모바일 장치를 위한 XML 파서의 설계

장주현° 노희영
강 원 대 학 교 컴 퓨 터 과 학 과
jjangju98@naver.com°, rohhy@kangwon.ac.kr

Desing of XML Parser for Mobile device

Ju-Hyun Jang° Hi-Young Roh
Dept. of Computer Science, Kangwon National University

요 약

XML은 기존의 HTML과 SGML의 단점을 보안한 마크업 언어로써, 큰 대역폭, 많은 메모리 양, 높은 CPU속도를 가지는 유선 환경에서부터 저 대역폭, 적은 메모리 양, 낮은 CPU속도를 가지는 모바일 장치까지 사용이 확대되어 지고 있다. 현재 모바일에서 사용되는 XML 파서중에서는 PULL 모델 기반 Kxml파서[1]만이 모바일 장치를 고려한 파서이다. 모바일 장치에서 XML의 많은 사용을 위해서는 저메모리를 사용하며 빠른 파서에 대한 연구가 필요하다. 본 논문에서는 piccolo파서에서 사용한 Parser generator tool인 JFlex를 사용하고, 파싱 모델중 가장 빠르고 저 메모리를 사용하는 Pull 모델을 적용함으로써, 빠른 Token 추출과 이벤트 형 정의를 통해 좀더 빠른 XML파서를 제안하고자 한다.

1. 서 론

XML(eXtensible Markup Language)은 기존의 SGML(Standard Generalized Markup Language)과 HTML(Hyper Text Markup Language)의 장점을 혼합한 마크업 언어이다. 또한 최근에는 Internet의 발달과 통신망의 발달로 인해 mark-up 언어가 Personal Computer, Server Computer 들과 같은 유선 장치만이 아니라 PDA, cellular-phone등 무선 network 장치에도 많이 사용되어진다. 이 장치들은 낮은 대역폭, 한정된 메모리, 한정된 저장매체라는 제한된 상황에 있으며, 이러한 무선 장치에서도 문서의 정확한 표현이 가능해야 한다.[2]

현재 사용되고 있는 파싱 모델로써는 Object, Push, Pull의 모델이 있으며 이 중에 가장 빠르고, 저 메모리를 사용하는 파싱 모델로는 Pull 모델의 파서이다.[3] Pull 모델의 파서는 기존 Push 모델을 변형한 것으로, 2002년에 Pull 모델의 표준화로 인해 응용프로그래머에게 많은 관심을 끌고 있다. 또한 Pull 모델의 표준화 이전에 벤치마킹결과에서 가장 빠른 속도를 나타낸 piccolo 파서는 기존의 파서 구현방식과는 다르게 hand-wirte방식이 아닌 parser geneator tool인 JFlex와 BYacc를 사용함으로 기존 파서 보다 월등한 파싱 속도를 나타내는 방안에 대해 연구하였다.[4]

Kxml 파서는 Killobyte xml의 준말로써 Killobyte를 사용하는 장치, 즉 적은 메모리를 사용하는 장치를 위한 파서이고, 이를 위하여 Pull 모델을 적용하였다. 또한 현재 Applets, MIDP(Mobile Information Device Profile), CLDC(Connected Limited Device Configuration)을 기반으로 한 Palm pilots나 cellular phone에서 사용되고 있다.[1] 본 논문에서는 모바일 환경에서 빠른 파싱을 하는 파서를 구현하기 위한 방안으로, Pull 모델을 적용하고, 구현방안으로는 JFlex를 이용하 빠른 파싱을 하는 파서를 제안하고자 한다.

2. 관련연구

2.1 JX-PullParser

JX-PullParser는 기존의 파서보다 빠른 파싱을 하기 위하여 Parser Generator tool인 JFlex, BYacc/J를 사용하였다. JFlex는 어휘분석 도구로써, BYacc/J는 구문분석 도구로 사용하여 기존 파서보다 빠른 파싱 속도를 내었다. 아래 그림 1은 JFlex와 BYacc/J를 이용한 JX-Pull 파서의 구조 이다.

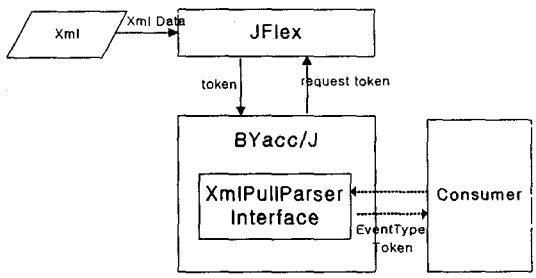

그림 1. JX-PullParser의 구조

JX-PullParser는 기존의 파서와의 벤치마킹에서 기존 파서보다 빠른 속도를 나타내었다. 특히 모바일 장치에서 사용하는 soap문서의 파싱은 다른 문서들의 파싱에서 보다 나은 파싱 속도를 나타내었다. 아래 그림 2는 기존 파서들의 벤치마킹 결과이다.

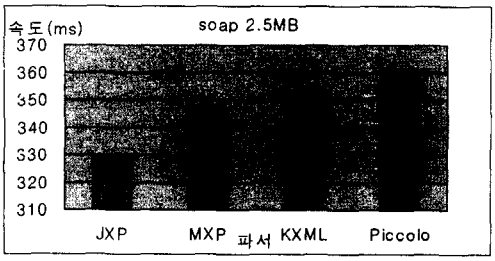

그림 2. soap문서의 기존 파싱 속도

위 그림에서 비교한 JXP는 JX-Pullparser이고, MXP, KXML은 Pull 기반의 Piccolo는 Push기반의 파서이다. Piccolo파서는 Parser Generator tool을 사용한 파서이다.

2.2 Kxml 파서

Kxml 파서는 기존의 파서같이 hand-write 구현 방법을 사용하였다. 또한 XML namespace를 지원하고 SGML 형태를 지원하는 파서이다. Kxml 파서의 가장 큰 특징은 적은 메모리를 사용한다는 것이다. 또한 기존 PullParser들이 지원하지 않는 메모리 구조 지원을 위해서, kDom을 옵션으로 지원한다. 또한 모바일 장치를 위한 WBXML, WML을 옵션으로 지원한다.

또한 Kxml파서의 생성자에서는 현재 시스템에서 사용할수 있는 메모리를 체크 하여 읽어드릴 버퍼의 크기를 설정한다.

```
srcBuf = new char[Runtime.getRuntime().freeMemory()
 >= 1048576 ? 8192 : 128];
```

그림 3. Kxml의 버퍼 크기 설정 알고리즘

위의 그림3은 Kxml에서 버퍼 크기 설정 알고리즘이다. 알고리즘은 현재 사용할수 있는 시스템의 메모리를 판단하여 104876 byte보다 크면 8192를 배열 크기로 작으면 256을 버퍼의 크기로 할당한다.

3. Mob-XP 파서의 설계

본 논문에서 제안한 파서는 아래 그림 4와 같다.

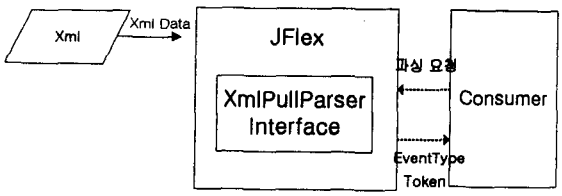

그림 4. Mob-XP 파서 구조

기존의 JX-Pullparser와는 달리 파싱에 관련된 모든 처리는 JFlex의 단독 수행으로 이루어진다. 응용프로그램의 파싱요청이 생기면 JFlex는 설정된 파일을 읽고 파일의 Token 추출과 이벤트 형을 정의하고, 정의된 이벤트 형은 응용프로그램의 요청시에 반환하게 된다.

본 논문에서 제안한 파서의 Token정의와 구문정의는 XML spec(Third edition)[5]을 기준으로 한다. 또한 namespace를 위해서 1998년 11월에 정의된 namespace spec과 1999년 1월에 정의된 namespace spec에 정의된 접두사정의를 모두 지원한다. 또한 본 논문에서 제안한 파서는 validating을 지원하지 않는 non-validating 파서이다. 이는 현 xml의 발전 추세와 외부 validating 툴이 많이 존재 하고, 또한 모바일 장치에서의 외부 dtd사용이 점차 줄어드는 추세를 반영한 것이다.

3.2 버퍼 크기 할당 알고리즘

Kxml 파서에서는 버퍼 크기를 두 가지만을 사용하였다. 하지만 본 논문에서는 이를 좀더 세분화시켜서 세밀한 메모리를 할당이 가능하게 하였다. Kxml과 같이 현재 사용할 수 있는 메모리량을 얻어내고 그에 따른 메모리 할당을 한다.

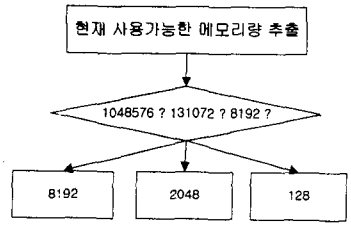

그림 5. 버퍼 크기 할당 알고리즘

3.3 XML 선언 및 엔티티 리퍼런스

본 논문에서 사용한 XML 선언의 처리 방법은 JX-Pullparser에서 사용한 방법과 동일한 방법을 사용하였다. XML 선언 토큰의 추출시에 얻어지는 정보들은 변수에 할당하고, 할당된 값을 응용프로그램의 요청시에 반환하는 구조로 파싱이 된다. 엔티티 리퍼런스의 처리는 해쉬테이블을 이용하였다. XML spec에 미리 정의되어 있는 엔티티 들은 파싱의 시작과 함께 엔티티를 키로 리퍼런스 값을 값으로 하여 저장된다. 내부 엔티티정의에 있어서도 해쉬테이블을 이용하여 기존에 정의된 엔티티의 유,무를 판단하여 키와 값을 저장하게 된다. XML 문서내에서 엔티티 토큰의 발견시에는 해쉬테이블을 참조하여 값을 반환해 준다.

4. 결론 및 향후 연구 과제

본 논문에서는 모바일 장치에서 사용하는 XML의 빠른 파싱을 위해서 모바일 파서의 설계에 대해 제안하였다. 빠른 파싱을 위해서 Pull모델을 적용하였고, 구현방안으로 JFlex를 사용하였다. 또한 Kxml파서에서 사용한 버퍼 할당 방법을 좀더 세분화 시켜서 적용하였다. 사용된 것들을 그대로 사용하였다. 또한 Token정의와 문법 정의는

XML spec(third edition)을 적용시켰다. 향후 연구과제로는 파싱의 시작시에만 버퍼 할당을 하는 것이 아니라 파싱의 중간,중간에 좀더 동적인 버퍼 할당 방법에 대해 연구해야겠고, 제안한 파서의 구현 및 모바일 장치에서 다른 파서와의 속도비교 역시 향후에 연구해야 할 과제이다.

참고문헌

[1]Stefan Haustein, http://kxml.enhydra.org/index.html

[2]Dennis Sosnoski , http://www.ibiblio.org/xml/

[3]Dennis M. Sosnoski , http://www.javaworld.com/javaworld/jw-02-2002/jw-0208-xmljava_p.html

[4] 장주현,노희영 JFlex와 BYacc/J를 이용한 Xml Pull Parser 설계, 한국정보과학회 추계 학술발표논문집, 제 30권 제1호 (B), 31 ~ 33

[5]http://www.w3.org/TR/2004/REC-xml-20040204