

HTML 필터 기능을 갖춘 WAP 게이트웨이 시스템 구축

(Construction of an WAP Gateway System Equipped with HTML Filtering)

박 기 현[†] 강 동 우^{**} 권 정 선^{***}

(KeeHyun Park) (Dong Woo Kang) (Jung Sun Kwon)

요 약 최근 이동 무선통신 환경에 대하여 점차 관심이 집중되고 있다. 이동 무선통신 환경을 구축하기 위하여 WAP(Wireless Application Protocol) 및 WAP 게이트웨이(Gateway) 등이 제안되었는데, WAP 게이트웨이는 이동 무선통신 환경에서도 기존의 유선 네트워크를 함께 이용할 수 있도록 하였다. 그런데 이런 시도는 새로이 WML(Wireless Markup Language)로 구축된 웹사이트들에게만 적용될 뿐이며, HTML로 구축된 기존의 많은 웹사이트들을 지원하지 못한다.

본 논문에서는 새로운 WML 웹사이트들뿐만 아니라, 기존의 HTML 웹사이트들의 내용도 이동 무선통신 환경에서 사용될 수 있도록 하기 위하여, HTML 필터 기능을 갖춘 WAP 게이트웨이 시스템을 설계하고 구현하였다. 이 시스템은 WAP 서버, 웹 클라이언트, 프로토콜 변환기, WML 인코더/디코더, HTML 필터 등으로 구성되어 있다. 이동 무선단말기 에뮬레이터를 사용한 실험에서 이 시스템을 사용한 결과, WML 및 HTML 웹사이트들에 대해서 수동적인 변환 없이도 이동 무선단말기에서 검색이 가능하였다.

Abstract There has been intensive interest on mobile wireless communication environments in recent years. In order to establish mobile wireless communication environments, WAP (Wireless Application Protocol) and WAP Gateways have been proposed, which make it possible to use existing wired networks in mobile wireless communication environments also. However, many existing HTML Web sites cannot be supported in this attempt, although new WML (Wireless Markup Language) Web sites can be supported.

In this paper, an WAP Gateway system equipped with HTML filtering is designed and implemented so that existing HTML Web sites as well as new WML Web sites can be accessed in mobile wireless communication environments. The system consists of a WAP server, a Web client, a protocol converter, a WML encoder/decoder, and a HTML filter. An mobile wireless terminal emulator is used in order to experiment the system in mobile wireless communication environments. The experiment shows that the mobile wireless terminal emulator can access freely HTML web sites as well as WML web sites without any human intervention with the help of the system.

1. 서 론

기존 유선 인터넷 환경은 데스크톱 이상의 컴퓨터와

높은 대역을 제공할 수 있는 네트워크를 기반으로 하지만, 이동 무선통신 환경에서는 현재의 데스크톱 기준으로 접근하기에는 전력 소모량, 메모리 및 디스플레이 크기, 전송 속도, 안정성 등에서 많은 어려움이 있다. 따라서 유선 인터넷의 표준을 그대로 사용하는데 한계가 있으므로, 기존 표준을 크게 벗어나지 않으면서 이동 무선통신 환경에 적합한 프로토콜을 만들려는 움직임이 일어났다. 이에 따라 여러 통신업체들을 중심으로 구성된 WAP(Wireless Application Protocol) 포럼에서 WAP을 제안하였다. WAP의 목적은 디지털 셀룰러 전화

· 본 연구는 2000년 중소기업청 기술혁신과제의 부분적인 지원을 받았다.

† 중신회원 : 계명대학교 컴퓨터및전자공학부 교수

khp@kmuucc.kmu.ac.kr

** 학생회원 : 계명대학교 컴퓨터공학과

dwkang@kmuucc.kmu.ac.kr

*** 비 회 원 : 계명대학교 컴퓨터공학과

vital@kmuucc.kmu.ac.kr

논문접수 : 2001년 4월 12일

심사완료 : 2001년 6월 1일

(PCS 등)와 무선 터미널에서의 인터넷 서비스 및 다른 종류의 무선 통신망 기술에서 운용될 수 있는 무선 프로토콜 규격과 콘텐츠, 응용기술을 개발하는데 있다. 또한 기존의 유선 네트워크를 함께 이용하기 위하여 WAP 게이트웨이(Gateway)를 제안하였다[9].

WAP 게이트웨이는 기존의 HTTP/TCP/IP[6] 기반의 패킷과 WSP/WTP/WDP[11,12] 기반의 패킷간의 상호 변환을 담당함으로써, 기존의 유선 네트워크도 이동 무선통신 환경의 인프라로 사용할 수 있게 한다. 그러나, 이런 시도는 통신 프로토콜 자체에만 적용될 뿐이지, 기존의 웹 콘텐츠에 대해서는 고려되지 않았다. 즉, WML(Wireless Markup Language)[10]로 구축된 웹사이트들의 콘텐츠를 이동 무선단말기들이 기존의 유선 네트워크 및 WAP 게이트웨이를 경유해서 검색할 수 있지만, HTML(Hyper Text Markup Language) 위주로 구축된 기존 웹사이트들의 콘텐츠들을 검색할 수는 없다.

따라서 본 논문에서는 새로운 WML 문서뿐만 아니라, 기존 웹사이트의 HTML 문서도 이동 무선단말기들이 검색할 수 있도록 HTML 필터 기능을 갖춘 WAP 게이트웨이 시스템을 설계하고 구축하였다. 이 시스템은 WAP 게이트웨이의 고유 기능뿐만 아니라, HTML 문서를 WML 문서로 자동 변환시켜 주기 때문에 이동 무선통신 환경을 위한 새로운 콘텐츠의 작성 없이도 기존 HTML을 이용한 서비스가 가능하도록 한다. 이 시스템은 WAP 서버, 웹 클라이언트, 프로토콜 변환기, WML 인코더/디코더, HTML 필터 등으로 구성되어 있다. 구현된 게이트웨이 시스템이 올바르게 동작하고 있는지를 실험하기 위해서 Phone.com에서 제공하는 인터넷폰 애플레이터[13]를 이용하였다. 이 실험을 통하여 WML 및 HTML 웹사이트들에 대해서 수동적인 변환 없이도 이동 무선단말기에서 검색이 가능함을 확인하였다.

본 논문의 구성은 다음과 같다. 2절에서는 WAP 및 WAP 게이트웨이의 구조 및 동작에 대하여 살펴본다. 3절에서는 본 논문에서 구축한 HTML 필터링 기능을 갖춘 WAP 게이트웨이 시스템의 구성과 각 구성 요소들의 설계와 구현에 대하여 설명한다. 인터넷폰 애플레이터를 이용한 게이트웨이 시스템의 실험은 4절에서 다룬다. 마지막으로 5절에서는 본 논문에 대한 결론 및 향후 연구 계획에 대하여 기술한다.

2. 배경

2.1 WAP 모델

WAP은 이동 무선통신 단말기와 유무선 네트워크 서버 사이의 통신을 가능하게 하기 위하여 다음과 같은

표준 컴포넌트들을 정의한다[1].

- 표준 이름 모델: 웹 표준 URL을 사용하며 기능도 동일하다.
- 콘텐츠 타입(Type): 웹 타입과 동일한 여러 타입들을 가진다.
- 표준 콘텐츠 형식: 웹 기술에 기반을 둔 display markup, 날짜 정보, 전자 카드 객체, 이미지, 스크립트 언어 등을 포함한다.
- 표준 통신 프로토콜: 이동 무선통신 환경에 적합하도록 설계된 WAP 프로토콜 스택(WSP, WTP, WDP 등)을 정의한다.

WAP은 무선 환경에 적합하도록 최적화하고 기타 관련 기술이 추가됐지만, 근본적으로는 기존의 유선 인터넷상의 클라이언트/서버 환경[2]과 유사하다. 그림 1은 WAP 모델을 보여준다. WAP 콘텐츠와 응용프로그램은 웹 콘텐츠와 유사한 형식으로 작성되며, 웹에서 사용하는 유선 통신 프로토콜들(HTTP/TCP/IP)을 사용하여 전송된다. 사용자는 이동 무선통신 단말기의 사용자 에이전트(Agent)인 마이크로 브라우저(Micro browser)를 통하여 웹서버에게 정보를 요청하게 되고, 이 요청은 무선 통신 프로토콜(WSP/WTP/WDP)에 의해서 전달된다. WAP 게이트웨이는 프록시(Proxy) 기술을 사용하여 유선 통신 프로토콜과 무선 통신 프로토콜간의 상호 변환을 담당함으로써, 유무선 통신을 원활히 할 수 있게 한다. 또한 인코딩/디코딩 작업을 통하여 통신 부하를 줄이는 역할도 한다.

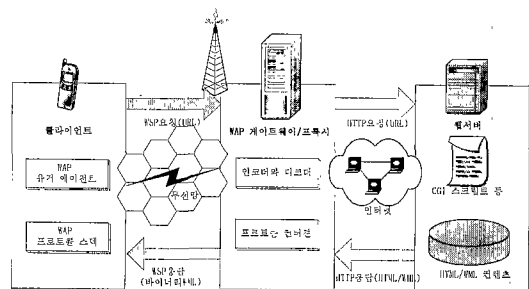


그림 1 WAP 모델

2.2 WAP의 프로토콜 스택(Protocol Stack)

WAP 구조는 빠르게 변화하는 이동 통신 디바이스의 응용 프로그램 환경을 위해 확장성이 뛰어나야 하기 때문에, WAP은 계층적 프로토콜 스택 구조로 이루어져 있다[8,9,11,12]. 그림 2는 WAP 프로토콜 스택의 계층들 및 이들에 대응되는 인터넷 프로토콜 스택의 계층들

을 보여준다. 각 계층은 상위 계층뿐만 아니라, 잘 알려진 인터페이스를 통해 다른 서비스나 응용프로그램도 접근할 수 있도록 디자인되었다. WAP 프로토콜 스택과 인터넷 프로토콜의 차이점은 단지 WAP 프로토콜 스택이 좁은 대역폭에 알맞게 설계되었다는 점과 CDMA 및 GSM 등과 같이 국가 및 망 사업자마다 다른 베어러(Bearer)에 맞게 설계되었다는 점이다.

2.2.1 WAE(Wireless Application Environment)

WAE의 주된 목적은 서비스 제공자와 개발자가 상호 대화할 수 있는 환경을 제공함으로써, 다양한 무선 플랫폼 위에서 작동할 수 있는 애플리케이션과 서비스를 능률적으로 구축할 수 있게 하는데 있다. WAP 브라우저 환경, WML, WML 스크립트, WTA(Wireless Telephony Application), WTAI(Wireless Telephony Application Interface), 그림, 전화번호, 달력정보 등의 데이터가 포함된 콘텐츠 형식이 포함된다.

2.2.2 WSP(Wireless Session Layer)

WSP는 세션 서비스를 위한 두 개의 인터페이스로 구성되며, WTP 위에 동작하는 연결형 서비스와 WDP 위에서 동작되는 비연결형 서비스 등이 있다. HTTP의 세션을 맺는 기능과 동일하며, 세션 헤더의 교환과 캐피빌리티 협상(Capability negotiation) 등 최적화된 콘텐츠 전달을 위한 상태를 제공한다. WSP는 좁은 대역폭과 긴 응답시간을 지닌 네트워크에서 작동할 수 있도록 최적화 되어 있으며 푸시, 중지, 재생 기능도 포함되어 있다.

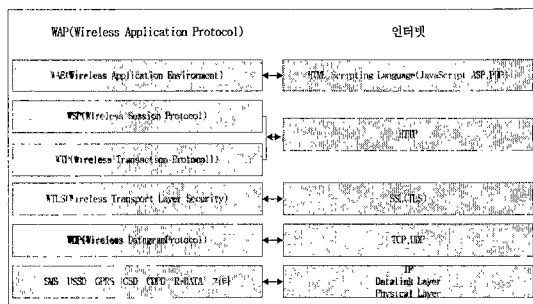


그림 2 WAP 및 인터넷 프로토콜 스택 계층

2.2.3 WTP(Wireless Transaction Protocol)

WTP는 WDP 서비스 위에서 실행되며, 인터넷 접속 전용 컴퓨터를 위한 비교적 간단한 트랜잭션을 위한 프로토콜로서, 3개의 클래스(신뢰할 수 없는 단방향 요청, 신뢰할 수 있는 단방향 요청, 신뢰할 수 있는 양방향요

청)를 지원한다. 또한 대역폭에서 벗어난 데이터에 대한 선택적인 승인(Acknowledgement) 기능, 전송할 메시지의 수를 감소시키기 위한 PDU(Protocol Data Unit)로의 연결과 지연의 통지 및 비동기 트랜잭션을 수행한다.

2.2.4 WTLS(Wireless Transport Layer Security)

WTLS는 공식적으로 SSL(Secured Socket Layer)로 알려진 산업표준형 TLS(Transport Layer Security)에 기반을 둔 보안 프로토콜이다. WAP의 트랜스포트 프로토콜과 함께 좁은 대역폭의 통신 채널에서 사용하도록 설계되어 있다. WTLS 기능은 데이터 무결성, 비밀 보장성, 사용자 인증, 서비스 중단 보호 기능 등이다. 응용 프로그램은 자신의 필요에 따라 WTLS의 항목들을 활성화시킬 수도 있고 비활성화시킬 수도 있다.

2.2.5 WDP(Wireless Datagram Protocol)

WDP는 좁은 대역폭에 적합한 비연결형 서비스를 제공한다. WDP에서 상위 계층 프로토콜들에게 일관된 인터페이스를 제공하기 때문에 상위계층 프로토콜은 무선 네트워크의 구성에 관계없이 자신의 기능을 독립적으로 수행할 수 있다. WDP에는 다양한 베어러 서비스(CDMA, GSM 등) 위에서 작동할 수 있도록, 각 서비스에 적합한 동작을 정의해 놓았다.

2.3 WML(Wireless Markup Language)

WML은 XML(eXtensible Markup Language)에 기반을 두고 개발한 언어로서, 대역폭이 좁은 이동 무선통신 환경에 적합하도록 설계되었다[10]. 따라서, WML의 문법 규칙들은 HTML과 약간 다른 특징을 보인다. WML은 XML의 부분집합으로서, XML의 문자 집합을 따르며 대소문자를 구분한다. WML 태그(Tag)는 그 사용이 HTML과 비슷하지만, WML은 시작 태그가 있으면 끝을 알리는 태그가 반드시 있어야 한다. 또한, 문서 정의를 위하여 XML의 DTD(Document Type Definition)[4]를 사용하며, WML 문서의 시작은 XML 문서 서문으로 시작해야 한다. HTML에서의 <html> 및 <body> 태그들은 WML에서의 <wml> 및 <card> 태그들에 각각 해당하지만, 1개의 <wml> 태그 속에 여러 개의 <card> 태그들이 나타날 수 있으며, <card> 태그들간의 이동도 가능하다.

HTML의 확장 버전에서 정의되어 있는 태그들(<frame>, <table> 속의 <table>, 등)은 정의되어 있지 않는데, 이것은 이동 무선통신 단말기 및 네트워크의 물리적 및 환경적인 제약을 고려한 것이라고 생각된다. WML 문서의 기본적인 구성은 표 1과 같으며, 그림 3은 WML의 전체적인 구조를 보여준다.

표 1 WML 문서 구성

```

<?xml version="1.0"?> <!-- WML 문서 시작(XML의 DFD)
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD wml 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml> <!-- 카드 덱(deck)의 시작
<card id="card1"> <!-- 첫 번째 카드의 시작
  <p>
    Hello World. <!-- 첫 번째 카드 내용
  </p>
</card> <!-- 첫 번째 카드의 끝
<card id="card2"> <!-- 두 번째 카드의 시작
  ... <!-- 두 번째 카드 내용
</card> <!-- 두 번째 카드의 끝
</wml> <!-- 카드 덱의 끝
    
```

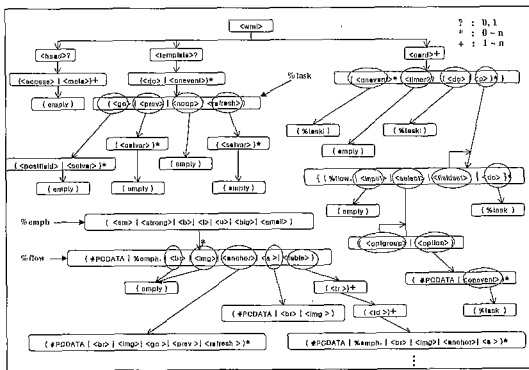


그림 3 WML의 태그 구조

3. WAP 게이트웨이(Gateway) 시스템의 설계 및 구현

WAP 게이트웨이는 WAP 프로토콜(WSP/WTP/WDP)을 사용하는 이동 무선통신 네트워크와 인터넷 프로토콜(HTTP/TCP/IP)을 사용하는 유선통신 네트워크를 연결 시키기 위하여 필요한 조치(프로토콜 변환, 인코딩/디코딩 등)를 취한다. 본 논문에서는 이러한 기능을 가진 WAP 게이트웨이에 HTML 필터 기능을 추가함으로써, 기존 웹서버의 HTML 문서를 WML 문서로의 자동변환이 가능하도록 하였다. 본 논문에서 구현한 HTML 필터 기능을 갖춘 WAP 게이트웨이 시스템을 사용하면 이동 무선통신 네트워크를 위한 WML 콘텐츠를 별도로 작성하지 않고서도 기존 웹사이트의 HTML 콘텐츠를 이용한 서비스가 가능하므로, 새로운 콘텐츠 작성에 대한 시간과 노력을 상당히 절약하는 효과가 있다.

3.1 WAP 게이트웨이 시스템의 구성

본 논문에서 제안하는 WAP 게이트웨이 시스템은 그림 4와 같이, WSP 서버 부분, HTTP 클라이언트 부분,

프로토콜 변환기 부분, HTML 필터 부분, WML 인코더/디코더 부분 등으로 구성된다.

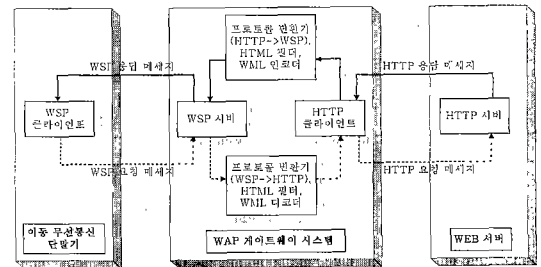


그림 4 WAP 게이트웨이 시스템의 구성 요소들

3.2 WSP 서버 부분

WAP 게이트웨이 시스템의 WAP 서버 부분은, WAP 게이트웨이 시스템이 이동 무선통신 단말기에 대해서 WAP 서버 역할을 할 수 있도록 WSP 형식에 따라 단말기와 응답과 요청을 수행한다. 이동 무선통신 단말기의 WAP 브라우저로부터 요청을 받아서 WAP 프로토콜 스택의 WSP 계층 역할을 하는 부분이다. WSP 계층 작업이 끝나면, WSP 서버는 프로토콜 변환기로 메시지를 보낸다.

WSP는 상위 또는 하위 단계에서 발생하는 이벤트(Event), 이벤트에 의해 발생되어 전송되는 PDU(Protocol Data Unit), PDU에 의해 야기되는 세션의 생성, 메소드(Method)의 처리, 다음에 행할 동작 등을 정의한다. 이벤트는 WSP의 상위 계층(WAE) 또는 하위 계층(WTP)에서 발생되며 WSP의 동작을 정의한다. PDU는 전송되는 데이터의 집합을 정의한다. 표 2와 표 3은

표 2 WAP 서버 부분의 이벤트들

이벤트	설명
S-Connect.ind	세션 연결에 사용되며, 연결의 성공 여부를 클라이언트에게 통지
S-Connect.res	캐피탈리티 협상을 제공
S-Disconnect.ind	세션 종료에 사용되며, 종료의 성공 여부를 클라이언트에게 통지
S-Suspend.ind	세션의 일시적 중지 사용. 세션 재개 혹은 종료에 의해 중지 해제
S-Resume.ind	세션 재개에 사용
S-Resume.res	세션 재개에 사용
S-MethodInvoke.ind	서버에 의해 수행될 동작을 요구하는 데 사용
S-MethodInvoke.res	동작 요구에 대한 응답을 클라이언트에게 돌려주기 위해 사용
S-MethodResult.req	동작 요구에 대한 응답을 클라이언트에게 돌려주기 위해 사용
S-MethodResult.cnf	동작 요구에 대한 응답을 클라이언트에게 돌려주기 위해 사용
S-MethodAbort.ind	아직 종료되지 않은 동작 요구를 취소하는데 사용

표 3 WAP 서버 부분의 PDU들

PDU		설 명
Session 관련	Connect	세션 연결을 위해 전송
	Connect Reply	Connect PDU의 응답으로 전송
	Redirect	Connect PDU에 대한 응답으로 전송 바뀐 주소의 서버에 연결시키거나 세션 연결시의 부하 균형을 위해 사용
	Disconnect	세션 종료를 위해 사용
Method 관련	Get	서버에게 Request 콘텐츠를 전송하지 않는 확장 메소드들을 위해 사용 HTTP의 GET, OPTION, HEAD, DELETE, TRACE 메소드들을 위해 사 용
	Post	서버에 Request 콘텐츠를 전송하는 확 장 메소드들을 위해 사용 HTTP의 POST, PUT 메소드들을 위 해 사용
	Reply	요청에 대한 일반적인 응답 PDU

표 4 WSP 이벤트 정의(예)

```

struct S-Connect.ind
{
  AddressType Address,
  HeaderType Client Headers,
  CapabilityType Requested Capabilities,
  integer session ID)

struct S-Disconnect.ind
{
  integer Reason code,
  integer Redirect Security,
  AddressType Redirect Addresses,
  HeaderType Error Headers
  String Error Body,
  integer session ID)
    
```

표 5 WSP PDU의 정의(예)

이름	타입	데이터 값
Type	uint8	PDU 타입 정의 값
Version	uint8	프로토콜 버전
CapabilityLen	uintvar	Capability 필드의 길이
HeadersLen	uintvar	Header 필드의 길이
Capabilities	octets	S-Connect.req 이벤트의 요청 케이 피빌리티의 값
Headers	octets	S-Connect.req의 Client 헤더의 값

WSP 서버 부분에서 정의되는 이벤트들과 PDU들을 각
각 보여준다. 이들의 예는 각각 표 4와 표 5와 같다.

이벤트와 PDU에 의한 일반적인 WSP의 세션 연결 메
카니즘을 그림 5에서 보여준다. 클라이언트에서 발생
한 S-Connect.req 이벤트에 의해 Connect PDU가 생성되
어 서버로 전송이 이루어지면, 서버에서는 S-Connect.

ind 이벤트에 의해 세션 연결 요청을 받게 된다. 서버는
S-Connect.res 이벤트에 의해 ConnectReply PDU를 클
라이언트에게 전송하게 되며, 클라이언트는 S-Connect.
cnf 이벤트를 통해서 세션 연결을 확인하게 된다.

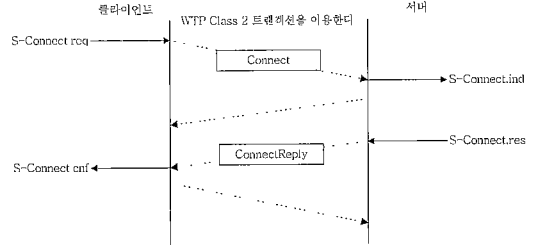


그림 5 WSP의 일반적인 세션 생성 메카니즘

3.3 프로토콜 변환기 부분

프로토콜 변환기 부분은 WSP와 HTTP 프로토콜의
상호 변환을 수행한다. 무선 이동통신 단말기에서 요청
한 WSP 요청 메시지를 HTTP 요청 메시지로 변환을
하며 반대로 웹 서버에서 오는 HTTP 응답 메시지를
WSP 응답 메시지로 변환을 하게 된다.

3.3.1 프로토콜 변환 방법

WSP 프로토콜과 HTTP 프로토콜간의 비교는 대략
표 6과 같다. 무선환경에서 보다 효율적인 전송을 위하
여 WSP 프로토콜은 축약된 형태를 취하기 때문에,
HTTP 프로토콜에서 WSP 프로토콜로의 변환은 HTTP
헤더(Header) 정보를 파싱(Parsing)하여 각 파라미터와
값을 분석한 후 WSP로의 변환을 수행한다. 변환된
WSP 헤더 필드들은 일반적인 정보, 요청 정보, 응답 정
보 또는 엔티티(Entity) 정보를 포함하며 인코딩 과정에
서 잘 알려진 토큰들과 데이터 값들은 이진값들로 매핑
이 되어지며 불필요한 정보들은 제거된다.

표 6 HTTP 프로토콜과 WSP 프로토콜

	HTTP	WSP
요청 메시지 (WSP-> HTTP)	<method><resource identifier> <HTTP version><crlf> [<Header> : <value>] <crlf> [<Header> : <value>] <crlf> blank line <crlf> [Entity body]	URLen(uintvar) HeadersLen(uintvar) URI(UriLen octets) ContentType(multiple octets) Headers(octets) Data(multiple octets)
응답 메시지 (HTTP ->WSP)	<HTTP version><result code> [<explanation>]<crlf> [<Header> : <value>] <crlf> [<Header> : <value>] <crlf> blank line <crlf> [Entity body]	Status(uint8) HeadersLen(uintvar) ContentType(multiple octets) Headers(octets) Data(multiple octets)

3.3.2 프로토콜 변환 예

WSP 요청 메시지에서 HTTP 요청 메시지의 변환과 HTTP 응답 메시지의 WSP 응답 메시지로의 변환의 예를 표 7 및 표 8에서 보여주고 있다.

표 7 WSP 요청 메시지의 HTTP 요청 메시지로의 변환(예)

WSP 요청 메시지		변환된 HTTP 요청 메시지
메시지	설명	
0x40	Type(Get)	Get http://www.kmu.ac.kr HTTP 1.1 Accept:text/html User-agent:netscape 4.0 <method><resource identifier><HTTP version>
0x0D	URLLen(14 octet)	
www.kmu.ac.kr	URL(www.kmu.ac.kr)	
0x00		
0x80	Headers(Accept)	
0x82	(text/html)	

표 8 HTTP 응답 메시지의 WSP 응답메시지로의 변환(예)

HTTP 응답 메시지	변환된 WSP 응답 메시지	
	메시지	설명
HTTP/1.1 200 ok	0x04	Type(Reply)
Server:NCSA/1.3	0x02	Status(200)
Mime_version:1.0	0x0B	HeaderLen(14)
Content_type:text/html	0x91	Content_Type(Content-Type)
Content_Length :58	0x82	(text/html)
<html>	0xA2	Headers(server 0x26)
...	NCSA/1.3 0x00	(NCSA/1.3)
<html>	0x8D	(Content-length)
...	0x3A	(58)
<html>	<html>	data(<html>)
...
<html>	</html>	(</html>)

3.4 HTTP 클라이언트 부분

HTTP 클라이언트 부분은, WAP 게이트웨이 시스템이 웹서버에 대하여 웹클라이언트 역할을 할 수 있도록 HTTP 형식에 따라 웹서버와 요청과 응답을 수행한다. 프로토콜 변환기를 거친 요청 메시지를 웹서버로 보내어 응답 메시지를 받은 후 프로토콜 변환기로 응답 메시지를 되돌려 보낸다. 또한 HTTP 클라이언트 부분에 에러 검색 기능을 갖추어서 불필요한 통신이 이루어지지 않도록 한다[3]. 본 논문에서는 WAP 게이트웨이 시스템과의 연결을 제외하고는, 기존의 HTTP 클라이언트를 그대로 사용했으므로 자세한 설명은 생략한다.

3.5 HTML 필터 부분

WAP 게이트웨이 시스템의 HTML 필터 부분은 HTML 문서를 WML 문서로 변환시킨다. 웹서버로부터의 문서는 HTML 형식으로 표현되어 있으나, 이동 무선통신 단말기에는 WML 브라우저가 장착되어 있으므로, WML 형식에서의 변환이 필요하다. HTML 필터는 그림 6과 같이 주 모듈(Main module), 파싱 모듈

(Parsing module), 변환 모듈(Converting module) 등으로 구성된다. 주 모듈은 HTML 필터의 입력과 출력, 그리고 다른 모듈들과의 중간 매개 역할을 한다. 파싱 모듈은 주 모듈로부터 입력 데이터를 받아 HTML 형식에 따라 파싱을 하는데, 태그 이름, 속성(Attribute) 이름, 속성 값, 태그에 포함된 텍스트까지 한꺼번에 처리된다. 변환 모듈에는 파싱된 데이터를 WML 문법에 맞게 실제 변환을 행하는 메소드들이 구현되어 있다.

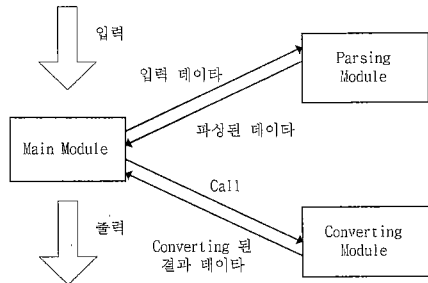


그림 6 HTML 필터의 모듈 구조

그림 7은 HTML 필터의 상태전이도(State transition diagram)의 일부를 보여준다. 그림에서 스캔하는 입력 문자열은 화살표 위에 위치하고, 상태전이에 따른 동작은 원으로 표시된 상태에 연결된 사각형 안에 기술되어 있다.

앞에서도 언급했듯이, 사용 환경의 차이 때문에 HTML 태그들과 WML 태그들 사이에는 다른 점들이 존재하므로, HTML 필터를 구현할 때 HTML의 모든 태그들을 완전하게 WML 태그들로 변환할 수가 없다. HTML 태그들 중에서 일대일로 완전하게 WML 태그들로 변환되는 것들이 있고(표 9 참조), 약간 변형되게 변환할 수밖에 없는 것들도 있다(표 10 참조). 또한, 현재의 WML 태그들로는 표현할 수 없는 것들이 있는데, 본 논문에서는 이런 HTML 태그들을 제외하고는 모두

표 9 태그 변환표(일대일 대응)

HTML 태그	변환된 WML 태그
<html>	<wml>
<body>	<card>
<address>	<i>
<cite>	
<input>	<postfield>
<q>	<i>
<head>, <meta>, <a>, <big>, <option>, , , , <select>	변환없음

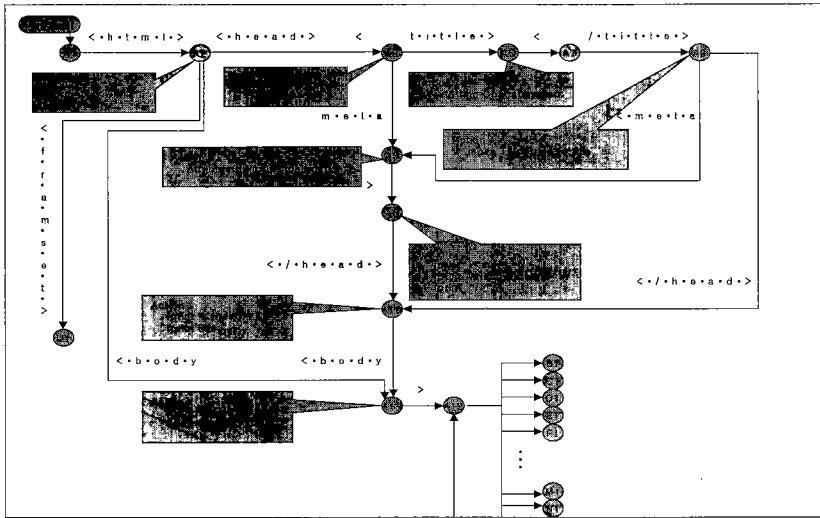


그림 7 HTML 필터의 상태전이도

구현하였다. 표현할 수 없는 태그들로서는 <applet>, <comment>, <embed>, <hr>, <map>, <marquee>, <script>, <basefont> 등이다.

표 10 태그 변환표(일부 변형 변환)

HTML 태그	변환된 WML 태그
<form>	<do> 및 <go>의 조합
<h1>, <h2>...	문자 크기 무시하고 출력
 혹은 	(중첩 리스트를 위해서) 리스트의 레벨 번호 삽입한 후, 리스트 내용 출력
 혹은 <td>	해당하는 리스트의 레벨만큼 공백을 출력한 후, 리스트 내용 출력
<table>	WML에서도 지원되는 태그이지만, 단말기의 협소한 출력화면 때문에 과 같은 방법으로 처리
<title>	"TITLE : title 내용" 출력
<frame>	각 frame에 대해서 "FRAME 번호" 출력한 후, <a> 추가
	"IMAGE : optional image 내용" 출력
<area>	이미지 문자 'a'를 출력하고 <a> 추가
<dt>	"* 내용" 출력
<dl>	"○ 내용" 출력
<dd>	"o 내용" 출력
<bold>, <caption>, <center>, <code>, <div>, , <kbd>, <listing>, <nobr>, <pre>, <person>, <samp>, <small>, <strike>, <sub>, <sup>	"내용"만 출력

3.6 WML 인코더 및 디코더 부분

무선통신의 전송 효율을 높이기 위하여 WML은 WSP에 의해 전송되기 전에 인코딩된다. 인코딩은 알려진 태그들과 문자들을 WAP 포럼에서 정의되어진 테이블[10]에 의하여 인코딩한다. WSP 클라이언트 측에서는 인코딩 된 WML을 다시 디코딩하여 이동 무선통신 단말기 브라우저에 출력을 하게 된다. 인코딩 및 디코딩은 테이블 참조에 의한 단순한 변환이므로, 여기서는 자세한 설명을 생략한다.

4. 실험

4.1 개발 및 실험 환경

그림 8은 본 논문의 WAP 게이트웨이 시스템을 위한 개발 및 실험 환경을 보여준다. Windows NT 서버에 설치된 Phone.com의 인터넷폰 에뮬레이터가 이동 무선통신 단말기의 역할을 한다. 이 서버에는 UP 브라우저도 설치되어서 단말기의 마이크로 브라우저 역할을 하게 한다.

HTML 필터가 있는 WAP 게이트웨이 시스템은 레드햇 리눅스 6.2가 설치되어 있는 리눅스 서버에 설치하였다. 비록, Windows NT 서버와 리눅스 서버가 유선으로 연결되어 있으나, Windows NT 서버의 인터넷폰 에뮬레이터가 WSP만을 인식하므로, WAP 이동 무선통신 환경이 조성되었다고 할 수 있다. 마지막으로, 웹서버는 아파치 웹서버를 설치한 리눅스 서버에 설치하였다.



그림 8 WAP 게이트웨이 시스템 개발 및 실험 환경

4.2 실험 결과

실험은 두 가지 형태로 이루어졌다. 첫 번째는, 이동 무선통신 단말기(인터넷폰 에뮬레이터)의 UP 브라우저가 요청한 자료가 웹서버에 WML 문서로 저장되어 있는 경우이다. 이럴 경우에는, HTML 필터를 거치지 않고 전송된다. 웹서버에서 WAP 게이트웨이 시스템으로 보내는 메시지의 헤더에 관련 정보가 수록되게 하였으므로, 구별이 쉽게 된다. 그림 9는 이런 예를 보여준다.

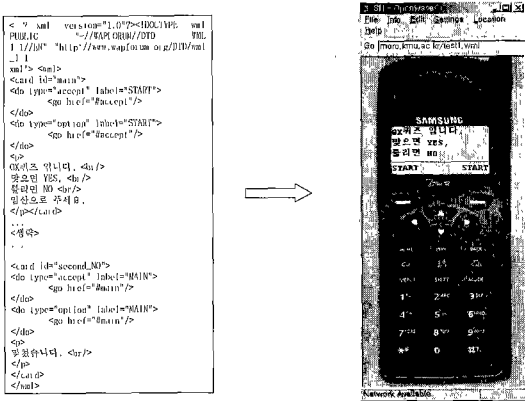


그림 9 WML 문서를 이동 무선통신 단말기에서 검색한 결과

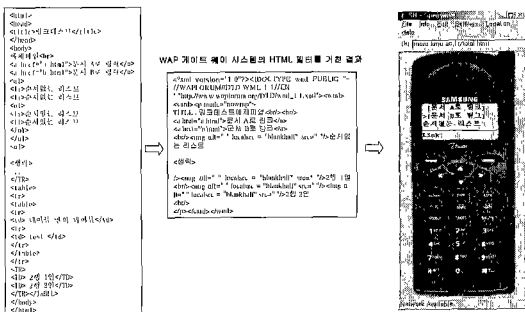


그림 10 HTML 문서를 이동 무선통신 단말기에서 검색한 결과

두 번째는, 이동 무선통신 단말기가 요청한 자료가 웹 서버에 HTML 문서로 저장되어 있는 경우이다. 이럴

경우에는, HTML 필터를 거친 후 전송된다. 그림 10에서는 HTML 문서가 HTML 필터를 거쳐서 WML 문서로 바뀐 후, 단말기로 전송되어 화면에 나타나는 예를 보여준다. 그림 11은 일반 웹사이트에 접속하여 얻은 HTML 문서를 HTML 필터를 거친 후, 단말기의 화면에 표시하는 예를 보여준다.

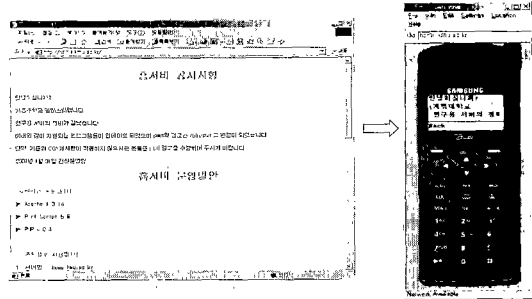


그림 11 웹사이트의 문서를 이동 무선통신 단말기에서 검색한 결과

5. 결론 및 향후 과제

본 논문에서는 HTML 필터를 탑재한 WAP 게이트웨이 시스템을 설계하고 구현하였다. 이 시스템은 WSP 서버 부분, HTTP 클라이언트 부분, 프로토콜 변환기 부분, HTML 필터 부분, WML 인코더/디코더 부분 등으로 구성된다. WSP 서버 부분(HTTP 클라이언트 부분)은 WAP 게이트웨이 시스템이 이동 무선통신 단말기(웹서버)에 대해서 WAP 서버(웹클라이언트) 역할을 하게 한다. 프로토콜 변환기 부분은 이동 무선통신 네트워크와 유선 인터넷을 원활히 연결하기 위하여 관련 프로토콜들을 변환하는 작업을 수행한다. HTML 필터 부분은 HTML 문서를 WML 문서로 변환함으로써, 일반 웹서버의 HTML 문서를 변경하지 않고서도 이동 무선통신 단말기에서 검색할 수 있게 한다. 통신의 효율을 위해서 인코더/디코더 부분이 있다. 본 시스템을 구현한 후, Phone.com의 인터넷폰 에뮬레이터 및 UP 브라우저 등을 이용하여 실험한 결과, 모든 변환이 순조롭게 이루어짐을 확인했다.

그러나, 본 WAP 게이트웨이 시스템에 부족한 기능들을 나열하면 다음과 같다. 첫 번째, 인터넷폰 에뮬레이터의 한계 때문에 대용량의 HTML 문서를 소화할 수 없었다. 이것은 HTML 문서를 여러 세그먼트들로 나누어서 연속적으로 처리할 수 있는 기능이 추가되면 해결할 수 있을 것이다. 두 번째, WML 및 UP 브라우저의

한계 때문에 멀티미디어 자료에 대한 변환이 불가능했다. 이것은 WAP 포럼에서 제안하고 있는 WBMP (Wireless BitMaP) 포맷으로의 자동 이미지 변환 기능이 추가하면 이미지 자료에 대한 해결 방안은 될 수 있으리라고 생각된다. 이 해결 방안이 성공적으로 구현되기 위해서는 WML 및 브라우저 등이 WBMP 포맷을 먼저 지원해야 할 것이다. 마지막으로, 현재 HTML 스크립트 언어에 대한 처리가 미흡하기 때문에, 앞으로 이 부분도 개선하고자 한다.

참 고 문 헌

[1] C. Arehart, Professional WAP, WROX Press, July 2000.
 [2] Douglas E. Comer, Computer Networks And Internets, Prentice Hall, 1997.
 [3] R. Fielding, J. Getty, T. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol HTTP/1.1," Network Working Group Request for Comments: 2068, January 1997.
 [4] M. Leventhal, D. Lewis, M. Fuchs, Designing XML Internet Application, Prentice Hall, 1998.
 [5] C. Perkins, "IP Mobility support," Network Working Group Request for Comments: 2002, October 1996.
 [6] W. Richard Stevens, TCP/IP Illustrated, Volume 3, Addison-Wesley, 1996.
 [7] W. Richard Stevens, Unix Network Programing, Prentice Hall, 1991.
 [8] WAP Forum, "Wireless Application Protocol Wireless Application Environment Specification Version 1.1," SEPC-WAPArch, May 1999.
 [9] WAP Forum, "Wireless Application Protocol Architectures Specification," SPEC-WAPArch, April 1998.
 [10] WAP Forum, "Wireless Application Protocol Wireless Markup Language Specification Version 1.1," SPEC-WAPArch, June 1999.
 [11] WAP Forum, "Wireless Application Protocol Wireless Session Protocol Specification," SPEC-WAPArch, November 1999.
 [12] WAP Forum, "Wireless Application Protocol Wireless Transaction Protocol Specification," SEPC-WAPArch, June 1999.
 [13] UP.SDK Developer's Guide, <http://developer.phone.com/html/doc/41/devguide/>, Openwave System Inc.
 [14] 강동우, 박기현, "WAP 게이트웨이에 대한 연구", 정보처리학회 추계학술논문집, 제7권, 제2호, pp.1063-1066, 2000년 10월.
 [15] 권정선, 박기현, "무선통신 환경을 위한 HTML 필터

구축", 정보처리학회 추계학술논문집, 제7권, 제2호, pp.1561-1564, 2000년 10월.
 [16] 홍준호, 송건철, 김정석, About WAP, 영진닷컴, 2001년 1월.



박 기 현

1979년 2월 경북대학교 전자공학과 졸업 (전자계산학과 전공, 공학사). 1981년 2월 한국과학기술원 전자계산학과 졸업 (이학석사). 1990년 9월 미국 Vanderbilt 대학교 전자계산학과 졸업(공학박사). 현재 계명대학교 컴퓨터및전자공학부 교수. 관심분야는 운영체제, 병렬처리, 성능분석임.



강 동 우

1999년 2월 계명대학교 컴퓨터공학과 졸업. 2000년 3월 ~ 현재 계명대학교 대학원 컴퓨터공학과 석사과정 재학 중. 관심분야는 무선통신 프로토콜, 네트워크 소프트웨어 등



권 정 선

2000년 2월 계명대학교 컴퓨터공학과 졸업. 2000년 3월 ~ 현재 계명대학교 대학원 컴퓨터공학과 석사과정 재학 중. 관심분야는 네트워크 소프트웨어, 보안 등