

자연어를 기반으로 한 WYSIWYG WML 편집기의 설계 및 구현

한상진, 한판암
경남대학교 컴퓨터공학과

e-mail:greatbindae@hawk.com.kyungnam.ac.kr

Design and Implementation of a WYSIWYG WML Editor based on Natural Language.

Sang-Jin Han*, Pan-An Han*
Dept of Computer Engineering, Kyungnam University

요 약

IMT-2000의 상용화와 함께 무선 단말 장치와 PDA의 사용 용도가 단순히 상대방과의 통화 기능에 국한되지 않고 고속 데이터 통신이나, 멀티미디어의 대용량의 자료를 송수신하는 형태로 변화되고 있다. 이러한 변화에 따라 시공적인 제약없이 사용 가능한 무선 인터넷 연구가 WAP(Wireless Application Protocol) Forum을 중심으로 활발하게 진행되고 있다[1]. 무선 인터넷 환경을 구현하기 위해서 WML(Wireless Markup Language)을 사용하여 저작한다. 본 논문에서는 사용자가 보다 편하고 빠르게 WML을 작성할 수 있도록 하기 위해서 복잡한 명령어 구조나, 태그를 몰라도 자연어의 사용만으로 WML 문서를 쉽게 저작 할 수 있는 편집기를 구현한다. 이 편집기는 Nokia, 에릭슨에서 지원하는 에뮬레이터를 사용하지 않고 자체에 내장되어 있는 에뮬레이터를 통해서 직접 눈으로 확인하면서 작업할 수 있어 문서 저작의 생산성을 향상시킬 수 있는 자연어 사용을 기반으로 하는 WYSIWYG WML Editor이다.

1. 서론

데스크탑 컴퓨터를 이용한 웹 검색은 이동성이 떨어지는 단점을 가지고 있다. 이에 따라 이동성이 강한 소형 휴대용 장비에 대한 관심이 높아지고 있다. 이 같은 이동성이 강한 단말기로는 PDA와 휴대폰을 들 수 있다. 이들은 시공간적인 제약없이 웹에 존재하는 많은 양의 정보를 습득 가능하기 때문에 관심이 고조되고 있다. 이와 같은 관점에서 본 논문에서는 WAP를 지원하는 휴대폰을 이용해서 무선 인터넷 서비스를 제공하는데 사용되는 WML 문서를 작성함에 있어서 WML 기본 구조와 tag 등의 에비지식이 없이 쉽게 구현 가능한 편집기를 구현한다. 현재 나와 있는 몇몇의 저작도구들은 하나같이 문법과 tag를 알아야 사용할 수 있다. 그러나 본 연구에서는 HTML 저작도구인 나모, 드림위버와 같은 자연어를 사용할 수 있는 편집기를 구현한다. 이는

직접 제작한 WML 문서를 WYSIWYG 방식으로 편집, 수정, 미리보기가 가능한 저작도구이다.

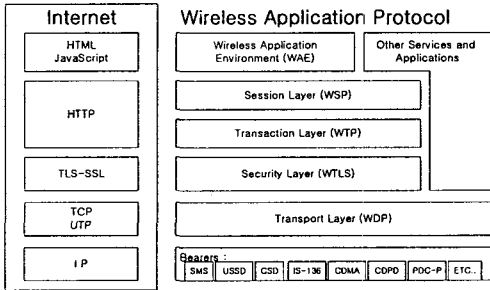
2. 관련 연구

2.1 WAP의 구조

WAP의 구조는 인터넷과 매우 유사하다. 서버에 저장된 애플리케이션에 접속하기 위해서 클라이언트는 WAP 게이트웨이와 연결을 설정하고, 콘텐츠를 요구하게 된다. 게이트웨이는 WAP 클라이언트로 부터 이 요청을 인터넷에서 쓰이는 형식으로 변환하고 WAP 서버에 전달한다. 서버는 이 요청에 대한 응답을 보내고, 게이트웨이는 WAP 형식으로 변환하여 WAP 클라이언트로 보낸다[6].

2.2 WAP 프로토콜 스택

WAP는 5개의 서로 다른 계층 구조로 이루어져 있으며, 각 계층은 기능적으로 나누어져 특정한 서비스를 제공한다[2][7]. 그림 1은 WAP 프로토콜의 스택과 인터넷 프로토콜의 비교를 나타내고 있다 [4][6].



[그림 1] WAP 프로토콜 스택

2.2.1 WAE

무선 애플리케이션 환경(Wireless Application Environment : WAE)은 개발자들에게 현존하는 콘텐츠에 최적화되어 있다. 그리고 제한된 용량의 기기와 통신 가능하게 특정 포맷과 서비스를 사용할 수 있도록 해준다.

2.2.2 WSP

무선 세션 프로토콜(Wireless Session Layer : WSP)은 정규화 된 형태로 애플리케이션 사이에 데이터 교환이 가능한 두 가지 프로토콜을 가지고 있다.

2.2.3 WTP

무선 처리 프로토콜 서비스(Wireless Transaction Layer : WTP)는 신뢰성과 비-신뢰성 처리를 실행하며, WDP 레이어나 최적 보안 레이어 WTLS에서 동작한다.

2.2.4 WTLS

무선 전송 보안 레이어(Wireless Transport Layer Security : WTLS)는 TLS(Transport Layer Security) v1.0에 기반한 최적 레이어이다. 이 TLS는 SSL(Secure Socker Layer) v3.0에 기반을 두고 있는 인터넷 프로토콜의 하나이다.

2.2.5 WDP

무선 데이터그램 프로토콜(Wireless Datagram Protocol : WDP)는 WAP 스택의 최하위 레이어이다. 이는 이식성이 좋은 프로토콜이기 때문에 다른 이동 네트워크 사이에서도 작동 가능하다[4][6].

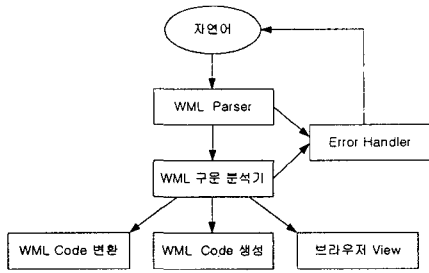
3. 설계 및 구현

3.1 모듈의 설계시 고려사항

WML문은 기본적으로 이동 단말기 상에서 사용하기 적합하도록 구현된 언어이다. 이는 이동 단말기 한 화면을 나타내는 카드(Card)와 이런 카드들의 모임인 데크(Deck)로 구성 되어있다. 또한 HTML과 같이 각각 태그(Tag)들로 구성되어 있다. 하지만 초보자들이 카드와 데크, 태그들을 사용하여 저작하기는 부담스럽다. 현재 나와 있는 몇몇의 저작 도구들은 각각의 에디터를 가지고 있고, 그에 해당하는 브라우저를 가지고 있지만, 초보자들이 사용하기는 쉽지않다. 또한 자신이 저작한 WML문을 다른 툴을 사용하여 결과를 확인해야 하는 번거로움이 있다. WYSIWYG 기능을 지원하는 에디터(예, 나모 웹에디터, 드림위버)를 사용할 때 저작의 용이성과 문서를 저작함에 있어 생산성이 월등히 증가하게 된다. WYSIWYG가 지원되는 에디터를 사용할 때는 사용자가 태그에 대해서 모르더라도 쉽게 WML문서를 만들 수 있다. 그리고 시각적인 저작 환경을 제공할 수 있으므로 사용하기 편리하다.

위의 설계 원칙에 따라 구현하기 위해서 WML 편집기 툴은 그림 2와 같이 6개의 모듈로 구성된다.

- ▶ 입력 받은 내용을 가시적으로 보여주는 입력 모듈
- ▶ 각각의 입력 내용의 유효성을 검사하고 파싱하는 파싱모듈
- ▶ 파싱모듈에서 받은 내용으로 WML 코드로 변환하는 WML 코드 변환 모듈
- ▶ 각각의 모듈에서 에러를 감지하고 수정하게 될 에러 핸들러(Error Handler) 모듈
- ▶ WML 코드 생성을 위한 WML 코드생성 모듈
- ▶ 작업한 WML 문서를 보여주기 위한 브라우저 view 모듈



[그림 2] 모듈 구성도

3.2 입력-파싱 모듈

입력-파싱 모듈은 사용자가 저작하려고 하는 내용을 워드프로세서 사용하는 것처럼 입력창에서 입력하면 표 1 WML 태그[3]의 파싱 규칙에 의해서 문서를 파싱한다. 사용자가 필요에 따라 입력한 내용을 Head, 스타일, 선택, 링크, Card Action, 사용자 입력, 이미지 입력, 표, paragraph으로 구분하여 코드 변환 모듈에서 사용할 데이터를 생성하게 된다.

[표 1] WML 태그의 파싱 규칙

메뉴	요소
Head	<template>, <head>, <access>, <meta>
스타일	, <i>, <u>, , , <big>, small>, <u>,
선택	<select>, <option>, <optgroup>, <fieldset>
링크	<a>, <anchor>
Card Action	<do>, <go>, <prev>, <noop>, <onevent>, <postfield>, <fefresh>, <timer>
사용자 입력	<input>
이미지 입력	
표	<table>, <td>, <tr>
paragraph	<p>

파싱 과정에서 발생한 에러를 핸들링하기 위한 에러처리 모듈을 사용하여, 입력받은 내용을 WML 문서의 구조 정보를 가지고 있는 WML1.2 DTD를 근거로하여 문서의 논리적 구조를 검사한다.

3.3 WML 코드 변환 모듈

WML 코드 변환 모듈은 사용자가 입력한 자연

어를 분석하여 WML 문법에 맞는 코드로 만들어 주는 모듈이다. 이는 입력 받은 내용을 WML 코드로 자동 변경해 주고, 사용자가 수정을 했을 때 자동으로 WML 코드를 수정하는 모듈이다.

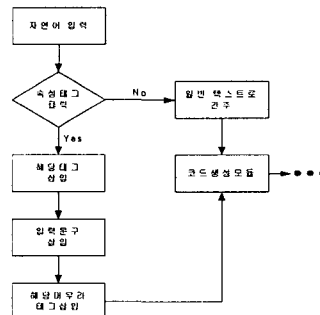
사용자가 입력한 자연어를 WML코드로 변환시키기 위해서 아래와 같은 원칙을 세워 작업이 편리하도록 했다.

- ▶ 기본적인 WML 문의 구조는 미리 정의해 생성시킨다.
- ▶ 사용자가 엔터를 치는 동시에 코드가 생성되도록 한다.
- ▶ 사용자가 수정을 하면 해당 소스코드를 모두 지우고 수정한 부분을 다시 코드 생성한다.

이 모듈에서 생성된 코드는 파서 모듈을 통해서 파싱되어, 결과를 디스 플레이 모듈로 보내 사용자 인터페이스를 통해서 갱신된 내용을 확인할 수 있다.

3.4 WML 코드 생성 모듈

그림 3과 같은 WML 코드 생성 모듈은 WML 코드 변환 모듈에서 생성된 원시 코드를 이용하여 문서 내에서 사용한 태그들의 구문규칙을 나타낸 WML1.2 DTD(Document Type Definition)에 의해 문서의 유효성을 검사하고, WML 코드를 생성과 저장 기능을 가진 모듈로서, 입력받은 내용으로부터 WML 문서의 코드를 생성한다.



[그림 3] 코드 생성 모듈

3.5 WML 브라우저

외부에서 불러온 WML 문서 혹은 사용자가 직접 저작한 결과물을 확인해 주는 부분으로서, 사용

자가 저작한 내용을 다른 툴을 사용하지 않고 저작한 내용을 확인할 수 있다. 본 논문에서 저작한 브라우저는 크게 세가지 부분으로 구성된다. 첫째로 이동 단말기의 디스플레이 부분에 해당하는 View 부분, 둘째로 사용자의 의사를 전달하고 화면을 스크롤 시키는 기능의 Control 부분, 셋째로 사용자가 직접 원하는 값이나 문자를 입력할 수 있는 입력부분으로 나눈다.

4. 기존기법과 연구결과 검토

기존에 텍스트 에디터와 본 연구에서 제시한 WYSIWYG WML 에디터를 사용하여 같은 분량을 저작할 때 각각 어느 정도 저작 능력을 보이는지 테스트하여 성능을 비교한다. 비교의 기준은 저작이 완료하여 view 화면에 나타내는데 걸리는 총 시간을 합하여 분석한다. 기존에 사용하던 툴은 windows 기반의 운영체제에 있는 메모장과, IDM Computer Solutions, Inc.사의 UltraEdit를 사용한다. 그런데 이 두 툴은 WML 코드를 보여 줄 수 있는 view가 별도로 없기 때문에 Nokia사의 에뮬레이터를 사용하여 작성한 소스 코드를 view하도록 한다.

[표 2] 각 소스 라인별 시간 측정표 (단위 : 초)

모델 라인수	A (메모장)	B (UltraEdit)	C (WYSIWYG WML)
50	473	482	365
100	870	890	611
150	1529	1541	1033
200	2025	2000	1340

표 2에 나타낸 바와 같이 A에디터의 실험결과 50라인에서 473초, 200라인에서 2025초가 소요되었다. 그리고 B에디터의 실험 결과 50라인에서 482초, 200라인에서 2000초가 소요되었다. 그런데 A, B는 일반 텍스트 에디터이기 때문에 실험 결과와 같이 시간의 차이는 조금씩 있다 하여도 문서를 저작하는 생산성에는 크게 영향을 미치지 못한다. 그러나 C에디터의 결과 50라인에서 365초, 200라인에서 1340초가 소요되었다. A, B에디터에 비해 많은 시간을 단축시킬 수 있음을 의미한다. 이는 적은 양의 문서를 저작할 때는 일반 에디터와 차이를 보이지 않지만, 많은 양의 문서를 저작할 때 시간이 단축되고 생산성을 향상시킨다.

5. 결론

실험 결과를 살펴보면 각각의 소요 시간만을 체크하여 작업에 효율과 생산성을 높일 목적으로 사용하기 위해서는 WML 전용 에디터를 사용하면 훨씬 더 편리함을 알 수 있다. 또한 전용 에디터를 사용하면 사용자의 편의성과 생산성의 증대를 가져오는 효과를 얻을 수 있다. 그리고 사용자는 WML에 대한 기초지식이 없더라도 툴의 사용법을 익히면 누구나 쉽게 이동 단말기용 콘텐츠를 개발 할 수 있다.

참고문헌

- [1] Michael Leventhal, David Lewis, Matthew Fuchs, *Designing XML Internet Application*, Prentice Hall, 1998.
- [2] WAP Forum, "WAP Wireless Markup Language Specification Version 1.1" SPEC-WAPArch, June 1999.
- [3] Wireless Markup Language, WAP Forum, November 04, 119. URL : <http://www.Wapforum.org>
- [4] Charles Arehart, "PROFESSIONAL WAP", 2000
- [5] Wireless Application Language Public Key Infrastructure, WAP Forum, March 03 2000. URL : www.wapforum.org
- [6] WAP Forum, URL : <http://wapforum.org>
- [7] 강도우, 박기현, "WAP 게이트웨이에 대한 연구", *정보처리학회 추계학술논문집* 제7권 제2호 October 2000.