

WML(Wireless Markup Language) 마이크로 브라우저의 설계 및 구현

장지산○, 최철웅, 신화중, 김상국, 신동규, 신동일
세종대학교 컴퓨터공학과
e-mail:jsjang@gce.sejong.ac.kr

A Design and Implementation of WML(Wireless Markup Language) Micro Browser

Ji-San Jang○, Chul-Oong Choi, Hwa-Jong Shin, Sang-Kuk
Kim, Dong-Kyoo Shin, Dong-Il Shin
Department of Computer Engineering, Sejong University

요약

WAP(Wireless Application Protocol)을 기반으로 한 무선 인터넷 분야의 발전과 더불어 엄청나게 늘어난 무선 인터넷 이용자들의 콘텐츠에 대한 요구가 다양해지고 있다. 따라서 CP(Content Provider)들은 무선 장비에 맞게 설계된 WML(Wireless Markup Language)과 WMLScript를 무선 장비를 사용하지 않고, 실제와 비슷한 상황에서 콘텐츠(Content)를 개발할 수 있는 환경이 필요하게 되었다. 본 논문에서는 무선 웹 콘텐츠를 좀 더 쉽게 개발할 수 있도록 도와주는 WML 마이크로 브라우저의 구현에 대해 기술한다. WML 마이크로 브라우저는 WML 파싱(parsing) 모듈과 브라우징(Browsing)모듈로 구성되고, 클라이언트측 WAP 스택으로 요청한 결과로 얻은 콘텐츠를 화면에 출력한다.

1. 서론

인터넷 기술의 발달은 네트워크에 연결된 컴퓨터만 있으면 사용자가 언제 어디서나 자신이 원하는 정보를 손쉽게 취득하는 것을 가능하게 하였다. 현재 개인용 컴퓨터들은 노트북을 제외하고 휴대하면서 사용하기는 거의 불가능하며, 휴대하면서 사용한다 하더라도 네트워크에 연결해서 인터넷을 이용하는 것은 쉬운 일이 아니다.

그렇기 때문에 최근 WAP(Wireless Application Protocol)[1]을 기반으로 하는 무선 인터넷이 급속도로 성장하고 있고 휴대용 무선 장비의 보급 또한 매우 빠르게 이루어지고 있다. 이러한 무선 인터넷의 발전과 더불어 웹 콘텐츠에 대한 사용자의 요구도 복잡하고 다양해지고 있다. 무선 인터넷 상의 웹 콘텐츠는 WML(Wireless Markup Language)[2]로 이루어져 있으며, WML은 휴대용 무선 장비에서의 효과적인 웹 접근을 위한 언어인 HDML(Handheld Device Markup Language)[3]을 기반으로 만들어졌다. WML은 HDML의 기본 구조인 데크(Deck)와 카

드(Card) 단위의 동작 스키마를 채택하고 있으며, 현재 WML 1.2 버전까지 발표되었다. WML과 WMLScript[4]는 무선 인터넷의 콘텐츠의 개발에 필수적이며, 그에 관련된 많은 응용 프로그램이 개발되고 있다. 휴대용 무선 장비가 아닌 개인용 컴퓨터 상에서, WML로 이뤄진 사이트를 방문하거나 실제와 비슷한 상황에서 무선 인터넷 관련 콘텐츠를 개발할 수 있는 WML 브라우저의 제작이 절실하게 요구된다.

이에 따라 상용화 서비스를 위한 무선 인터넷과 WML이나 WMLScript[4]로 이루어진 콘텐츠 개발을 손쉽게 할 수 있는 응용 프로그램의 개발이 다양하게 이루어지고 있다.

본 논문은 기존의 에뮬레이터 및 SDK에 대해 살펴보고, WML 마이크로 브라우저의 설계 및 구현 사항에 대해 기술하고, 마지막으로 결론과 향후 추가로 연구될 과제를 제시한다.

2. 관련연구

국내의 WML 애플레이터의 개발 현황에 대하여 살펴본다.

2.1 넷플 WML 애플레이터[5]

1999년 8월 국내 최초로 WML 애플레이터를 개발하였으며, WML 콘텐츠 개발자가 작성한 문서 또는 외부에 연결된 WAP 페이지를 실제 단말기와 유사한 인터페이스로 직접 테스트해 볼 수 있다. WML 문서와 WMLScript 문서를 지원하기 위해 Encoder 및 Decoder, Compiler 및 Interpreter가 모두 내장되어 있다. 그러나 WMLScript가 지원되지 않으며, WAP 게이트웨이와의 연동 테스트가 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 1.2를 기반으로 하여 테스트하고, 디버깅할 수 있다. 다음은 구성요소이다.

- WML, WMLScript Editor.
- Push 서비스를 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 [8,10]

- log window.
- WAP Browser, WML 1.1과 WMLScript 1.1 지원.
- 로컬이나 Link에 연결 후 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-9 character set만을 지원.

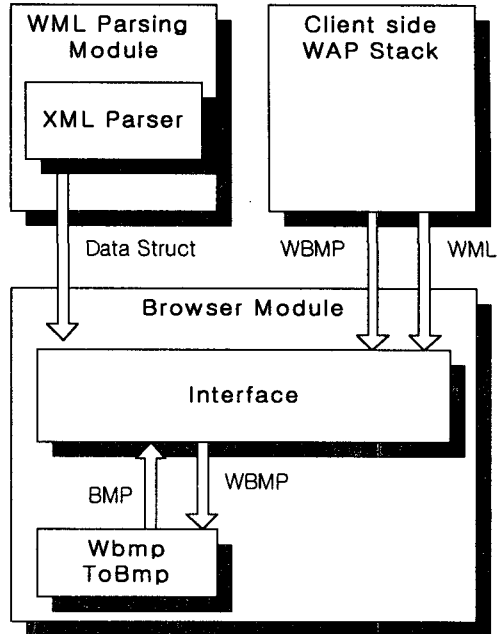


그림 1 WML 마이크로 브라우저의 전체 구조

3. WML 마이크로 브라우저의 설계 및 구현

WML 마이크로 브라우저의 전체적인 구조는 그림 1과 같이 WML 파싱(parsing) 모듈과 브라우징(Browsing) 모듈로 구성되며 각 모듈별 설계 및 구현 사항은 다음과 같다.

3.1 WML 파싱(Parsing) 모듈

WML 마이크로 브라우저 설계 중 가장 중요한 부분으로 클라이언트용 WAP 스택으로부터 수신한 Decoding된 WML 문서를 XML DOM 파서를 이용하여 파싱하여 데이터 구조체에 저장한다.

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 문서 등의 구성 상에 오류를 점검하며 문서의 종류에 따른 계층 모델을 형성하고 문서 내용간에 관련성을 바탕으로 유사도를 추출할 수 있는 기반이 되도록 한다.[11]

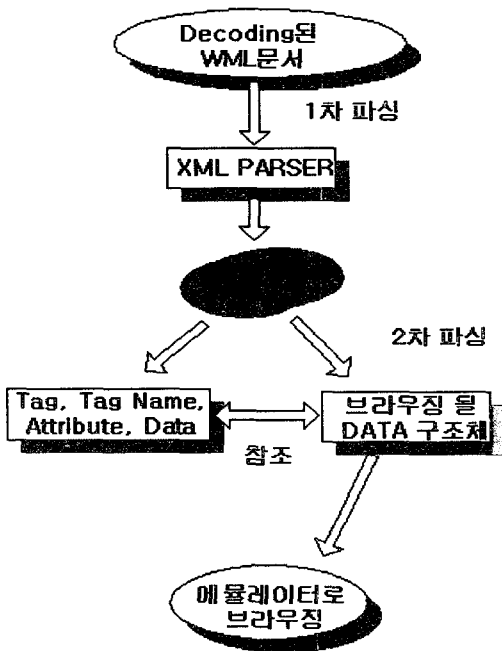


그림 2 WML 마이크로 브라우저의 내부 동작도

3.1.2 데이터 구조체

그림 2에서와 같이 WML Content Server로부터 수신된 WML 문서는 클라이언트 측 스택을 지나면서 Decoding과정을 거친 후 XML DOM 파서를 이용하여 각 Card 별로 데이터 구조체에 저장되는 1차 파싱 과정을 거친다. Card별로 저장된 WML 문서는 WML 마이크로 브라우저로 실제 브라우징되는 데이터 구조체와 각 태그, 태그 이름, 에트리뷰트를 저장하는 구조체로 XML DOM 파서 외부에서 다시 한번 2차 파싱과정을 거치게 되고, 브라우징 모듈에서 이 데이터 구조체를 통합하여 브라우징을 하게 된다.

3.2 브라우징 모듈

클라이언트측 WAP 스택에 콘텐츠를 요청해서 콘텐츠를 받는다. 이 때, 수신한 콘텐츠가 WML 문서일 경우와 WBMP 파일일 경우를 각각 나눠서 다른 모듈에 보내 처리하게 한다.

3.2.1 WML 문서 처리

클라이언트측 WAP 스택으로부터 수신한 콘텐츠가 WML 문서이면, 그것을 파싱 모듈로 전송한다. 브라우징 모듈은 파싱 모듈이 카드별로 파싱을 마친 결과물인 데이터 구조체를 이용해 화면에 출력한다. 만약에 화면에 출력하는 해당 카드 내에 WBMP 파일이 있다면 WBMP 파일을 다시 클라이언트측 WAP 스택으로 요청한다.

3.2.2 WBMP 파일 처리

WML 문서를 처리하는 도중에 브라우징 모듈이 WBMP 파일이 필요하면 해당 파일을 클라이언트측 WAP 스택에 요청해서 필요한 WBMP 파일을 수신한다. WBMP는 일반 PC 환경에서 출력할 수 있는 파일의 형식이 아니므로 브라우징 모듈 내의 WbmpToBmp 모듈로 보내 화면에 출력할 수 있는 형태로 바꾸어 브라우징 모듈 내의 Interface 모듈로 보내 화면에 출력하는 화면에 첨가하도록 한다.

만약 요청된 WBMP 파일이 지정된 곳에 존재하지 않을 때에는 WBMP를 뺀 카드를 출력한다.

3.3 구현 및 실행환경

Windows NT 4.0 상에서 C++로 구현되어 있는 클라이언트용 WAP 스택과 함께 WML 마이크로 브라우저를 구현하였다. 구현된 WAP 스택에서 WTLS(Wireless Transport Layer Security Specification) 부분은 제외되었다.

실행은 클라이언트용 WAP 스택을 실행시킨 상태에서 WML 마이크로 브라우저를 실행시켜 동작을 살펴보았다.

4. 결론 및 향후 과제

본 논문에서는 WML 콘텐츠 개발 과정에서 실제 휴대폰과 동일한 환경과 현실감을 제공해 개발자가 쉽게 WML 문서의 오류를 수정할 수 있는 WML 마이크로 브라우저의 설계 및 구현에 대해 기술하였다.

향후에 아래와 같은 주제들에 관한 연구 및 개발

이 이루어져야 한다.

- 클라이언트용 WAP 스택 중 구현되지 않은 WTLS 부분의 구현
- WMLScript 지원을 위한 가상머신 및 해석기
- WML, WMLScript 에디터(Editor)
- GIF나 JPEG 파일을 WBMP 파일로 변환시켜 주는 WBMP 컨버터(Converter)
- WBMP 파일을 편집하고 생성할 수 있는 WBMP Tool

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

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

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, June. 2000,
<http://www.wapforum.org/>

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

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

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

[8]Eircsson,
<http://www.ericsson.com/developerzone/>

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