

## 모바일 컴퓨팅을 위한 WAP 프로토콜 스택

이 양 선\*

### 1. 서 론

인터넷 비즈니스 시장은 물리적인 시간과 공간의 제약을 받는 유선 인터넷 환경의 시장 개념에서 시·공간의 제약을 해소하고 언제(any time), 어디서나(any where) 자유롭게 사용할 수 있는 무선 인터넷 환경의 시장 개념으로 이동 중에 있다. 그러나 아직까지도 현재의 인터넷 환경은 단위 네트워크들을 케이블로 연결한 유선망에 기반을 두고 있기 때문에 시·공간적 제한 사항들을 완전하게 해소해 주지 못하고 있는 실정이다. 이런 이유로 이동 단말기나 PDA와 같은 Post PC를 이용한 무선 인터넷에 관한 연구가 활발하게 진행되고 있는 상황에서 Phone.com, Ericsson, Nokia, Motorola 등의 기업이 WAP 포럼을 결성하여 무선 인터넷 규격인 WAP (Wireless Application Protocol)을 발표하였다. WAP의 구조는 크게 Application Layer(WAE), Session Layer(WSP), Transaction Layer (WTP), Security Layer(WTLS), Transport Layer(WDP)로 구분할 수 있는데 이는 유선 인터넷상의 각각의 프로토콜 계층을 무선 환경으로 변환시킨 것이다.

기존 HTML 기반의 유선 인터넷 환경의 콘텐츠를 이동 단말기에서 서비스 받기 위해서는 유선

환경에 비해 상대적으로 열악한 전송 속도와 디스플레이 환경을 극복해야만 하는데 WAP 모델에서는 XML 기반의 WML(Wireless Markup Language)과 Gateway를 통해 인코딩된 바이너리 형태의 전송을 통하여 해결하였다. 이 중 WAP Gateway는 WDP(Wireless Transport Layer), WSP(Wireless Session Layer)와 같은 WAP 프로토콜 스택의 구현을 통한 HTTP 프로토콜과 해당 WAP 프로토콜 사이의 변환 및 WML 문서의 인코딩 기능 등을 제공한다[1,10,11,13].

WAP 프로토콜 스택은 무선 인터넷 서비스 제공을 위한 WAP Gateway로 사용자 단말기와 Web/WAP 서버 사이에 위치하여 프로토콜 변환 기능과 제공되는 문서의 인코딩 기능 등을 제공하는 미들웨어 시스템이다. 구현된 WAP Gateway는 플랫폼에 독립적으로 작동될 수 있도록 Java 언어를 사용하여 구현하였으며 WDP, WSP layer만을 이용하여 통신하는 WSP 비연결형 통신 기능과 WDP, WTP, WSP layer를 이용하는 WSP 연결형 통신 기능이 모두 수행 가능하다. WAP 프로토콜 스택 구현에 사용한 WAP 규격은 1.2 버전이다.

### 2. WAP의 개요

WAP은 1997년에 에릭슨(Ericsson), 모토롤라

\*서경대학교 컴퓨터공학과 교수

(Motolora), 노키아(Nokia), 그리고 언와이어드 플래닛(Unwired Planet - 현재의 Phone.com)이 주축으로 WAP 포럼(http://www.wapforum.org)을 창설하고 기존 인터넷에서 사용된 개념과 기술을 무선 서비스 플랫폼에 도입하여 무선 통신망에서도 인터넷을 자유롭게 이용하기 위해 발표된 기술이다[1,13].

### 2.1 WAP의 역사

WAP은 GSM(Global System for Mobile Communications), TDMA(Time Division Multiple Access), CDMA(Code Division Multiple Access), CDPD(Cellular Digital Packet Data) 등을 포함한 모든 무선 네트워크에 연결할 수 있는 모바일 컴퓨터용 아키텍처로 에릭슨, 모토롤라, 노키아, 언와이어드 플래닛(Phone.com) 등 이동통신 업체들이 1997년 결성한 WAP 포럼에서 개발했다[27].

이후 AT&T, 벨사우스 와이어리스 데이터, IBM을 포함해 세계 60여 개의 유력 통신, 컴퓨터 업체들이 WAP 규격을 지원하고 있다. 이와 관련, 핀란드의 노키아 등의 업체에서 최근 WAP 규격의 GSM 방식 휴대전화를, 한국의 삼성전자 등의 업체에서는 CDMA 방식의 휴대전화를 각각 선보이고 이를 통한 인터넷 접속 서비스에 적극 나서고 있다.

또한, 통신망 제공 업체들은 WAP 프로토콜을 통해 전자우편은 물론 날씨, 교통정보, 홈뱅킹(Home-Banking)과 같은 인터넷서비스를 제공하고 있다. 뿐만 아니라, 모바일 컴퓨터 업체들도 단말기에 채용되는 인터넷 검색용 브라우저(Micro Browser) 기능을 강화하기 위해 WAP을 채용하고 있는 실정이다[24].

WAP 포럼에서는 WAP에 대한 프로토콜 규격을 제정하여 발표하고 있으며 1.0, 1.1, 1.2를 지나

2.0을 발표하였다[13].

### 2.2 WAP의 기능 및 구조

WAP은 단순한 하나의 프로토콜이 아니라 무선통신에 관한 여러 가지의 프로토콜의 집합을 지칭하는 이름이다. 이것은 디지털 이동 전화기, 페이지(Pager), 휴대형 개인정보단말기(PDA: Personal Digital Assistant) 또는 그 밖의 무선 단말기들에게 인터넷 통신과 진보된 전화 서비스를 제공하는 프로토콜을 말한다. 이런 서비스를 제공하기 위해서는 기존 WWW 모델과는 다른 형태의 모델이 필요한데 그 이유는 휴대용 기기(hand held device)들이 기존의 고정된 거치형 컴퓨터와는 그 성능에 있어서 많이 부족하기 때문이다. 기존의 컴퓨터와 휴대용 기기간의 차이점으로는 낮은 성능의 CPU, 적은 양의 메모리, 낮은 전송률, 안정적이지 못한 연결, 안정적이지 못한 전원 공급, 컴퓨터에 비하여 상대적으로 열악한 입출력 장치 등이다[11].

WAP 모델을 도식화하면 [그림 1]과 같다.

WAE 사용자 에이전트(User Agent)는 사용자에게 특별한 기능을 제공하는 클라이언트 쪽의 소프트웨어로 이것은 WWW에서의 웹 브라우저와 같은 역할을 수행한다. WAP 게이트웨이는 프로토콜 변환 작업을 수행하며 클라이언트와 웹 서버 중간에서 WML를 인코딩 하거나 디코딩 한다. 오리진 서버(Origin Server)는 웹 서버와 같은

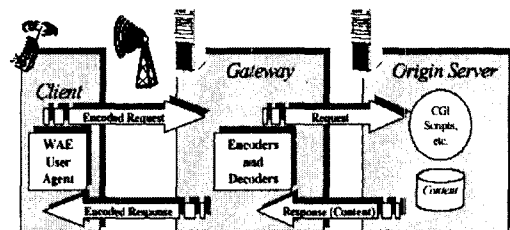


그림 1. WAP 모델

역할을 수행한다[2].

WAP의 특징은 낮은 대역폭 등 무선환경에 최적화된 응용 프로토콜이라는 점과, 바이너리 WML과 경량 무선 프로토콜 사용으로 제한된 처리 능력을 갖는 휴대용 기기에 최적의 솔루션 제공한다. 또한, HTML/XML 기반의 인터넷 표준 응용 및 프로토콜을 수용하고 있으며 이질적인 다양한 통신망에 독립적인 프로토콜이다.

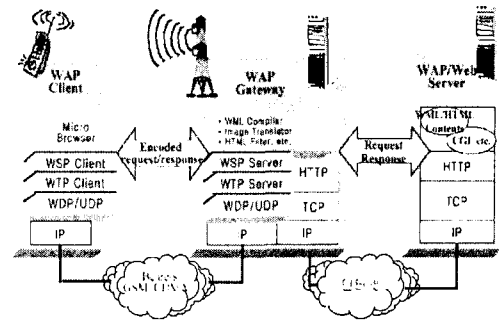


그림 3. 프로토콜 및 레이어 간의 통신

### 2.3 WAP의 구성요소

WAP 프로토콜 스택은 6개의 레이어 계층으로 구성되며 각 계층은 Application Layer(WAE), Session Layer(WSP), Transaction Layer (WTP), Security Layer(WTLS), Transport Layer(WDP) 그리고 베어러(Bearer)들로 이루어진다.

WAP의 아키텍처는 [그림 2]와 같다[1].

WAP 프로토콜 스택이 TCP/IP 모델에 기반한 HTTP(Hyper Text Transfer Protocol) 아키텍처의 각 레이어를 무선환경에 적합한 해당 WAP 레이어로 변환시킨 관계를 그림으로 나타내면 [그림 3]과 같다.

#### 2.3.1 WAE

WAE(Wireless Application Environment)는 WAP 포럼이 정한 클라이언트와 서버 사이의 일

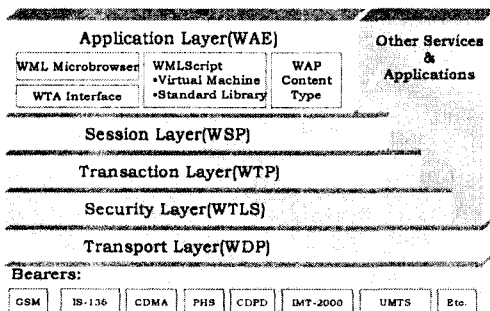


그림 2. WAP 아키텍처

반적인 프레임워크로 WAP 사용자 에이전트 (WAP 클라이언트 상에서 수행되는 WML 브라우저와 같은 소프트웨어)와 WAP 게이트웨이를 정의하고 있다. WAE는 URL(Unified Resource Locator)과 HTTP에 의존하는 WWW(World Wide Web)과 비슷하게 디자인되었다[2].

WAE의 대표적 구성요소는 다음과 같다.

- WML(Wireless Markup Language) - HTML과 비슷하나 휴대형 기기에 쓰이기 위한 라이트웨이트 마크업 언어(lightweight markup language)
- WMLScript - JavaScript와 비슷한 라이트웨이트 스크립트 언어(lightweight script language)
- Wireless Binary XML - WML 데크가 바이너리 형태의 단순한 표현(compact representation)으로 컴파일되는 방식에 대한 규약
- WTA(Wireless Telephony Application) - 전화서비스와 프로그래밍 인터페이스

WAE는 사용자 에이전트가 정보를 표현하는 방식과 사람과 기계 사이의 인터페이스(man-machine interface)는 규정하지 않고 있다. 이는 사용자 에이전트 제작 회사에서 각자의 방식으로 작성할 수 있도록 하였다.

### 2.3.2 WSP

WSP(Wireless Session Protocol)[5]는 HTTP와 같은 요청-응답(Request-Reply) 프로토콜을 구현하기 위하여 디자인되었는데 이 프로토콜을 이용하여 클라이언트가 Request를 생성하면 서버는 Reply를 통하여 응답한다[9]. WSP는 WAP의 어플리케이션 레이어에게 두 가지 세션 서비스들을 위한 일관된 인터페이스를 제공한다. 첫째는 WTP 레이어 위에서 작동하는 연결형 서비스(Connection-Oriented Services)이며, 다음은 WDP 레이어 위에서 작동하는 비연결형 서비스(Connection-less Services)이다.

WSP는 현재 브라우징 어플리케이션(Browsing Application - WSP/B)들을 위한 서비스들의 조합으로 구성되어 있다. WSP/B는 다음과 같은 기능들을 제공한다. HTTP/1.1을 무선 전송에 적합한 형태로 인코딩, 장기간 세션 상태, 세션 이동에서 세션의 일시 중단(Suspend)과 재개(Resume), 신뢰적인 혹은 비신뢰적인 데이터 푸시를 위한 공통적 기능, 프로토콜 특성 협의 등이다.

WSP 계열 프로토콜들은 상대적으로 긴 지연(long latency)을 가진 저대역 전송 네트워크(low-bandwidth bearer network)를 위해 최적화되어 있으며, WSP/B는 WAP 프락시(proxy)가 WSP/B 클라이언트와 표준 HTTP 서버를 연결하도록 설계되어진다.

### 2.3.3 WTP

WTP(Wireless Transaction Protocol)[6]는 데이터그램 위에서 작동하며 소규모 클라이언트(이동 단말기, PDA 등)에서 구현하는데 적합한 라이트웨이트 트랜잭션 지향 프로토콜(light weight transaction-oriented protocol)을 제공한다. WTP는 안전한 혹은 안전하지 않은 무선 데이터그램 네트워크 상에서 효율적으로 동작하며 다음과 같

은 특징들을 제공한다[10]. 트랜잭션 서비스의 세 가지 분류(비신뢰적 단방향 요청, 신뢰적 단방향 요청, 신뢰적 양방향 요청), 트랜잭션, 선택적 사용자간 신뢰성, 선택적 out-of-band data on acknowledgement, 보낼 메시지의 수를 줄이기 위한 PDU 연결과 지연된 acknowledgement, 비동기적 트랜잭션 등이다.

### 2.3.4 WTLS

WTLS(Wireless Transport Layer Security)는 산업 표준인 Transport Layer Security (TLS) Protocol에 기반한 보안 프로토콜이다. WTLS는 WAP 전송 프로토콜(WDP)과 함께 사용되며 협대역 통신 채널에서 사용되기 위해 최적화 되었다. WTLS는 또한 단말기 사이에 안전한 통신을 위해 쓰이곤 한다. 즉 전자 사업 카드 교환에서 인증 같은 것을 말한다.

어플리케이션들은 그들의 보안 요구들과 하부 네트워크의 특징에 따라 WTLS 기능들을 가능하게 하거나 불가능하게 선택적으로 할 수 있다(즉, 보안을 하위 레이어에서 이미 제공하고 있다면 생략할 수 있다)[13].

### 2.3.5 WDP

WDP(Wireless Datagram Protocol)[7]는 전송 레이어 프로토콜로 다양한 네트워크 형식들에 의해 지원되는 데이터 지원 베어러(bearer) 서비스 상에서 작동한다. 일반적인 전송 서비스처럼 WDP는 WAP의 상위 레이어 프로토콜에 일관적인 서비스를 제공하며 가능한 베어러(bearer) 서비스들 중 하나 위에서 투명하게 통신한다.

WDP 프로토콜들은 상위 프로토콜인 보안에 공통된 인터페이스를 제공하므로 세션과 어플리케이션 레이어들은 하부 무선 네트워크에 독립적으로 기능을 이용할 수 있다. 이는 하부 베어러의 지정된 특징들에 전송 레이어를 채택함으로써 이

뤄진다. 전송 레이어 인터페이스와 기본적 특징들을 일관적으로 유지함으로써, 전체적 상호 운영성은 중간 게이트웨이들을 이용하여 이루어 질 수 있다[11].

일반적인 WDP의 구조는 [그림 4]와 같다.

### 3. WAP 프로토콜 스택

WAP 프로토콜 스택은 WAP 클라이언트의 request가 들어오면 이를 해석한 후에 해당 요청에 대한 결과를 Web/WAP 서버로부터 받아들이고 이를 무선 전송에 알맞은 데이터 형식으로 인코딩하여 다시 WAP 클라이언트에게로 reply하여야 한다[1,11].

구현된 WAP 프로토콜 스택은 플랫폼에 독립적으로 작동될 수 있도록 자바 언어를 이용하여 구현하였으며 WDP, WSP 레이어 만을 이용하여 통신하는 WSP 비연결형 통신 기능과 WDP, WTP, WSP 레이어를 이용하는 WSP 연결형 통신 기능이 모두 수행 가능하다.

#### 3.1 시스템 구성도

게이트웨이는 크게 다음과 같은 다섯 부분으로 구성된다. 첫째, 자바 패키지 - 전체 게이트웨이에서 사용하는 기본 라이브러리로 JDK(Java

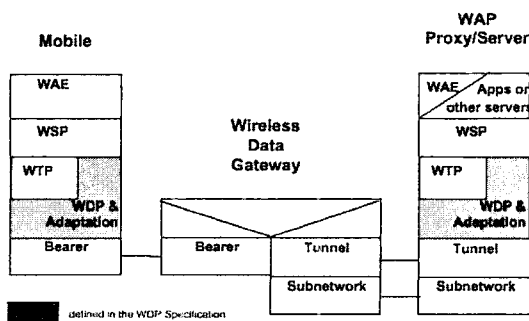


그림 4. 일반적 WDP 구조

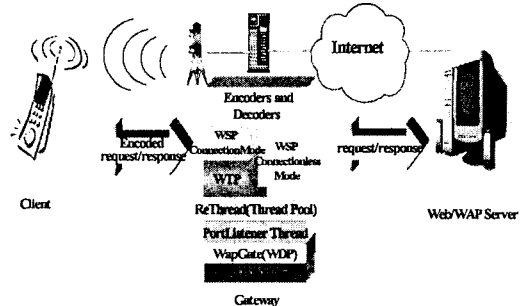


그림 5. WAP 게이트웨이 전체 구성도

Development Kit) 1.3을 사용하며 실제로 사용되는 패키지는 java.net(DatagramSocket, DatagramPacket, InetAddress), java.io(InputStream, BufferedInputStream), java.util(Timer, Vector), java.lang(Thread, String, Exception)이다. 둘째, WDP 레이어 및 관련 스레드 - WAP 클라이언트로부터 UDP 패킷을 송수신하는 부분으로 Runnable 인터페이스를 구현하여 스레드로 실행되는 메인 클래스인 WapGate와 동시에 여러 포트의 UDP 패킷을 감시하는 PortListener 스레드 그리고 Thread Pool인 ReThread로 구성된다. 셋째, WTP 레이어 - WSP 연결형 통신에 함께 사용되며 트랜잭션 관리 기능을 담당한다. 넷째, WSP 레이어 - WSP 레이어의 기능은 크게 연결형과 비연결형으로 구별되는데 전자는 WTP 레이어 함께 사용되어 WSP의 세션 관리 기능을 담당하고 후자는 WSP의 메소드 호출과 응답 그리고 웹 서버와의 통신을 담당한다. 다섯째, 인코더 & 디코더 - WML 문서를 무선 전송에 적합한 바이너리 형태의 WML로 상호 변환하는 기능을 수행하며 별도의 WMLCompiler 패키지로 구성된다.

#### 3.2 WAP 프로토콜 스택의 실행 과정

다음은 Wap 프로토콜 스택의 테스트 환경 및 Nokia Wap Toolkit을 통한 실행 과정을 나타낸 것이다.

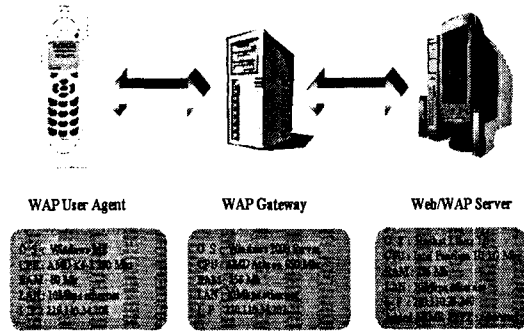


그림 6. WAP 프로토콜 스택의 실행과정

WAP 프로토콜 스택의 실험 모델은 WAP 사용자 에이전트로서의 휴대용 단말기와 WAP 게이트웨이, 그리고 Web/WAP 서버가 상호 통신하는 것이다[2]. 하지만 휴대용 단말기를 WAP 사용자 에이전트로 사용하는 것은 무선 통신망 라우터인 CSD 라우터(Circuit Switch Datagram Router)를 이용해 직접 실험을 해야하기 때문에 본 실험에서는 WAP 시뮬레이터인 노키아 WAP 툴킷 2.0을 WAP 사용자 에이전트로 사용하였다.

WAP 사용자 에이전트와 WAP 프로토콜 스택 사이에 전송된 패킷의 내용을 분석해 보면 다음과 같다.

- WAP 사용자 에이전트 ⇨ WAP 게이트웨이의 GET Request

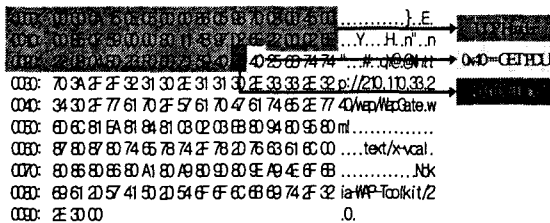


그림 7. 비연결형 요청 패킷

- WAP 게이트웨이 ⇨ WAP 사용자 에이전트의 Reply

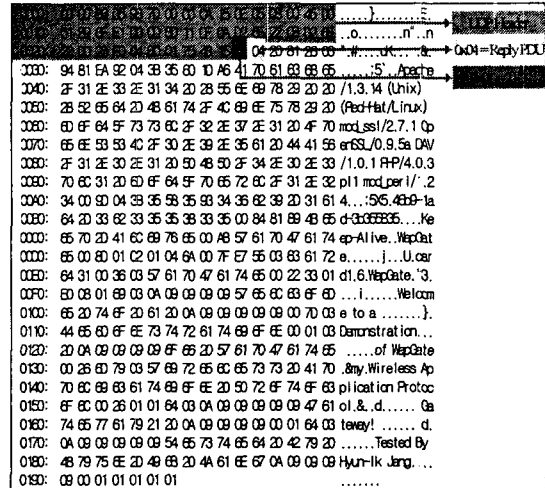


그림 8. 비연결형 응답 패킷

- WAP 사용자 에이전트 ⇨ WAP 게이트웨이의 Connect 요청

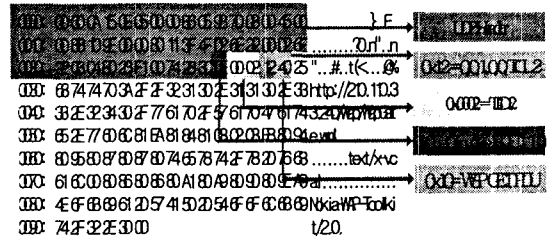


그림 9. 연결형 Connect 패킷

- WAP 게이트웨이 ⇨ WAP 사용자 에이전트의 Ack

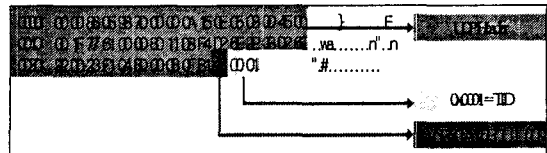


그림 10. 연결형 Ack 패킷

다음은 여러 개의 메뉴 항목 중에서 한 아이টে을 선택하여 수행하는 예제이다.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD WML
1.1/EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
  <card id="select card">
    <do type="accept" label="PRINT">
      <!-- accept 해당 card 로 이동한다 -->
      <go href="#print_select_value" />
    </do>
    <p>
      Which people do you know ?
      <!-- 원하는 아이টে을 선택한다 -->
      <select name="person" ivalue="1:3"
        multiple="true">
        <option value="Kim ByungGon">
          Kim ByungGon</option>
        <option value="Choi SuJong">
          Choi SuJong</option>
        <option value="KangHyunGyu">
          Kang HyunGyu</option>
        <option value="Kim DaeSung">
          Kim DaeSung</option>
        <option value="Kim DaeUng">
          Kim DaeUng</option>
        <option value="Park WanYoung">
          Park WanYoung</option>
        <option value="Ha HongMin">
          Ha HongMin</option>
      </select>
    </p>
  </card>

  <card id="print_select_value">
    <p>
      <!-- 선택된 아이টে을 모두 출력한다 -->
      Selected People : <br/>
      $person
    </p>
  </card>
</wml>
```

위의 프로그램을 실행하면 [그림 11]의 초기 화면이 나오고, Option 버튼을 누르면 Up 스크롤과

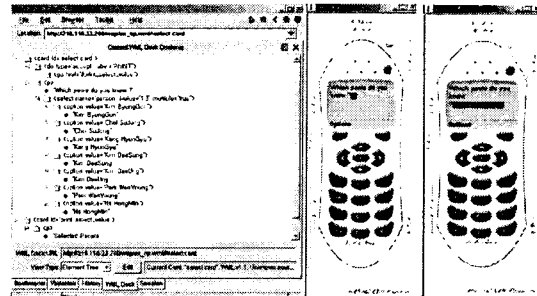


그림 11. 예제-아이টে을 선택 초기화면 결과화면

Down 스크롤 그리고 ok 버튼을 이용하여 아이টে을 고를 수 있다. 원하는 아이টে을 선택한 후, 화면 좌 하단의 OK 문자 아래에 있는 버튼을 누르게 되면 결과화면에서와 같이 선택된 아이টে을 출력 된다.

다음은 ASP와 연동하여 Email을 보내는 프로그램의 예이다.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD
WML 1.1/EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
  <card id="card_address_to">
    <do type="accept" label="enter">
      <go href="#card_address_from"/>
    </do>
    <p>
      <!-- 받는 사람의 E-mail 주소를 입력받음 -->
      To:
      <input name="address_to" format="*x"/>
    </p>
  </card>

  <card id="card_address_from">
    <do type="accept" label="enter">
      <go href="#card_title"/>
    </do>
    <p>
      <!-- 보내는 사람의 E-mail 주소를 입력받음 -->
      From:
```

```

<input name="address_from" format="*x"/>
</p>
</card>

<card id="card_title">
  <do type="accept" label="enter">
    <go href="#card_content"/>
  </do>
  <p>
    <!-- 보내는 글의 제목을 입력 받음 -->
    Title:
    <input name="title" format="*x"/>
  </p>
</card>

<card id="card_content">
  <do type="accept" label="send">
    <!-- accept 버튼을 누르면 http://localhost/
    mirzam/send.asp에 매개변수를 전달한다
    -->
    <go href="http://localhost/mirzam/send.asp?
    address=$address&title=
    $title&content=$content"/>
  </do>
  <p>
    <!-- 본문의 내용을 입력받음 -->
    Content:
    <input name="content" format="*x"/>
  </p>
</card>
</wml>

```

[ASP Source Code]

```

<%@Language = "VBScript"%>
<%
response.ContentType = "text/vnd.wap.wml"

Dim content,address_to,address_from,title
'선언한 변수에 wml로부터 넘겨진 값을 넣는다.
content = request("content")
address_from = request("address_from")
address_to = request("address_to")
title=request("title")

```

' 받은 정보를 CDONTS를 사용하여 SMTP Service로 보낸다.

```

Set ObjMail = CreateObject("CDONTS.Newmail")
ObjMail.From = address_from
ObjMail.To = adress_to
ObjMail.subject = title
ObjMail.Body = content
ObjMail.Send
set ObjMail = Nothing

```

'다시 wml파일로 돌아간다.  
Response.redirect "email.wml"  
%>

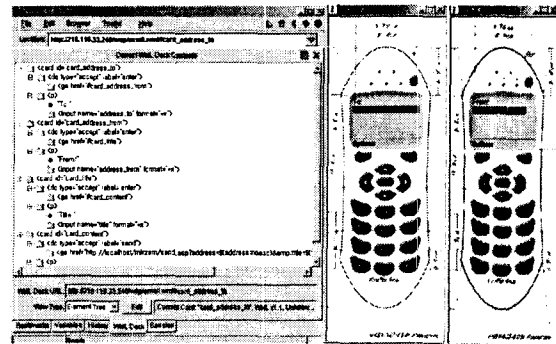
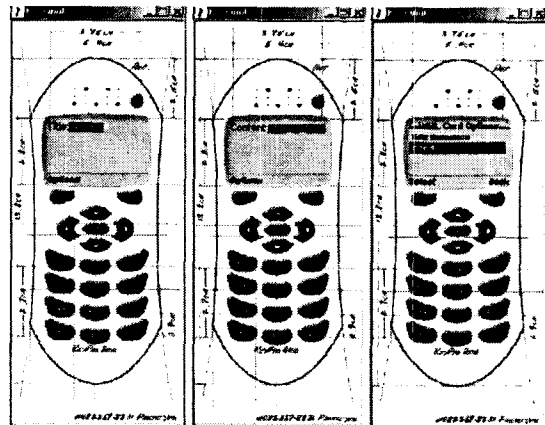


그림 12. 수신자의 e-mail 주소를 입력한 송신자의 e-mail 화면



e-mail 제목 입력 e-mail 내용 입력 e-mail 보내기

4. 결론

WAP 프로토콜 스택은 최근 그 수요가 폭발적



으로 증가하고 있는 무선 인터넷 관련 기술 중에서 WAP 포럼 중심의 WAP 기술과 WAP 기반 환경을 구축하는데 있어 필수적인 프로그램으로 WAP 규격 1.2를 기반으로 플랫폼 독립적인 Java 언어로 설계하고 구현하였다.

WAP 게이트웨이는 WAP 사용자 에이전트의 요청을 각각의 스레드를 생성하여 처리하였는데 매번 새로운 스레드를 생성할 경우에 발생하는 오버헤드와 메모리 낭비 문제를 재사용 가능한 Thread Pool인 ReThread 클래스를 이용하여 해결하였다. 또한, WSP 연결형 통신과 WSP 비연결형 통신 모듈을 구현하기 위해서는 버클리 소켓의 select() 함수와 같은 다중 포트 감시 기능이 필요로 한데 자바 언어에서는 이 기능이 제공하지 않기 때문에 이를 포트 Listener인 Datagram SocketThread 클래스를 통해 구현하였다. 또한, 자바 언어로 WAP 게이트웨이를 구현하였기 때문에 별다른 소스 코드의 수정없이 자바 가상 머신이 구현되어 있는 Windows, Unix, Linux 등 여러 플랫폼에서 실행할 수 있다. 또한, 클래스를 통한 상속과 같은 객체 지향 언어의 장점을 이용하여 코드의 재사용성을 높였다.

본 논문에서 구현한 게이트웨이가 보다 효율적으로 작동하기 위해서는 WAP 규격 1.2에서 추가된 Push 기술에 관한 지속적인 연구와 이에 대한 기능 구현이 요구된다.

### 참 고 문 헌

[1] WAP Forum, "Wireless Application Protocol Architecture Specification", Apr., 1998.  
 [2] WAP Forum, "Wireless Application Environment Specification", Mar., 2000.

[3] WAP Forum, "Wireless Markup Language Specification", Feb., 2000.  
 [4] WAP Forum, "WMLScript Language Specification", March, 2000.  
 [5] WAP Forum, "Wireless Session Protocol Specification", June, 2000.  
 [6] WAP Forum, "Wireless Transaction Protocol Specification", June, 2000.  
 [7] WAP Forum, "Wireless Datagram Protocol Specification", June, 2000.  
 [8] W. Richard Stevens, UNIX Network Programming Volume 1,2, 2ed, Prentice Hall, 1998.  
 [9] W. Richard Stevens, TCP/IP Illustrated Volume 1,2 Addison-Wesley, 1995.  
 [10] Elliotte Rusty Harold, Java Network Programming, 2E, O'Reilly, 2000.  
 [11] Charles Arehart et al. Professional WAP, WROX Press, 2000.  
 [12] The Kannel project, <http://www.kannel.org>.  
 [13] WAP Forum, <http://www.wapforum.org>.



이 양 선

- 1985년 동국대학교 전자계산학과(공학사)
- 1987년 동국대학교 대학원 컴퓨터공학과(공학석사)
- 1993년 동국대학교 대학원 컴퓨터공학과(공학박사)
- 1994년 3월~현재 서경대학교 컴퓨터공학과 교수
- 2000년 2월~현재 멀티미디어학회 이사
- 관심분야 : 프로그래밍 언어, 모바일 컴퓨팅, 분산 객체 시스템
- 저서 : 자바입문(생능), C기초+ $\alpha$ (피시북)