

## WIPI 모바일 단말을 위한 Pull 모드 XML 처리기

정순지<sup>o</sup>, 유호연<sup>o</sup>, 김지현<sup>o</sup>, 강미연<sup>\*</sup>, 정원호<sup>\*</sup>

<sup>o</sup> 덕성여자대학교 컴퓨터공학부

<sup>\*</sup> (주)iCanTek 기술연구소

mykang@icantek.com, whchung@center.duksung.ac.kr

### A Pull-Mode XML Processor for WIPI-Based Mobile Devices

Soon-Ji Jung<sup>o</sup>, Ho-Yeon Eu<sup>o</sup>, Ji-Hyun Kim<sup>o</sup>, Mi-Yeon Kang<sup>\*</sup> and Won-Ho Chung<sup>o</sup>

<sup>o</sup> Division of Computer Engineering, Duksung Women's University

<sup>\*</sup> Research Center, iCanTek Co.

#### 요 약

기존의 다양한 무선 인터넷 플랫폼이 WIPI라는 무선 인터넷 플랫폼으로 통합되면서 플랫폼 간의 호환성과 이식성을 제공하여 다양하고 풍부한 콘텐츠들이 개발되고 있다. 인터넷 상의 콘텐츠 표현의 주요 표기 언어인 XML을 파싱하고 뷰잉 하기 위한 XML 처리기 역시 이제, 각종 모바일 단말에 내장되어야 할 필수적인 소프트웨어 중의 하나가 되었다. 본 논문에서는 WIPI 기반의 모바일 단말로의 내장을 위한 효율적 XML 처리기가 설계 구현된다. XML 처리기는 HTTP를 기반으로 하여 서버와 XML 데이터를 교환할 수 있으며, 모바일 단말에 적합하도록 CPU 부하를 덜어주고, 메모리 사용량을 줄이면서, 크기가 작은 Pull 모드 XML 파서를 핵심 컴포넌트로 가지고 있다. 또한, XML 처리기를 사용하여 인터넷을 통해 XML 문서를 전송 받아, 모바일 단말에서 볼 수 있는 간단한 전자책을 개발하였다.

#### 1. 서론

무선 네트워크의 발달과 모바일 기기의 대중화에 따라 최근의 컴퓨팅 환경은 PDA, 스마트폰, 셀룰러폰 등 모바일 환경으로 급변하고 있으며, 인터넷 데이터 접근 및 처리를 위한 다양한 인터넷 접근용 모바일 단말이 속속 개발되어 상품화되고 있다. 뿐만 아니라, 극히 제한된 분야에서 사용되었던 실시간 장치들이 모바일 단말에서도 사용되면서, 고속의 실시간 처리를 위해 내장형 소프트웨어가 가져야 할 중요한 특성들인 특정 하드웨어에 가장 적절하게 고정되는 경직성과 경량화에 대한 요구가 모바일 단말에 까지 증가하고 있다[1]. 뿐만 아니라, 기존의 다양한 무선 인터넷 플랫폼이 WIPI(Wireless Internet Platform for Interoperability)라는 무선 인터넷 플랫폼으로 통합되면서 플랫폼 간의 호환성과 이식성을 제공하여 다양하고 풍부한 콘텐츠들이 개발되고 있다 [2][3]. 또한, 확장성 표기언어인 XML(eXtensible Markup Language)이 인터넷 상의 정보 표현의 표준으로 결정됨에 따라 [4], 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. XML 처리기도 이제는 내장형 소프트웨어로서의 그 기능을 수행함에 있어서, 경량화, 고속화 등을 요구받고 있으며, 저대역폭, 적은 메모리, 낮은 CPU 속도를 가지는 모바일 단말에까지 사용이 확대되어 가고 있다[5].

본 논문에서는 WIPI 기반의 모바일 단말로의 내장을 위한 효율적 XML 처리기가 설계 구현된다. XML 처리기는 HTTP를 기반으로 하여 서버와 XML 데이터를 교환할 수 있으며, 모바일 단말에

적합하도록 CPU 부하를 덜어주고, 메모리 사용량을 줄이면서, 크기가 작은 Pull 모드 XML 파서를 핵심 컴포넌트로 가지고 있다. 또, XML 처리기를 사용하여 인터넷을 통해 XML 문서를 전송 받아, WIPI 단말에서 볼 수 있는 간단한 전자책을 개발하였다.

#### 2. 관련 연구

##### 2.1 WIPI

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 콘텐츠간의 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 모바일 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[6].

##### 2.2 XML 파서

일반적으로, XML 문서는 element, attribute, entity,

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. 이러한 XML 파서로 SAX(Simple API for XML)와 DOM(Document Object Model) 기반 파서가 사용되고 있다. SAX는 이벤트 기반의 파서이며, 특정 XML 구성 요소(element, attribute 등)가 나타날 때마다 응용 프로그램이 callback 메소드를 통해 파싱된 XML 구성 요소를 이벤트로 전달받는 Push 방식의 파서이다. DOM은 메모리에 트리 구조를 가지며, 전체 문서를 스캔하여 마주치는 각각의 구성 요소의 객체 트리를 구축한다. 이 경우, XML 문서의 스캔이 완료된 후 생성되는 트리에 접근하여 수정할 수 있다. 트리 기반의 DOM 파서는 스캔 되는 문서의 양에 따라 과도한 양의 메모리를 사용하는 단점이 있다. 일반적으로 이벤트 기반의 SAX 파서가 트리 기반의 파서보다 CPU, 메모리 등을 적게 소모하고, 수행속도가 빠르지만, 이벤트 기반의 Push 모드 파서는 항상 XML 문서의 처음부터 끝까지 순차적인 접근만 가능하며, 또한 파서에 의해 구성 요소가 나타날 때마다 그 내용이 응용 프로그램으로 전달되고, 문서의 끝까지 파싱이 완료되면 파서의 수행이 멈추기 때문에, 응용 프로그램이 필요한 구성 요소에 대한 요청이나 조종이 불가능하며, 유연성을 가지기 어렵다.

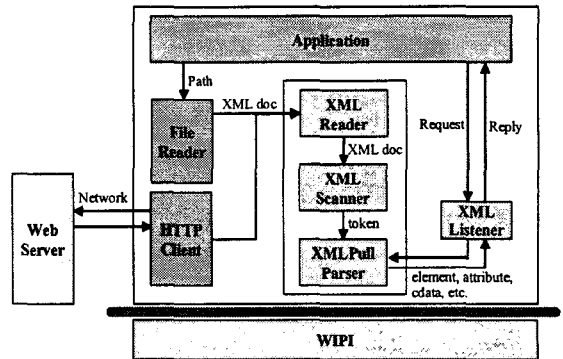
본 논문에서는 이러한 기존의 파서가 가지는 문제점을 해결하기 위하여 응용 프로그램이 필요시에 파서에 필요한 구성 요소를 파싱해 줄 것을 요청하고, 파싱된 결과를 응용 프로그램이 가져가는 Pull 방식의 파서를 개발하였으며, 이 파서는 응용 프로그램의 요청이 있기 전까지는 수행을 멈추지 않음으로써, 필요시에 언제든지 원하는 파싱 작업이 가능하며, 응용 프로그램에서 불필요한 구성 요소에 대해서는 처리하지 않기 때문에 Push 모드의 파서보다 더 효율적이다.

### 3. Pull 모드 XML 처리기의 구성 및 동작

본 논문에서는 구현한 XML 처리기의 핵심 컴포넌트인 XML 파서는 Pull 모드의 파서이다. 구현된 파서는 XML 데이터의 어느 한 구성 요소에 접근하기 위해서 전체 문서를 모두 파싱할 필요가 없다. 또한, 파서가 XML 구성 요소를 응용 프로그램의 이벤트 처리 메소드에 Push 하도록 허용하는 기존의 SAX 파서의 동작을 따르지 않는다. 대신에 응용 프로그램이 호출할 수 있는 Pull 메소드를 구현하였으며, 파싱의 제어권을 파서가 아닌 응용 프로그램이 가진다. 따라서, XML 정보의 파싱을 위한 처리 시간에서 효율적이며, 불필요한 파싱 작업이 수행되지 않음으로써 버퍼 사용이 최소화 되어 메모리의 소모가 적다. 특히, 기존의 Push 모드 파서는 XML 문서 전체가 파싱의 기본 단위이지만, Pull 모드 파서에서는 XML의 구성 요소가 기본 파싱의 단위이므로, 네트워크를 통해 스트리밍 되는 XML 문서의 파싱도 가능하다.

본 논문에서 개발한 XML 처리기는 [그림-1]과 같이 HTTPClient, FileReader, XMLReader, XMLScanner,

XMLPullParser, XMLListener 등 6개의 핵심 모듈로 구성된다.



[그림-1] XML 처리기의 구성 및 동작

FileReader와 HTTPClient는 XML 문서를 파일 또는 원격지로부터 읽어 들이는 역할을 한다. 응용 프로그램이 요청하는 XML 문서가 현재 단말에 파일로써 존재하는 경우는 FileReader가 해당 파일을 읽어, 메모리 상에 저장하여, XML 파서가 파싱 할 수 있도록 한다. HTTPClient는 모바일 단말 상에 존재하는 XML 문서가 아닌 인터넷 상의 다른 시스템에 존재하는 XML 문서를 모바일 단말로 다운로드 받아 XML 파서가 이 XML 문서를 파싱하여 문서의 적합성 및 구분 분석을 수행할 수 있도록 한다. HTTPClient는 HTTP 프로토콜을 통해 웹 서버에 접속하여, 서버로부터 XML 문서를 WIPI 단말기로 전송 받을 수 있다. 또한, XML 문서를 파싱하는 과정에서 외부 파일이 참조되는 경우, 해당 파일을 다시 해당 서버에게 요청하여 단말기로 전송받아 작업을 처리한다. 처리해야 할 XML 문서를 한꺼번에 단말로 다운로드 받지 않고, 필요시에 서버에게 전송 요청하므로, 모바일 단말의 리소스 낭비를 줄일 수 있으며, HTTP 서버와의 접속 및 필요 정보를 쉽게 얻어 올 수 있도록 유용한 메소드를 정의하여 제공하고 있다. 또한, HTTPClient는 다른 XML 처리기 구성 요소와 독립적으로 동작하도록 하여, 사용자가 필요로 하지 않을 경우 해당 부분을 포함하지 않을 수 있도록 모듈화 되어 있다.

XMLReader, XMLScanner, XMLPullParser는 Pull 모드 XML 파서의 핵심 모듈이며, 각각은 독립적인 쓰레드로 수행된다. XMLReader는 FileReader 또는 HTTPClient로부터 전달 받은 XML 문서를 스트링 형태로 변환하여, XMLScanner로 전달한다. 이러한 과정을 거치는 이유는, WIPI Java에서 제공하는 데이터 형식 변환 방법을 자유롭게 적용 할 수 있도록 하기 위해서이다. XMLScanner는 XML 문서를 토큰 단위로 분리하여 XMLPullParser에게 전달하는 역할을 하는데, 이때, XMLPullParser는 XML 문서 구조의 적합성 여부를 확인하고, 토큰을 XML의 구성 요소인 element, attribute, character data 등으로 구분한다. XMLPullParser는 응용 프로그램의 요청이 있을 시에 수행되는 모듈로 동작의 제어권이 응용 프로그램에 있으며, XMLScanner는 XMLPullParser의 요청이 있을 시에 수행되며 동작의 제어권이 XMLPullParser에 있다. 처리기를 구성하고 있는 EventListener는 응용 프로그램과 XMLPullParser간의 인터페이스 역할을 수행한다. 응용 프로그램의 요청을

XMLPullParser로 전달하고, XMLPullParser의 수행 결과에 대한 응답을 응용 프로그램으로 전달한다. 따라서, EventListener에는 응용 프로그램과 XMLPullParser에서 정보 요청 및 전달을 위해 필요로 하는 메소드가 정의되어 있다. 정의된 메소드는 [표-1]과 같다. 응용 프로그램이 파싱하고자 하는 구성 요소를 EventListener를 통해 요청하면, XMLPullParser는 XMLScanner에게 넘겨주고, XMLScanner는 메모리 상의 XML 문서를 읽어 가면서 token을 인식하고, 얻어진 token을 XMLPullParser에게 넘겨 준다. XMLPullParser는 문법적 결함 여부를 판단 한 후, 구성 요소를 결정하게 되며, 그 결과를 EventListener를 통해 응용 프로그램에 알린다. 응용 프로그램에서는 EventListener가 제공하는 메소드를 사용하여 구성 요소의 내용을 읽어오고, 해당 작업을 수행한다. 이러한 과정은 계속 반복된다.

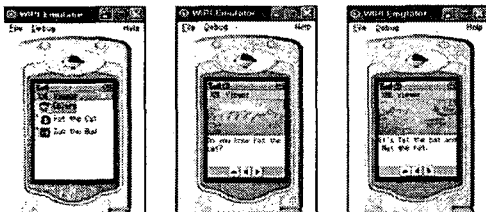
XML 처리기는 WIPI 기반으로 구현되었으며, XML 처리기를 이용하는 응용 프로그램으로 간단한 모바일 단말용 전자책을 개발하여, XML 처리기의 동작을 확인하였다.

[표-1] XML 파서의 EventListener 모듈에 정의된 메소드

메소드	설명
startParser	Parser를 시작한다.
stopParser	Parser를 멈춘다.
hasNext	XML 문서에서 구성 요소가 있는지 확인한다.
next	다음 구성 요소의 처리를 요청한다.
setEventType	처리한 구성 요소의 타입을 설정한다.
getEventType	처리한 구성 요소의 타입을 얻는다.
setElementName	처리한 구성 요소의 이름을 설정한다.
getElementName	처리한 구성 요소의 이름을 얻는다.
setText	처리한 구성 요소에 동반되는 내용을 설정한다.
getText	처리한 구성 요소에 동반되는 내용을 얻는다.
setAttributeCount	처리한 구성요소가 가진 속성의 개수를 설정한다.
getAttributeCount	처리한 구성요소가 가진 속성의 개수를 얻는다.
setAttributeName	속성의 이름을 설정한다.
getAttributeName	속성의 이름을 얻는다.
setAttributeValue	속성의 값을 설정한다.
getAttributeValue	속성의 값을 얻는다.
notifyResult	처리 결과를 알려준다.

4. 응용

본 연구에서 개발된 XML 파서를 기반으로 하는 응용으로 전자책을 볼 수 있는 간단한 Book-Viewer를 개발하였다.



[그림-2] 전자책 실행 예

Book-Viewer에서 처리하고 있는 element와 attribute는 [표-2]에 보여준 바와 같다. [그림-2]는 Aroma WIPI 에뮬레이터 상에서 [7] 웹 서버와 접속하여 Book-Viewer에서 동화책을 보여주는 예제이다. XML 문서와 이미지 파일 등은 <http://www.younmeus.net> 서버로 접속하여 전송받았다.

[표-2] 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
attribute	src	파일의 위치 image, sound element에서 사용
	loop	재생 횟수 sound element에서 사용
	href	이동할 페이지 prev, next, home element에서 사용

5. 결론 및 향후과제

본 논문에서는 CPU 부하를 덜어주고, 메모리 사용량을 줄이면서, 크기가 작은 Pull 모드 XML 파서와 뷰어를 핵심 컴포넌트로 가지는 XML 처리기가 구현되었으며, 그 응용으로 전자책을 개발하였다. 향후에는, 개발한 XML 처리기의 성능 향상 방법을 연구하고, 네트워크를 통해 스트리밍 되는 문서를 처리할 수 있는 XML 응용 프로그램을 개발할 계획이다.

참고 문헌

- [1] E. A Lee, "What's Ahead for Embedded Software?," IEEE Computer, Vol. 33, No. 9, Sept. 2000
- [2] 이상윤 외, "한국 무선 인터넷 표준 플랫폼(WIPI)의 표준화 현황 및 발전 전망," 한국정보과학회지, 22권 1호, 2004
- [3] 홍준성, "모바일 플랫폼 기술현황 및 발전방향," 한국정보과학회지, 22권 1호, 2004년 1월
- [4] XML, <http://www.w3c.org>
- [5] 감미연 외, "WIPI 기반의 모바일 단말을 위한 내장형 XML 파서 및 뷰어," 한국정보과학회 춘계학술대회 논문집, 2004
- [6] 박수원 외, "위피 모바일 프로그래밍," 한빛미디어, 2003
- [7] MobileJava, <http://www.mobilejava.co.kr>