

WAP(Wireless Application Protocol) 에뮬레이터(Emulator)의 설계 및 구현

최철용⁰ 장지산 신화종 김상국 신동규 신동일
세종대학교 컴퓨터공학과
{cwchoi, jsjang, shinhj, kimsk, shindk, dshin}@gce.sejong.ac.kr

An Design and Implementation of WAP(Wireless Application Protocol) Emulator

Chul-Oong Choi, Ji-San Jang, Hwa-jong Shin, Sang-Kuk Kim,
Dong-kyoo Shin, Dong-il Shin
Department of Computer Engineering, Sejong University

요 약

무선 인터넷 시장의 급속한 발전으로 인해 콘텐츠에 대한 사용자들의 다양한 요구가 증가되고 있고, 그로인해 WML과 WML Script를 개발자가 실제와 같은 상황에서 편리하게 개발할 수 있는 환경이 필요하게 되었다. 본 논문에서는 CP(Content Provider)들이 쉽고 편리하게 WML(Wireless Markup Language) 콘텐츠(Content)를 개발, 브라우징하기 위한 WAP(Wireless Application Protocol) Emulator의 설계와 구현에 대하여 기술한다. WAP Emulator는 WML 파싱(parsing) 모듈과 인터페이스(Interface)모듈, 그리고 Data Send/Receive 모듈로 구성되며, 디코딩(Decoding)된 WML 문서를 파싱하여 브라우저로 보여준다

1. 서론

WAP(Wireless Application Protocol)[1]을 기반으로 한 무선 인터넷의 급속한 성장으로 인해 휴대용 전화기나 PDA(Personal Digital Assistant)와 같은 무선 터미널을 이용한 다양한 정보 습득이 언제 어디서나 가능하게 되었고, 그로 인해 휴대전화 및 관련 휴대용 무선장비를 이용하는 웹 콘텐츠 이용자의 급속한 증가를 가져왔다. WML(Wireless Markup Language)[2]는 이러한 웹 콘텐츠를 개발하기 위해 만들어졌으며, 휴대용 무선장비에서의 효과적인 웹 접근을 위한 언어인 HDML(Handheld Device Markup Language)[3]을 기반으로 만들어 졌다. HDML의 기본 구조인 데크(Deck)와 카드(Card)단위의 동작 스키마를 채택하여 데크와 카드들 간의 하이퍼 링크(hyper link)등 HDML의 작동원리를 그대로 승계하고 있으며, 현재 WML 버전 1.2까지 발표되었다.

이용자에게 유용하고 매력적인 방법으로 진보된 서비스를 제공하면서, 이용자의 요구를 충족시키고 다양한 종류의 콘텐츠 개발을 손쉽게 할 수 있는 각종 애플리케이션이 필요하게 되었다. 그리고, 이러한 휴대용 무선장비를 위한 웹 콘텐츠 제공 서비스는 현행 휴대전화 서비스뿐 아니라 차세대서비스로 일컬어지는 IMT2000에

서와 같은 멀티미디어를 중심으로 하는 통신서비스에서는 필수적인 기능이라 하겠다. 상용화 서비스를 위한 WML 응용 프로그램의 제작이 전 세계적으로 활발히 이루어지고 있는 현재 상황에서, WAP Site를 브라우징하거나 Web 서버 내에 있는 WML 및 WMLScript[4] 문서의 오류를 검증하고 개발자가 그 오류를 다른 틀을 이용하여 수정하기 쉽게 하면서, PC상에서 이동통신 단말기와 동일한 현실감으로 무선 사이트에 접속할 수 있도록 하는 에뮬레이터가 절실하게 요구된다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 에뮬레이터 및 SDK(Software Development Kit)에 대해 알아보고, 3장에서는 에뮬레이터에 대한 설계 및 구현사항에 대해 기술한다. 4장에서는 결론 및 향후 과제에 대해서 기술한다.

2. 관련연구

국내의 WML 에뮬레이터의 개발 현황에 대하여 살펴본다.

2.1 네틀 WML 에뮬레이터[5]

1999년 8월 국내 최초로 WML 에뮬레이터를 개발하였으며, WML 콘텐츠 개발자가 작성한 문서 또는 외부에 연결된 WAP 페이지를 실제 단말기와 유사한 인터페이

스로 직접 테스트 해볼 수 있다. WML 문서와 WML Script문서를 지원하기 위해 Encoder 및 Decoder, Compiler 및 Interpreter가 모두 내장되어 있다. 그러나 WMLScript가 지원되지 않으며, WAP 게이트웨이와의 연동테스트가 불가능하다.

2.2 UP SDK[6,10]

- HDML, WML 서비스와 애플리케이션을 생성.
- WTLS를 포함한 WAP 1.1 지원.
- UP Simulator를 포함.
- UP Simulator는 UP Browser가 실행 가능한 폰에서의 작업을 가상적으로 실험이 가능.
- 로컬이나 UP.Link에 연결해 사용.

2.3 Nokia WAP Toolkit[7,10]

Nokia WAP Toolkit은 PC기반의 개발 환경으로 WAP 1.2를기반으로 하여 테스트하고, 디버그할 수 있다. 다음은 구성요소이다.

- WML, WMLscript Editor.
- Push서비스를 포함한 WAP 1.2 서비스 지원
- Device Simulator.
- GIF와 JPEG를 변환해 주고 그림을 편집할 수 있는 WBMP Tool.
- VCARD(business card)와 VCAL(calendar events)의 지원 .

2.4 Ericsson WAPIDE SDK 2.1[8,10]

- log window.
- WAP Browser, WML 1.1과 WMLScript 1.1 지원
- 로컬이나 Link에 연결후 사용.

2.5 Motorola ADK[9,10]

- VoxML 1.2와 VoiceXML 1.0을 모두 지원.
- VoxML/VoiceXML 디버깅 가능.
- Wbmp의 높이에 제한이 없으나 너비가 스크린 너비만큼의 제한이 있음.
- WML encoder feature 에러 메시지.
- 무조건 왼쪽으로 정렬됨.
- UTF-8 character set만을 지원.

3. WAP 에뮬레이터의 설계 및 구현

WAP 에뮬레이터(Emulator)의 전체적인 구조는 그림 1 과 같이 WML 파싱(parsing) 모듈과 인터페이스(Interface)모듈, 그리고 Data Send/Receive 모듈로 구성되며 각 모듈별 설계 및 구현사항은 다음과 같다.

3.1 파싱(Parsing) 모듈

WAP Emulator 설계 중 가장 중요한 부분으로 Data Send/Receive Module로부터 수신한 Decoding된 WML 문서를 XML DOM 파서를 이용하여 파싱하여 데이터 구조체에 저장한다.

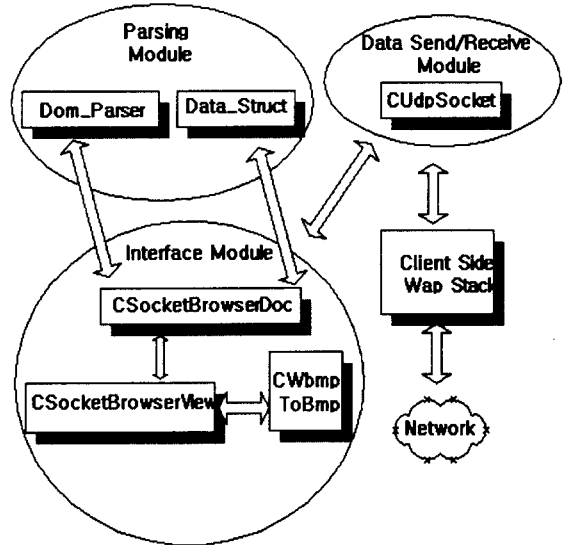


그림 1 WAP 에뮬레이터의 전체 구조

3.1.1 XML 파서(parser)

XML 파서는 모든 XML 문서의 처리의 기반이 되는 핵심 기술이며, XML 문서를 해석하고 필요한 정보를 추출하며 오류를 점검하는 기능을 제공한다. XML 문서를 XML 파서를 통하여 처리하여 현재 W3C에서 제시하고 있는 표준인 DOM(Document Object Model) 트리의 형태로 생성한다. DOM 트리는 XML 문서의 포매팅, 색인어 형성, 저장 변환 등에 있어서 필수적인 요소로서 XML 문서의 구조적 정보를 담고 있는 엘리먼트의 상호관계를 정의하게 된다. DOM 트리를 이용하여 XML DTD와 DI 및 EDI 문서등의 구성상에 오류를 점검하며 문서의 종류에 따른 계층모델을 형성하고 문서내용간의 관련성을 바탕으로 유사도를 추출할 수 있는 기반이 되도록 한다.[11]

3.1.2 데이터 구조체

그림 2에서와 같이 WML Content Server로부터 수신된 WML 문서는 Client 측 스택을 지나면서 Decoding 과정을 거친 후 XML Dom 파서를 이용하여 각 Card 별로 데이터 구조체에 저장되는 1차 파싱과정을 거친다. Card별로 저장된 WML 문서는 에뮬레이터로 실제 브라우징 되는 데이터 구조체와 각 태그, 태그 이름, 애트리뷰트를 저장하는 구조체로 XML Dom 파서 외부에서 다시 한번 2차 파싱과정을 거치게 되고, Interface 모듈에서 이 데이터 구조체를 통합하여 브라우징을 하게 된다.

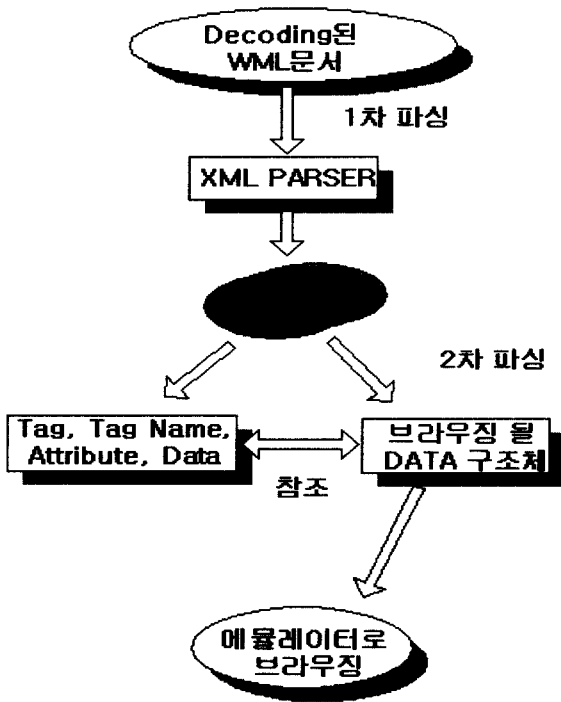


그림 2 WAP 에플레이터의 내부 동작도

3.2 Interface 모듈

서버에 WML컨텐츠를 요청하여, Data Send/Receive 모듈과 파싱 모듈을 생성한 후 Data 송/수신 모듈로부터 수신한 컨텐츠를 파싱 모듈로 전송하고 파싱 모듈에서 생성된 데이터를 브라우저한다.

3.3 Data Send/Receive 모듈

Interface 모듈에서 요청된 컨텐츠를 클라이언트 측 WAP 스택을 통해 수신(WML 문서 및 WBMP 파일)하고 수신된 컨텐츠를 다시 Interface 모듈로 전송한다.

3.4 구현환경

Window NT 4.0 상에서 C++로 구현된 클라이언트용 WAP 스택과 함께 에플레이터를 구현하였다. 구현된 WAP 스택에서 WTLS(Wireless Transport Layer Security Specification)부분은 제외되었다.

4. 결론 및 향후 관제

본 논문에서는 WML 컨텐츠 개발 과정에서 실제 휴대폰과 동일한 환경과 현실감을 제공해 개발자가 쉽게 WML 및 WMLScript 문서의 오류를 수정할 수 있는 WAP 에플레이터의 설계 및 구현에 대해 기술하였다. 향후에 아래와 같은 주제들에 관한 연구 및 개발이 이루어져야 한다.

- WAP 스택 중 WTLS 부분
- WMLScript 지원을 위한 가상머신 및 해석기

5. 참고 문헌

[1] Wireless Application Protocol Architecture Specification, WAP Forum, Apr 30. 1998, <http://www.wapforum.org/>

[2] Wireless Markup Language Specification Version 1.2, WAP Forum, February 19. 2000, <http://www.wapforum.org/>

[3] Handheld Device Markup Language Specification, <http://www.w3.org/TR/NOTE-Submission-HDML-spec.html>

[4] WMLScript Language Specification Version 1.2, WAP Forum, Jun. 2000, <http://www.wapforum.org/>

[5] 네플, http://www.netple.com/k_main/k.htm

[6] Phone.com, <http://www.phone.com/products/upsdk.html>

[7] Nokia, <http://www.nokia.com/corporate/wap/sdk.html>

[8] Ericsson, <http://www.ericsson.com/developerszone/>

[9] Motorola, http://mix.motorola.com/audiences/developers/developer_downloads.asp

[10] WAP design Ltd., <http://www.wapdesign.co.uk/tools.htm>

[11] W3C, "Extensible Markup Language", <http://www.w3.org/XML/>