP은 에릭슨, 모토로라, 노키아 등의 업체에 의해서 설립된 WAP 포럼(Forum)에 의해서 설립되었다. 무선장치(Wireless Devices)들은 제한된 CPU와 메모리, 배터리를 가지고 있으며, 간단한 사용자 인터페이스를 갖으며 무선네트워크는 적은 대역폭(Bandwidth)과 높은 잠재성(Latency), 그리고 저급한 활용성과 안전성을 갖는다. 그러나 WAP 에 대한 사항은 개방되어져 있으며 대중화되어져 있다. 따라서 이러한 것을 이용하여 사용자 들은 실제로 어느 곳이나 아무 때나 간단한 정보를 얻을 수 있으며, 효율적으로 다양한 네트워크와 장비를 활용한 사용자 인터페이스를 이용할 수 있다[5].

WAP 포럼은 몇 가지 특징을 갖는데 무선장치들에 서비스를 하며 인터넷을 활용한다. 두 번째는 모든 무선 네트워크를 망라한 범용적인 무선 프로토콜을 구성한다. 세 번째로, 광범위한 무선 네트워크와 장치 타입들을 활용하는 애플리케이션과 콘텐츠를 만든다.

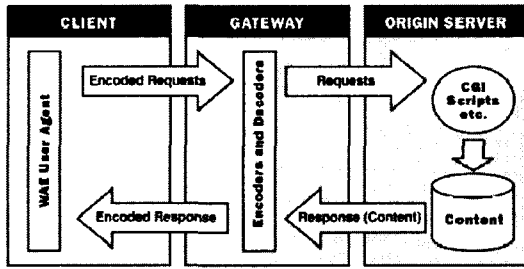


그림 1. WAP 구조
Fig 1. WAP Architecture

<그림 1>은 WAP 구조를 나타낸 그림이다. 게이트웨이 가 웹 서비스의 HTTP의 역할을 해주는 구조를 가지고 있으며, 원래의 서버에서 자료 및 콘텐츠를 가공 처리하는 역할을 수행한다. 이렇게 수행된 자료 및 콘텐츠를 WAP 게이트웨이를 통해서 원하는 사용자의 이동통신단말기에 보내준다(6). 이렇게 수행 시에 사용하는 WAP 프로토콜은 <그림 2>에 나타내었다.

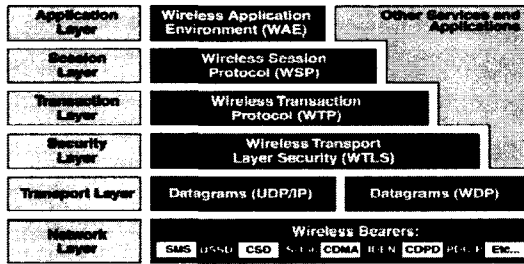


그림 2. WAP Protocol 구조
Fig 2. WAP Protocol Structure

2. 자바 무선 인터넷

자바는 내장형(Embedded) 장치에 효율적으로 사용할 수 있는 간단한 프로그래밍 언어이다. 자바는 프로그래밍 측면과 컴퓨팅 환경 측면을 동시에 가진 언어로서 자바 플랫폼은 컴퓨팅 환경의 자바를 의미하며, 플랫폼 구성은 자바가상머신(Java Virtual Machine)과 클래스 라이브러리(Class Library)라는 표준 API(Application Programming Interface)들의 집합으로 구성되어 있다(7).

자바는 1.1 버전까지는 애플리케이션이나 애플릿(Applet)의 개발에 목적을 두었으나, 향후 버전부터는 다양한 플랫폼에 맞는 자바가 등장하게 되었다. 그림 3은 자바 플랫폼의 구성도이다.

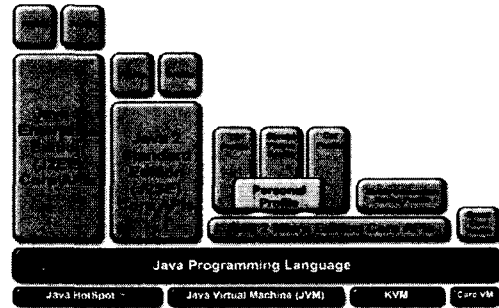


그림 3. 자바 플랫폼 구조
Fig 3. JAVA Platform Structure

<그림 3>에서 보면, 3개 영역으로 구분 할 수 있는데, 개발 목적에 따라 구분이 가능하다. 첫 번째가 엔터프라이즈 서버이고, 두 번째가 데스크톱을 나타내며, 세 번째가 내장형 장치의 영역으로 나눌 수 있다. 자바 기술에서 보면, 그림 3과 같이 구분을 할 수 있는데 J2EE(Java 2 Enterprise Edition)는 엔터프라이즈 서버부분을 나타내며, J2SE(Java 2 Standard Edition)은 데스크 탑용, J2ME(Java 2 Micro Edition)은 내장형(Embedded) 장치를 나타낸다.

자바 기술은 WAP을 사용하는 WML로 만들어진 무선 웹브라우저의 성능을 크게 향상 시켜준다(8). 자바는 계속하여 네트워크 상으로 연결을 하지 않아도 다운로드를 받아 데임이나 애니메이션 등의 콘텐츠를 이용할 수 있다. 자바 인터넷의 특징을 보면, 다양한 방식으로 서비스가 가능하며, 가장 중요한 서비스를 계속하여 제공받기 위해 네트워크 상에 연결이 되어 있을 필요가 없다는 점이다. 또 실시간으로 동적으로 다운로드가 가능하며, 사용자가 다운로드받아 사용할 애플리케이션을 선택하여 받을 수 있고, 플랫폼에 독립적이다. 따라서 각각의 플랫폼에 맞추어 애플리케이션을 제작할 필요가 없다는 장점을 가지고 있다. 보안문제도 자바가 이미 보안모델을 구축하고 있기에 이것을 이용하여 네트워크의 보안문제도 해결할 수 있다는 점이 장점이다.

III. 자바 애플리케이션 개발

1. 구현 환경

이동통신 전화에 자바 애플리케이션을 구현하기 위한 이동통신 환경은 하드웨어 구성부분과 소프트웨어 구성 부분으로 나눌 수 있다. 하드웨어 구성에 대한 부분은 아래 < 표 1>에 나타내었다.

표 1. 이동통신 전화 하드웨어
Table 1. Handheld Hardware

하드웨어 구성	
단말기	i-book Model Seriez
통신	Qualcomm Mobile Station Modem 3100
메모리	4MB Flash Memory 512 KB StaticRAM
화면	128 * 128 Pixel gray scale LCD
네트워크속도	64 Kbps

표 1에서 하드웨어 구성을 나타내었으며, 소프트웨어 구성에 대한 부분은 <표 2>에 나타내었다.

표 2. 이동통신 전화 소프트웨어
Table 2. Handheld Software

소프트웨어 구성	
KVM	230 Kbyte code size 108 Kbytes Heap Space
자바	64 Kbyte * 10 Application Storage J2ME, CLDC
인터페이스	Kitty Hawk API
폰트	한글 10*12, 영문 5*12
버전	최신버전 (계속적 업그레이드)

2. 자바 서비스 플랫폼과 Kitty Hawk API

자바 서비스 플랫폼은 크게 클라이언트와 콘텐츠 제공자(Content Server)로 구분할 수 있으며, 그림 4에 자바 서비스 플랫폼을 나타내었다.

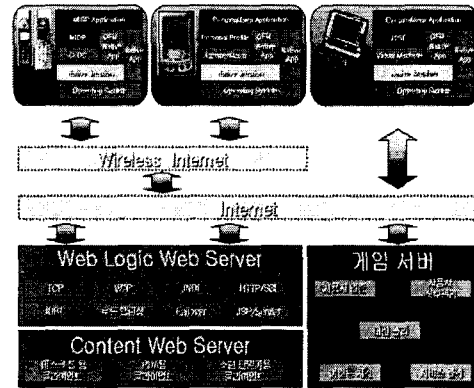


그림 4. 자바 서비스 플랫폼
Fig 4. Java Service Platform

Kitty Hawk에서 제공하는 API는 저급 API와 고급 API로 구분할 수 있으며 저급 API는 Canvas 클래스를 사용하여 구현할 수 있으며 스크린을 조정할 수 있다. 반면에 고급 API는 네비게이션, 문자입력, 리스트, 스크롤링을 구현할 수 있으며 문자위주의 애플리케이션을 구현할 수 있다.

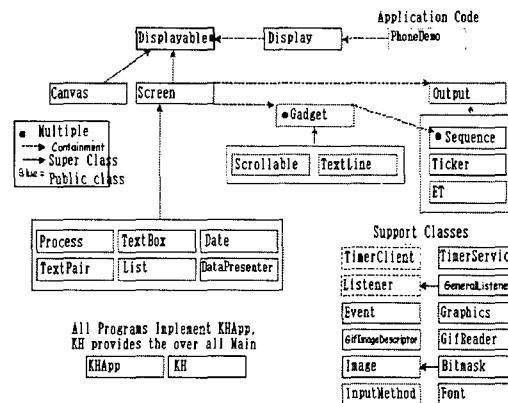


그림 5. Kitty Hawk 클래스 다이어그램
Fig 5. Kitty Hawk Class Diagram

<그림 5>는 Kitty Hawk 클래스 다이어그램을 나타내었다. Kitty Hawk은 저급 API와 고급 API를 한 화면에 같이 구현할 수 없다는 단점을 갖고 있다.

자바 애플리케이션이 구현되어 이동통신 단말기에서 실행되는 구조는 <그림 6>에서 나타내었다.

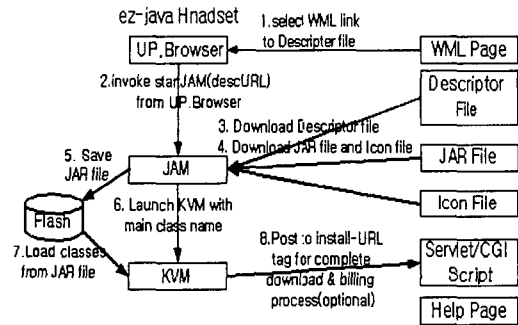


그림 6. 자바 애플리케이션 흐름도
Fig 6. Java Application Flow

〈그림 6〉에서, 사용자가 무선 브라우저를 이용하여 무선인터넷 서비스를 접속한 후에 애플리케이션을 선택하기 위해 WAP 페이지를 선택한 후 JAM 파일을 사용자가 URL을 입력하여 다운로드받는다. JAM 파일을 플래쉬 메모리에 올리고 마지막으로 KVM이 JAR 파일의 클래스 파일을 읽어 이동통신 단말기에서 자바 애플리케이션을 실행한다. JAM 파일의 구조는 그림 7에서 보듯이 프로그램 이름, 프로그램 버전, 개발자, 프로그램 메인 클래스 이름, JAR 파일의 위치를 나타내는 URL, JAM 파일의 위치를 기억하고 있는 URL, JAR 파일의 크기, 저장을 위한 메모리 크기 등으로 구성되어 있다.

```

Application-Name = JAM 메뉴에 뜨는 이름
Application-Version = 어플리케이션 버전
Application-Creator = 개발자 혹은 회사
Main-Class = 메인 클래스의 이름
JAR-File-URL = 어플리케이션의 URL
JAM-File-URL = JAM 파일의 URL
JAR-File-Size = 어플리케이션 파일 크기
ScratchPad-Size = 저장을 위한 메모리 크기
    
```

그림 7. JAM 파일 구조
Fig 7. JAM File Structure

IV. 구현 및 평가

1. 요구사항 및 모듈설계

시스템의 요구사항을 분석하여 유즈케이스(Use Case)

다이아그램으로 표현하여 보면 〈그림 8〉과 같다.

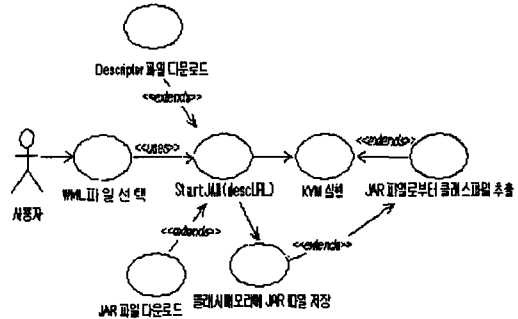


그림 8. 시스템 구현 유즈케이스 다이어그램
Fig 8. System Implement UseCase Diagram

〈그림 8〉을 살펴보면, 사용자는 WML 파일을 선택하고 선택된 애플리케이션을 실행한다. Descriptor 파일과 JAR 파일을 다운로드하고, JAM을 시작한다. 수행된 JAR 파일을 플래쉬 메모리에 저장하고, KVM을 실행하는 이러한 일련의 과정을 거쳐 JAR 파일로부터 클래스 파일을 추출하여 자바 프로그램을 실행한다 [9,10]. 여기서 JAR 파일로부터 클래스 파일을 추출하여 자바 프로그램을 실행하는 시스템의 핵심 클래스 다이어그램도 추출하여 나타낼 수 있다.

2. 시스템 구현

구현환경을 윈도우 2000 서버를 사용하였으며, 자바 JDK 1.3 버전을 사용하였다. 클래스 API로는 Kitty Hawk를 사용하였으며, 퀄컴 MSM(Qualcomm Mobile Station Modem) 3100을 사용하여 개발하였다.

접속 방법은 셀룰러폰을 사용하여 서비스 접속을 할 수 있고 시뮬레이터를 이용하여 접속하는 방법도 있다. 셀룰러폰에 자바 클래스를 자바 클래스 파일 정보저장소에 다운로드 받으면, 저장된 클래스 파일은 셀룰러폰에서 삭제하기 전까지는 영구적으로 남아 있다. 아래 표 3은 실제 WML을 사용하여 구현한 asp이다[1,11].

표 3과 같이 실제 WML을 사용하여 구현을 한 다음 셀룰러폰에 Preverify를 수행한 자바 클래스 파일을 압축하고 확장자를 .jam이라 저장하는 JAM 파일을 작성한다. 그리고 〈표 3〉과 같이 프로그램을 작성하여 셀룰러폰에 .asp 확장자를 갖도록 만든 프로그램을 업로드하면 된다.

```
<? xml version="1.0" encoding="KS_C_5601-1987" ?>
<!DOCTYPE wml PUBLIC "-//PHONE.COM/DTD WML 1.1//EN"
"http://www.phone.com/dtd/wml.dtd">
<wml>
  <card id="test" title="hakwa inform">
    <p>
      <select>
        <option onpick="kongjinfo.asp">공지사항</option>
        <option onpick="hakwaact.asp">학과활동</option>
        <option onpick="hakwainfo.asp">학과정보</option>
      </select>
      <do type="accept" label="선택"><go href=""/></do>
      <do type="prev"><prev/></do>
    </p>
  </card>
</wml>
```

표 3. WML로 실제 구현한 사례
Table 3. Implement of WML Example

3. 시스템 비교

무선 인터넷의 WAP 기반 서비스와 자바 기반 서비스간의 성능을 비교 평가 하기위해 평가항목을 선정하였다. 이 항목을 선정하여 일정한 시간대에 네트워크가 안정된 시간을 이용하여 평가하였으며, 동일한 환경을 갖도록 환경을 조성하여 평가하였다.

평가항목을 보면, 프로그램 다운로드 시간, 메모리 로드 시간, 데이터 전송시간, 메뉴 이동시간 등으로 나눌 수 있으며, 기타 항목으로는 플랫폼, 사용자 편의성, 보안문제등을 선정하였다. 아래 표 4는 WAP 기반 서비스와 자바기반 서비스간의 성능을 비교 분석한 표이다.

비교항목	WAP	JAVA
프로그램 다운로드	자원인원	실시간,동적 애플리케이션 다운로드
프로그램 다운로드 시간	필요 없음	30초
메모리 Load 시간	필요 없음	17초
데이터전송시간	2초 ~ 4초(평균)	3초(평균)
메뉴이동시간	2초 ~ 6초(평균)	4초 ~ 6초(평균)
네트워크	연결성	비연결성
애플리케이션 선택 기능	없음	있음
서비스	WAP 브라우저	KVM
플랫폼	종속적	독립적
사용자 편의성	없음(텍스트 위주의 메뉴제공)	있음(양상된 그래픽 제공)
보안문제	편견히 해결되지 않음	저버본안, 무선네트워크 보안

표4. 자바기반 서비스와 WAP 기반 서비스간의 비교
Table 4. Comparative of WAP based service and Java based service

V. 결론

본 논문은 무선인터넷에서 WAP 기반 서비스를 사용한 시스템과 자바기반 서비스를 사용한 시스템간의 성능을 비교 평가하였다. 평가 결과에서 보면, 흔히 자바기반 서비스 시스템이 WAP 기반의 시스템보다 월등하게 향상된 성능을 가졌다고 생각을 하겠지만, 실험의 결과는 성능의 면에서는 차이점이 크게 발생하지 않았다. 오히려 계속적으로 사용하지 않는 서비스를 이용한다면, WAP 기반 서비스가 우수하지만 사용자 편의성과 보안문제, 플랫폼등 그리고 네트워크에 비연결성을 가진다는 점에서 자바기반 서비스 시스템이 우수하였다. 추후 연구할 과제는 원활한 한글처리 및 코드의 크기가 큰 자바 애플리케이션을 최적화하여 실행하는 부분이며, 효율적인 보안문제에 대한 부분이 있다.

참고문헌

- [1] (주)애니빌무선인터넷연구소, 무선인터넷홈페이지만들기 영진닷컴, 2002
- [2] 남기범, 이진명 "무선웹기술과 전망", 2000. 6 정보과학회 제 18권 제 6호 p 32 - 38
- [3] http://www.wapforum.org/what/WAPWite_paper1.pdf[4] <http://www.sun.com/software/java/overview.html>
- [5] http://www.wapforum.org/what/EAPWhite_Paper1.pdf
- [6] LG - 텔레콤, Ez - I WAP Contents 개발규격서 (v1.6.1)
- [7] <http://sunsite.iisc.ernet.in/java/whatis/whats-jav/index.html>

- (8) 무선인터넷 편찬위원회, 무선인터넷백서 2001, 2000
- (9) Booch, The Unified Modeling Language User Guide, Addison Wesley, 1999
- (10) Martin Fowler, UML Distilled, Addison Wesley, 1997
- (11) <http://www1.wapforum.org/tech/document/WAP-218-GCAG-20010208-d.pdf>

저자 소개



오기욱

경원대학교 전산학과 공학사
승실대학교 컴퓨터학과 공학석사
승실대학교 컴퓨터학과 박사과정
현재 강원관광대학 컴퓨터정보과
교수