WIPI 기반의 모바일 단말을 위한 내장형 XML 파서 및 뷰어

강 미연*, 김도완*, 정원호°
* (주)ICANTEK 기술연구소
o 덕성여자대학교 컴퓨터과학부
mykang@icantek.com, whchung@center.duksung.ac.kr

An Embedded XML Parser and Viewer for WIPI-Based Mobile Devices

Mi-Yeon Kang* , Do-Wan Kim* and Won-Ho Chung°

* Research Center, ICANTEK Co.
o School of Computer Science, Duksung Women's University

요 약

유무선 인터넷의 발달로, PDA, 셀룰러폰, Hand-held PC 등 low-end로부터 high-end까지의 다양한 규모의 하드웨어 자원을 가지는 유무선 단말들이 속속 등장하고 있다. 웹 상의 정보 표현을 위한 표준으로 자리잡은 확장성 표기 언어인 XML을 위한 파서 및 뷰어는 그러한 단말에서의 자료 브라우징을 위해 필수적으로 임베딩 되어야할 소프트웨어 중의 하나이다. 본 논문에서는 국내의 모바일 표준 플랫폼 규격으로 채택된 WIPI 기반의 모바일 단말에 내장되어 수행되는 소규모 모듈 기반의 XML 파서가 설계, 구현되며, 그 응용으로 간단한 전자책 뷰어가 구현된다. 본 XML 파서는 경량급으로 셀룰러 폰과 같은 소규모 단말로의 내장을 위해 유용하다.

1. 서론

특정 하드웨어에 가장 적절하게 고정되는 경직성(fixedness)과 경량화는 내장형 소프트웨어가 가져야할 중요한 특성들이라 할 수 있다[1]. 이는 고속의 실시간 처리를 위해 취할 수 있는 설계 방법이라고 볼 수 있다. 과거에는 그러한 실시간 장치들이 극히 제한된 분야에서만 사용되었으나, 최근에는 인터넷 데이터 접근 및 처리를 위한 low-end PDA로부터 high-end hand-held PC까지 매우 다양한 인터넷 접근용 모바일 단말이 속속 개발되어 상품화되고 있다. 국내 연구소와 한국무선 인터넷 표준화 포럼(KWISF)에 의해 공동 개발되어, 최근 국내의 각 통신 사업자에 채택되어 셀룰러 폰의 응용 플랫폼으로 자리잡기 시작한 WIPI(Wireless Internet Platform for Interoperability)는 국내 모바일 단말의 표준 플랫폼 규격으로 채택되었으며, J2ME 혹은 BREW와는 달리 Java와 C 언어 모두를 수용할 수 있다는 특징을 가지고 있다[2-4]. 또한, 확장성 표기언어인 XML이 웹 상의 정보 표현의 표준으로 결정됨에 따라, 인터넷 상의 콘텐츠 표현의 주요 표기 언어인 XML 처리를 위한 내장형 파서 엔진, XML 뷰어 등도 이제, 각종 모바일 단말에 내장되어야 할 필수적인 소프트웨어 중의 하나가 되었다. 이러한 XML 처리기는 파서 엔진을 가장 기본적인 컴포넌트로 가지고 있으며, C 혹은 Java 언어를 기반으로 개발되어 발표되었다. 이러한 XML 파서 엔진을 축으로 하는 XML 처리기도 이제는 내장형 소프트웨어로서의 그 기능을 수행함에 있어서, 경량화, 고속화 등을 요구받고 있는 것이다[5].

본 논문에서는 국내 표준 모바일 플랫폼 규격인 WIPI 기반의 모바일 단말로의 내장을 위한 경량급 XML 파서와 뷰어가 설계 구현된다. 구현된 XML 파서는 3개의 모듈로 구성되며, 그 크기는 13Kbytes 내외이며, 하나의 모듈을 생략하면 10Kbytes까지 줄어든다. 또한, 개발된 XML 파서의 응용 분야로 인터넷을 통해 XML 데이터를 단말에 뷰잉할 수 있는 간단한 XML 뷰어가 구현되었으며, 간단한 전자책을 WIPI 에뮬레이터를 사용하여 시현하였다.

본 연구는 2003년도 산업자원부 중기거점과제 "디지털 가전형 POST-PC 플랫폼 기술개발사업" 지원으로 이루어졌음.

2. 관련 연구

2.1 WIPI(Wireless Internet Platform for Interoperability)

국내 이동통신사의 무선인터넷 플랫폼으로 KVM, SK-VM, GVM, MAP, BREW, WITOP 등이 사용되고 있다. SKT의 WITOP을 제외한 대부분의 플랫폼의 개발언어는 C 또는 자바 중 한 가지 언어만을 지원하고 있으며, 개발언어가 같더라도 플랫폼이 다르면 컨텐츠간의 상호 운영성이 없다. 이와 같은 문제의 해결을 위하여 모바일 플랫폼 표준화 연구가 진행되었고, 그 결과 2002년 말경 국내의 모바일 표준 플랫폼 규격으로 WIPI가 등장하였다. WIPI는 무선 단말기에 탑재되어 응용프로그램을 수행할 수 있는 실행환경을 제공하는 모발일 표준 플랫폼이며, J2ME, BREW 등과는 달리 Java와 C 언어 모두를 수용할 수 있다는 장점을 가지고 있다. WIPI는 플랫폼 이식성을 높이기 위한 표준화된 하드웨어 추상화 계층인 HAL(Handset Adaptation Layer)을 적용하여 플랫폼 호환성을 제공하며, 다양한 응용 프로그램 개발을 촉진하기 위해 기본 응용 프로그래밍 인터페이스(Basic Application Programming Interface)를 제공한다[4].

2.2 XML(Extensible Markup Language) 파서

일반적으로, XML 문서는 element, attribute, entitiy, DTD(Document Type Definition) 등 4개의 주요 구성 요소가 있다. element는 태그 데이터를 표현하는 것이고, attribute는

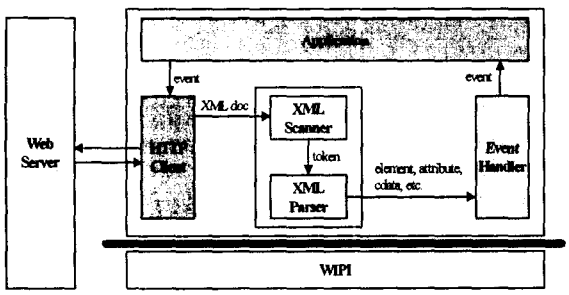
element에 대한 정보를 더하기 위해 사용된다. entity는 XML 문서에서 참조할 수 있는 데이터의 실질적 값이다. DTD는 개개의 XML 문서의 구조를 정의하는 XML의 선택 부분이다.

XML 파싱 작업은 문서로부터 태그의 의미와 구조에 기초하여 이러한 구성 요소를 추출해 내는 것을 의미한다.

XML 파서는 tree-based 파서와 event-based 파서로 분류한다. tree-based 파서는 메모리에 트리 구조로 XML 문서 전체를 읽으며, 트리의 각 노드는 문서의 데이터를 나타낸다. event-based 파서는 callback 메소드를 통해 파싱된 XML 구성 요소를 이벤트로 전달한다. 일반적으로 event-based 파서가 tree-based 파서보다 CPU, 메모리 등을 적게 소모하고, 수행 속도가 빠르다. 그러나 event-based 파서는 XML 데이터에 순차적인 접근만 가능하며, 문서의 수정, 추가가 어렵다. 반면 tree-based 파서는 데이터의 접근이 쉬우며, XML 문서를 조작하기가 용이하다.

3. XML 파서의 구성 및 동작

본 논문에서 구현한 XML 파서는 이벤트 기반의 파서이며, 이벤트를 처리하는 부분을 interface class에 callback 메소드로 정의 하였다. 응용 프로그램은 파서로부터의 파싱 이벤트를 받기 위해 이 callback 메소드를 구현하여야 한다. 구현된 XML 파서는 원격 접속을 위한 HTTP 클라이어트로 동작할 수 있다는 특징을 가지고 있다.



[그림-1] XML 파서의 구조

본 논문에서 개발된 XML 파서는 [그림-1]에서 보아 알 수 있듯이 HTTPClient, WIPIXMLParser, EventHandler 등 3가지 모듈로 구성되며, 각각은 독립적인 thread로 수행된다.

HTTPClient는 HTTP 프로토콜을 통해 웹 서버에 접속하여, 서버로부터 XML 문서를 WIPI 단말기로 전송받을 수 있으며, 전송받은 데이터는 메모리에 저장된다. 또한, 저장된 XML 문서를 파싱하는 과정에서 외부 파일이 참조되는 경우, 해당 파일을 다시 해당 서버에게 요청하여 단말기로 전송받아 작업을 처리한다.

WIPIXMLParser는 XMLScanner와 XMLParser로 구성되며, 문서의 실제 파싱을 담당한다. XMLScanner는 HTTPClient에 의해 전송되어 저장된 XML 문서를 토큰 단위로 분리하여 XMLParser에게 전달하는 역할을 하는데, 이때, XMLParser는 XML 문서 구조의 적합성 여부를 확인하고, 토큰을 XML의 구성 요소인 element, attribute, character data 등의 이벤트로 분류하여 EventHandler에게 전달한다.

[표-1] XML 파서의 EventHandler 모듈에 정의된 메소드

메소드	설명
startElement	element의 시작 태그 이벤트 처리
emptyElement	empty element 이벤트 처리
endElement	element의 종료 태그 이벤트 처리
attribute	attribute 이벤트 처리
charaterData	text data 이벤트 처리

EventHandler는 WIPIXMLParser로부터 전달받은 이벤트를 처리하는 callback 메소드를 정의한 interface class이며, XML 파서를 이용하는 응용 프로그램에 따라 적합하게 EventHandler의 메소드를 구현하여야 한다. 파서를 구성하고 있는 EventHandler에 정의된 메소드는 [표-1]과 같다. 즉, 응용 프로그램에서는 각 메소드에 해당하는 기능을 응용에 맞춰 구현하여야 하며, 이는 XML 파서의 한 표준인 SAX와 유사하다고 할 수 있다. 본 논문에서는 간단한 전자책이 이들을 기반으로 구현되었다.

4. XML 파서 비교 분석

본 논문에서 개발된 XM 파서를 구성하고 있는 각 수행 모듈의 크기는 [표-2]와 같다. 대략적인 크기가 13Kbytes를 차지하고 있지만, HTTPClient 모듈을 생략한다면 약 10Kbytes 크기로 줄일 수 있다.

[표-2] XM 파서를 구성하고 있는 각 수행 모듈의 크기

구분	사이즈 (JAR파일)
HTTPClient 모듈	2KB
WIPIXMLParser 모듈	10KB
EventHandler 모듈	1KB

WIPI 기반의 파서는 아니지만, 현재 무선 응용을 위한 자바 기반의 대표적인 파서로는 TinyXML[6], NanoXML[7], AElfred[8] 파서 등이 있으며, 이들의 특징들이 [표-3]에 요약되어 있다. 이들은 모두 J2ME 기반으로, 기본적으로 임베디드 자바 응용 또는 자바 애플릿에 사용되기 위해 디자인 되어졌다.

TinyXML은 event-based interface를 제공하는 10KB의 경량급 파서로, 본 논문에서 개발된 파서와 가장 유사하다는 것이 특징이다. NanoXML은 2가지 버전을 제공하고 있는데, simple-tree-based interface를 제공하는 초경량급 경우와 SAX를 지원하는 경우이다. SAX를 지원하는 경우 그 크기는 2배 이상으로 커짐을 알 수 있다. AElfred 파서는 다른 두 파서보다 크기가 크다. event-based 파서는 약 18KB이며, SAX Java API를 포함하는 경우는 30KB 크기를 가지고 있음을 알 수 있다.

이들 파서들은 모두 non-validating 파서이다. TinyXML,

AElfred 파서는 Entity를 지원하면서, event-based interface를 제공한다. SAX 버전 NanoXML과 AElfred 파서는 SAX를 지원하면서, 특히, NanoXML 파서는 simple-tree-based interface를 제공하는 것이 특징이다. 본 연구에서 개발된 파서는 기본적으로 TinyXML, AElfred 파서와 기능적으로 동일하며, HTTP 클라이언트 모듈을 포함하고 있는 것이 특징이라 할 수 있으며, 이 모듈은 필요에 따라 추가 혹은 제거할 수 있다.

[표-3] 무선응용을 위한 자바기반의 파서

구분	사이즈 (JAR파일)	기능
TinyXML event-based interface	10KB	• non-validating 파서 • Entity 지원 • event-based interface 제공
NanoXML tree-based interface	9KB	• non-validating 파서 • tree-based interface 제공 • SAX 지원
NanoXML SAX interface	21KB	
AElfred event-based interface	18KB	• non-validating 파서 • Entity 지원 • event-based interface 제공 • SAX 지원
AElfred SAX interface	30KB	
XML 파서 event-based interface	13KB	• non-validating 파서 • Entity 지원 • event-based interface 제공 • HTTP 지원

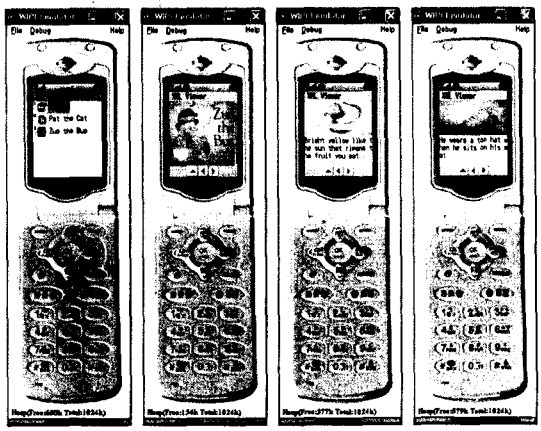
5. 응용

본 연구에서 개발된 XML 파서를 기반으로 하는 응용으로 전자책을 볼 수 있는 간단한 book-Viewer를 개발하였다. Book-Viewer에서 처리하고 있는 element와 attribute는 [표-4]에 보여준 바와 같다.

[표-4] Book-Viewer가 가지는 엘리먼트 및 attribute

구분	이름	의미
element	start	XML 문서의 시작
	title	XML 문서의 제목
	image	display되는 이미지 파일 src attribute를 포함
	sound	재생되는 사운드 파일 src, loop attribute를 포함
	description	display되는 내용
	prev	이전 페이지, href attribute를 포함
	next	다음 페이지, href attribute를 포함
	home	처음 페이지, href attribute를 포함
attribute	src	파일의 위치 image, sound element에서 사용
	loop	재생 횟수 sound element에서 사용
	href	이동할 페이지 prev, next, home element에서 사용

Book-Viewer는 XML 파서에서 제공하고 있는 EventHandler의 메소드들을 위에 정의된 element와 attribute를 처리하도록 구현한 것이다. [그림-2]는 아로마 WIPI 에뮬레이터[9] 상에서 웹서버와 접속하여 Book-Viewer에서 동화책을 보여주는 예제이다. XML 문서와 이미지 파일 등은 http://www.younmeus.net 서버로 접속하여 전송받았다.



[그림-2] Book-Viewer 응용 예

6. 결론 및 향후과제

본 논문에서 설계, 자원의 규모가 적은 모바일 단말에 효율적으로 내장 가능한 XML 파서가 설계 구현 되었으며, 그 응용으로 10 여개의 엘리먼트와 attribute를 가지는 전자책 뷰어인 Book-Viewer가 구현되었다. 개발된 파서는 최하 10Kbytes 크기를 가지는 경량급 내장형 파서이며, 모바일 단말을 위한 XML 파서로 유용하다.

참 고 문 헌

[1] E. A Lee, "What's Ahead for Embedded Software?," IEEE Computer, Vol. 33, No. 9, Sept. 2000
[2] 박수원 외, "위피 모바일 프로그래밍,", 한빛미디어, 2003
[3] 홍준성, "모바일 플랫폼 기술현황 및 발전방향," 한국정보과학회지, 22권 1호, 2004년 1월
[4] 이상윤 외, "한국 무선 인터넷 표준 플랫폼(WIPI)의 표준화 현황 및 발전 전망," 한국정보과학회지, 22권 1호, 2004년 1월
[5] 강미연 외, "구성적 임베딩을 위한 모듈 기반의 XML 처리기의 설계," 2002 한국정보과학회 추계학술대회 논문집
[6] TinyXML, "http://www.grinninglizard.com/tinyxml"
[7] NanoXML, "http://nanoxml.cyberelf.be"
[8] AElfred XML, "http://saxon.sourceforge.net/aelfred.html"
[9] MobileJAVA, http://www.mobilejava.co.kr