구성적 임베딩을 위한 모듈 기반의 XML 처리기의 설계

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

A Design of Module-based XML Processor for Structural Embedding

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, 홈패드, Hand-held PC 등 low-end로부터 high-end까지의 다양한 규모의 하드웨어 자원을 가지는 유무선 단말들이 속속 등장하고 있다. 그리하여, 이들로 임베딩되는 소프트웨어들도 과거 그들이 지녔던 경직성(fixedness)에서 탈피하여, 다양한 장치들을 수용할 수 있도록 구성적 특성을 요구받고 있다. 웹 상의 정보 표현을 위한 표준으로 자리잡은 확장성 표기 언어인 XML을 위한 처리기는 그러한 단말에서의 자료 브라우징을 위해 필수적으로 임베딩 되어야할 소프트웨어 중의 하나이다. 본 논문에서는 다양한 규모의 단말들에 구성적 임베딩이 가능한 모듈 기반의 XML 처리기가 설계, 구현된다.

1. 서 론

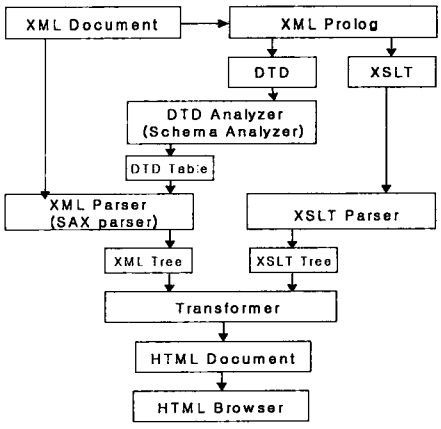
특정 하드웨어에 가장 적절하게 고정되는 경직성(fixedness)은 임베디드 소프트웨어가 가지는 중요한 특성 중의 하나이다. 이는 고속의 실시간 처리를 위해 취할 수밖에 없었던 설계 방법이라고 볼 수 있으며, 과거에는 그러한 실시간 장치들이 기능적 단순성을 가지고 있었으며, 극히 제한된 분야에서만 사용되었다[1]. 그러나, 최근 들어, 유, 무선 인터넷 기술의 발달로, low-end PDA로부터 high-end hand-held PC까지 매우 다양한 기능과 다양한 규모의 하드웨어 자원을 가지는 Post-PC 단말들이 속속 개발되어 상품화되고 있는 실정에 비추어 볼 때, 이제 임베디드 소프트웨어는 현존하는 하드웨어 및 소프트웨어 기술에 대한 적응성이 뛰어나도록 설계 개념이 바뀌어야 한다는 시각이 나타나고 있다[1]. 그리하여, OS 분야에서는 공개 소스 기반의 임베디드 OS를 지향하고 있으며[2], 프레임워크 기반의 임베디드 소프트웨어의 개발을 위한 연구도 활발히 진행 중에 있다[1,3]. 또한, 무선 인터넷 기반의 플랫폼으로 많이 이용되는 Java의 경우, 그 단점인 속도의 저하 문제를 극복하기 위해, 실시간 특성을 추가하는 것과 단말로의 효율적 임베딩을 위한 연구가 활발히 진행 중이며[4, 5], 각종 단말 장치들이 웹 서버로서의 기능을 수행하는 응용이 적지 않아, 최근에는 임베디드 웹 서버에 관한 연구도 활발히 진행 중에 있다[6, 7, 8].

확장성 표기언어인 XML이 웹 상의 정보 표현의 표준으로 결정됨에 따라, 웹 상의 콘텐츠 표현의 주요 표기 언어인 XML 처리를 위한 파서 엔진, XML 브라우저 등도 이제, 각종 단말의 필수적인 소프트웨어 중의 하나가 되었다. 이러한 XML 처리기는 파서 엔진을 가장 기본적인 콤포넌트로 가지고 있으며, C[9, 10, 11] 혹은 Java[12, 13] 언어를 기반으로 개발되어 발표되었다. 이러한 XML 파서 엔진을 축으로 하는 XML 처리기도 이제는 임베디드 소프트웨어로서의 그 기능을 수행함에 있어서, 뛰어난 구성적 적응성이 요구되고 있는 것이다. 본 논문에서는 다양한 규모의 단말에 효율적으로 임베딩 시키기 위한, 모듈 기반의 XML 처리기인 XML-Pro가 설계 구현된다. Low-end PDA 등을 위한 경량급 XML parser의 구성 및 high-end 단말을 위한 XML parser의 구성이 XML-Pro를 사용하여 가능하다.

본 연구는 2001년도 산업자원부 중기거점과제 "디지털 가전형 POST-PC 플랫폼 기술개발사업" 지원으로 이루어졌음.

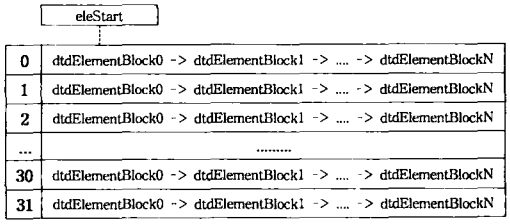
2. XML-Pro의 구성 및 동작

XML-Pro는 7개의 모듈로 구성된 XML 문서처리 엔진이다. DTD-Analyzer, Schema-Analyzer, XML-Parser (Valid parser Well-formed parser), SAX-Parser, XSLT-Parser, XSLT2HTML-Transformer 등 7개의 모듈이 그것들이다. Schema-Analyzer와 SAX-Parser는 현재 구현 중에 있으며, 그 전체적인 동작 과정은 [그림-1]에 보여준 바와 같다.



[그림-1] XML-Pro의 동작과정

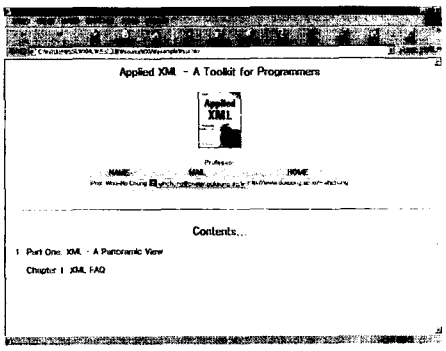
DTD는 XML 문서의 문법을 기술한 것으로, 크게 ENTITY, ELEMENT, ATTLIST로 구성되며 element 이름, element 간의 계층관계, attribute의 이름과 attribute 값, 형식 등에 대해서 정의하고 있다. XML 문서의 파싱은 문서에 대한 DTD를 기반으로 이루어진다. DTD-Analyzer는 DTD 문서를 scanning하면서 DTD Table을 구성하며, 0부터 31까지 32개의 entry로 구성되며, DTD에 정의된 element는 element 이름의 string 값을 합하여 modulo-32한 값과 일치하는 entry에 저장된다. 이는 파싱에 있어서 element 탐색을 빠르게 하기 위한 hashing table 역할을 하고 있다. 그 구조는 [그림-2]에 보여준 바와 같다.



[그림-2] DTD Table과 element block의 구성

XML-Parser는 XML 처리기의 가장 기본적인 모듈로, DTD와 XML 문서를 입력받아 DTD의 구문 검사 및 정당성 검사를 하고 인스턴스를 tree 형태로 변환하여 메모리상에 구성함으로써 사용자의 응용 프로그램이 문서에의 접근 또는 변경을 용이하게 한다. XML parse tree는 XML 문서에 사용된 모든 element에 관한 정보를 저장하고 있으며, XML 문서를 scanning하면서 각종 wellformedness 및 validity 여부를 조사하면서 구축된다.

XML은 구조적 정보만을 가지고 있고 layout 정보는 가지고 있지 않기 때문에 XML을 시각적으로 표현하기 위해서는 XML이 표시되는 방식에 대한 규정을 담고 있는 확장성 양식 언어 XSL을 사용한다. 이러한 XSL은 XSL-fo를 사용하는 방법과 XSLT를 사용하는 2가지 경우가 있으며, 본 논문에서는 후자를 따르기로 한다. 이는 XML 문서를 HTML로 변환을 거쳐, 기존의 브라우저를 통해 시각화시키는 방법이다. 이 경우에도 XSLT를 사용한 layout 문서에 대한 구문의 올바름을 조사해야하는데, XSLT-Parser가 이 기능을 담당한다. 그 결과물로 XSLT parse tree가 도출된다. 이렇게 하여 얻어진, XML parse tree와 XSLT parse tree를 구성하는 XSLT 템플릿과 패턴을 연결하여, XSLT2HTML-Transformer는 XML 문서를 HTML 문서로 변환 생성한다. 새로 만들어진 HTML 파일에 있는 내용은 XML 문서 파일로부터 추출한 정보이고, 포맷 구조는 XSLT에서 추출한 정보이다. 생성된 HTML 파일은 MSIE, Netscape 등 HTML을 지원하는 브라우저를 통해 보여진다. XML을 HTML로 변환함으로써 일반적으로 XML을 웹에서 이용하려면 MSIE 5.0 이상의 브라우저만을 이용해야한다는 제약을 해결하여, 인터넷 사용자들이 브라우저의 업그레이드나 새로운 application의 설치 없이 XML로 이루어진 문서를 사용할 수 있다. [그림-3]은 XSLT2HTML-Transformer에 의해 생성된 HTML 문서를 MSIE에서 브라우징한 모습이다. 본 논문에서 구현한 Transformer는 35개의 XSLT element 중 가장 많이 사용되는 11가지의 XSLT element만을 처리하고 있다. 이 element들은 실질적으로 XML 문서를 표현하는 데에 핵심이 되는 element들로써, 대부분의 XSLT 문서에서 사용하는 element들이다. 자세한 동작은 [11]을 참조하기 바란다.



[그림-3] XML 문서를 브라우징 한 모습

3. XML-Pro 비교 분석

XML-Pro를 구성하고 있는 각 수행 모듈의 크기는 [표-1]에 보여준 바와 같다. 현재, 구현 중에 있는 모듈은 그 기능을 포함할 수 있는 상위 모듈의 크기를 최대로 하여 추정하였다. 즉, Schema-Analyzer는 DTD-Analyzer를 기준으로 하였으며, SAX-Parser는 XML-Parser 보다 훨씬 작은 크기를 가지리라 예측되며, 실제로 Java로 구현된 SAX-Parser의 경우에도 대부분 10K 미만의 크기를 가지는 것으로 알려져 있다[13].

[표-1] XML-Pro를 구성하는 모듈의 크기

	XML-Pro모듈	크 기	비 고
1	DTD-Analyzer	28.5K	
2	Valid XML-Parser	13.5K	
3	Well-formed XML-Parser	< 13.5K	
4	(XSLT 파서 + Transformer)	38.5K	
5	공통모듈	21.5K	
6	Schema-Analyzer	28.5K(추정)	N/A
7	SAX-Parser	<< 13K(추정)	N/A

C로 구현된 대표적인 XML 파서 엔진인 Expat, RXP, 그리고 XML-Pro의 크기와 문자 처리 속도에 관한 비교가 [표-2]에 나타나 있다. 여기서, Expat과 RXP와의 비교를 위해, XML-Pro에서 XSLT-Parser와 Transformer를 제외하였다. [표-2]에서 보아 알 수 있듯이, XML-Pro가 크기 면에서 매우 경제적으로 구성되어 있음을 알 수 있다. 그리고, 샘플 XML 문서를 사용하여 단위 시간당 문자 처리 속도는 Expat이 XML-Pro에 비해 빠르게 나타나고 있는데, 이는 하드웨어 속도 차이에서 비롯된 것이다. 즉, XML-Pro는 UltraSparc 800MHz를 기반으로 하였고, Expat은 Pentium-IV 1.5GHZ를 기반으로 하였기 때문이다. RXP의 경우, 처리 속도가 현저히 떨어지는데, RXP는 parse tree를 화면으로 출력하는 시간까지 포함하고 있기 때문이다. 참고로, Java 기반의 XML parser 엔진이며 IBM에서 개발된 XML4J[13]는 약 500K의 크기를 가지고 있어, 자원의 규모 면에서 high-end Post-PC급에 적당하며, 경량급 고속성을 요구하는 장치에는 적당하지 않음을 알

수 있다.

[표-2] C 언어 XML 파서 엔진의 크기 및 처리 속도 비교

	XML-Pro	Expat	RXP
사용언어	C	C	C
시스템	Unix(Linux)	Windows-XP	Unix (Linux)
크기	63.5K	323K	164K
구성모듈	· DTD분석기 · Valid XML파서 · 공통모듈	· xmlparse.dll · xmltok.dll · xmlwf.exe	
문자처리속도	912Kcps	1160Kcps	56.2Kcps

4. XML-Pro의 구성적 적응성

XML-Pro는 하드웨어 자원이 충분하지 않은 low-end PDA를 위한 경량급 XML 처리기와, 충분한 하드웨어 자원을 가진 high-end Post-PC급 단말 장치를 위한 고급의 XML 처리기를 다르게 구성하여 임베딩 시킬 수 있도록 하고 있다. 예를 들어, 메모리가 매우 작은 low-end PDA의 경우, 경량급 XML 처리기를 필요로 하므로, 그 기능이 단순한 SAX level의 SAX-Parser만 구성할 수 있으며, 메모리가 충분한 경우에는 DOM level parser의 구축을 위해, DTD 분석기(혹은 Schema 분석기), valid XML parser, 그리고 viewing까지 가능하게 하려면 XSLT parser와 Transformer까지를 구성하여 임베딩 시키면 된다. 만약에 DTD를 사용하지 않는 경우에는 valid parser 대신에 well-formed parser를 사용하여 XML 처리기를 구성할 수 있도록 하고 있다. XML-Pro를 사용하여 구성이 가능한 XML 처리기의 구성 예가 [표-3]에 나타나 있다.

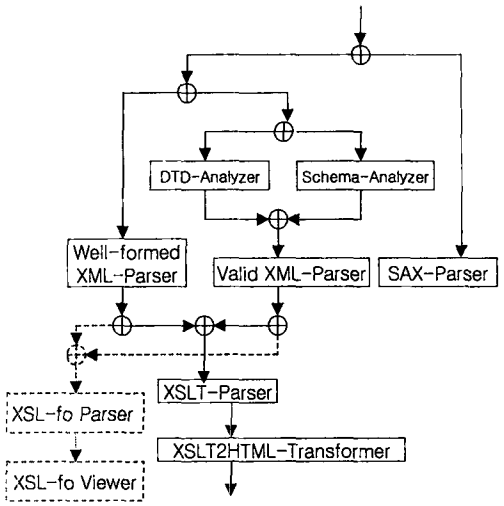
[표-3] XML-Pro로 구성하는 XML 처리기 예

	XML처리기	구성모듈	수행프로그램크기	공통모듈 포함
1	경량급 XML처리기	· SAX-Parser	<< 13K(추정)	<< 34.5K
2	중량급 XML처리기 (with DTD)	· DTD-Analyzer · Valid XML-Parser	42.0K	63.5K
3	중량급 XML처리기 (without DTD)	· DTD-Analyzer · Well-formed XML-Parser	< 42.0K	< 63.5K
4	FULL XML 처리기	· DTD-Analyzer · Valid XML-Parser · XSLT-Parser · Transformer	80.5K	102.0K

현재 XML-Pro에서 구성 가능한 XML 처리기 모든 예를 보여주고 있는 것이 [그림-4]이다.

5. 결론 및 향후과제

본 논문에서 설계, 구현한 XML-Pro는 다양한 규모의 단말에 효율적으로 임베딩 가능한 높은 구성 적응성을 가진 XML 처리기이다. 이러한 적응성은 향후 임베디드 XML을 위한 유용한 기반 기술을 제공할 수 있다.



[그림-4] XML-Pro에서 구성 가능한 XML 처리기 방식

현재, Schema-Analyzer, SAX-Parser 등이 구현 중에 있으며, 향후, [그림-4]의 점선으로 표시된 XSL-fo 기반의 parser와 viewer를 이용하여 XML 문서를 다른 문서 형식으로 변환하지 않고도 직접 표현될 수 있도록 확장되어야 할 것이다. 그리고 보다 효율적인 Embedding을 위한 방법과 속도 개선을 위한 방안이 연구되어야 할 것이다.

참 고 문 헌

[1] E. A Lee, "What's Ahead for Embedded Software?," IEEE Computer, Vol. 33, No. 9, Sept. 2000
[2] S. Ortiz Jr., "Embedded OSs Gain the Inside Track," IEEE Computer, Vol. 34, No. 11, Nov. 2001
[3] A. D. Pimentel et al, "Exploring Embedded-Systems Architectures with Artemis," IEEE Computer, Vol. 34, No. 11, Nov. 2001
[4] D. Mulchandani, "Java for Embedded Systems," IEEE Internet Computing, Vol. 2, No. 3, May/June 1998
[5] A. Abrardo and A. L. Casini, "Embedded Java in a Web-Based Teleradiology System," IEEE Internet Computing, Vol. 2, No. 3, May/June 1998
[6] N. Witchey, "Designing an Embedded Web Server," IEEE Internet Computing, Online, 1998
[7] http://www.virata.com/products/emserver1.htm, "Embedded Web Server," 2000
[8] http://www.virata.com/products/emserver2.htm, "Embedded Web Server Specifications," 2000
[9] R. Tobin, RXP, http://www.cogsci.ed.ac.uk/~richard
[10] J. Clark, expat, http://www.jclark.com./xml/expat.htm
[11] 강미연 외, "효율적 XML/XSLT 처리 시스템의 설계 및 구현," 2000 한국인터넷정보학회 추계학술대회 논문집
[12] J. Clark, XT, http://www.jclark.com/xml/xt.htm
[13] XML4J, "http://www.alphaworks.ibm.com/text/xml4j