휴대단말용 다중 마크업 문서 파싱 시스템 설계

최은정[0] 손지연 한동원
한국전자통신연구원
(ejchoi[0], jyson, dwhan)@etri.re.kr

Design of Multi-document Parsing System for Mobile Device

Eun-Jeong Choi[0] Ji-Yeon Son Dong-Won Han
Electronics and Telecommunications Research Institute

요 약

본 논문에서는 휴대단말용 유무선 통합 브라우저를 위한 다중 문서 파싱 시스템을 제안한다. 현재 유무선 인터넷 서비스를 지원하기 위해 단일화된 표준 마크업 언어가 없다. 따라서, 유무선 통합 브라우저를 설계하려면 이들 각기 다른 마크업 언어의 지원을 생각하지 않을 수 없다. 이를 지원하기 위해 본 논문에서는 모든 마크업의 공통 분모 격인 파서를 설계하였으며, 각기 다른 사양의 휴대단말에 적합하게 표현하기 위해 그래픽 사용자 인터페이스 객체를 생성하는 방법을 제안하기로 한다. 이를 위해, 파서는 마크업 언어의 그래픽 기능을 휴대단말에서 지원 가능한 그래픽 사용자 인터페이스 객체 형태의 결과물을 만들어 내다. 이 결과물은 추후에 브라우저의 사용자 인터페이스 모듈과 연동될 것이다. 이러한 파싱 시스템은 브라우저로 하여금 모든 언어를 파싱할 수 있도록 하는 한편, 여러 언어 표준을 지원하려는 브라우저에 부담을 최소화시키는 기법이다.

1. 서 론

휴대폰의 사용이 일반화 되면서 무선 인터넷 서비스를 지원하기 위해 기존의 HTML(Hyper Text Markup Language)로부터 파생된 마크업 언어들이 속속 등장하게 되었다.

기존의 HTML을 사용해 무선 인터넷을 서비스하지 않고, 다른 언어를 개발하게 된 데에는 크게 무선 선로와 휴대 단말의 제약 사항들 때문이었다. 무선 선로는 대역폭이 좁고, 에러발생률이 높으며, 휴대 단말은 화면 창 크기가 작고, CPU(Central Processing Unit), 메모리 등의 컴퓨팅 능력이 데스크 탑 PC(Personal Computer)에 비해 상대적으로 열등하다. 반면, 기존의 유선망에서 제공되던 HTML은 기능이 많고 처리과정이 복잡하여 휴대단말에 지원하기 어려웠던 것이다. 그래서, HTML의 기능 중 일부를 그대로 상속하는 한편 각 단말에 특화된 마크업 언어를 개발하게 된 것이다. 대표적인 예로써, HDML(Handheld Device Markup Language), WML(Wireless Markup Language), mHTML(Mobile HTML), cHTML(Compact HTML) 등이 등장하여 현재까지도 서비스되고 있다. 이들 마크업 언어들은 각 서비스 제공자 및 단말의 특성을 고려하여 개발된 언어들로서 서로간의 호환성은 고려되지 않고 있다.

2. 관련 연구

무선 인터넷을 지원하려는 사용자 에이전트에서 서로 다른 마크업 언어들을 지원하는 방법은 몇 가지 있다.

2.1 통일된 마크업의 지원

먼저, W3C(World Wide Web Consortium) 등에서는 프리젠테이션을 위한 모든 마크업 언어를 XHTML(eXtensible Hyper Text Markup Language) 모듈로 구성하려는 안을 내놓았다. 즉, 사용자 에이전트는 XHTML 모듈을 해석할 수 있는 파서를 내장하고, 각 단말은 특성에 따라 모듈을 구성하면 되는 것이다. 장기적으로 내다봤을 때 가장 이상적인 안이다. 그러나, 각기 다른 형태의, 다른 컴퓨팅 파워를 가진 휴대단말에서는 각 단말에 특화된 마크업 언어를 제공하고자 하기 때문에 웹을 지원하기 위한 마크업 언어를 표준화하는 데는 한계가 있다. 또한 단말기 혹은 서비스 제공자에 따라 다른 마크업 언어가 서비스되고 있는 현 시점에서는 지원 불가능하며 이러한 스펙을 따르지 않는 마크업은 지원하지 못하는 단점이 있다.

2.2 개별 파서의 도입

또 다른 방법으로는 각 마크업 언어에 따른 개별 파서 혹은 브라우저를 내장하는 것이다. 지원하고자 하는 각 언어를 파싱하기 위한 시스템 혹은 랜더링 모듈을 각각 독립적으로 구현하도록 하여 컨텐트의 종류에 따라 적당한 브라우저를 불러서 처리하도록 하는 방법이다.

대표적인 예로써, 마이크로소프트의 모바일 익스플로러를 들 수 있다. 모바일 익스플로러는 휴대폰 용 브라우저로써, WML과 XHTML을 파싱하는 모듈이 모두 내장되어 있어서 단말로 들어오는 컨텐트의 타입을 분석하여 각각의 브라우저를 실행시키는 듀얼 모드로 동작되도록 되어 있다.

이러한 방법은 단말 내에 지원하고자 하는 마크업 언어마다 동작되는 브라우저가 다르기 때문에 비효율적이다. 즉, 지원하고자 하는 마크업 언어가 수십 가지라면 같은

수의 브라우저 프로그램을 단말에 장착해야 하는 것이다. 이러한 방법의 또 다른 단점은 한가지 표준만을 지원하는 단일 문서밖에 처리할 수 없다는 것이다.

3. 파서 설계

3.1. 문서 구조 분석

본 논문에서는 현재 서비스 되고 있는 마크업 언어들을 단일 문서 구조, 내장형 구조, 모듈화 구조 등의 세 가지 형태로 분류하였다.

첫 번째는 단일 문서 구조이다. 이는 대부분의 인터넷 서비스에서 제공되는 문서의 형태로서 문서 종류는 다음과 같다.

- XHTML
- WML
- cHTML
- mHTML
- HTML

두 번째 구조는 내장형이다. 이것은 문서의 컨텐트 타입은 위 단일 문서 중 하나이지만, 문서의 루트 요소 내에 다른 마크업 언어가 하위 요소 형태로 내장되는 구조로서 다음과 같은 방식으로 제공된다.

- WML2
- namespace를 이용해 다른 형식의 마크업 문서를 내장하는 방법
- object 태그를 이용한 객체 내장
- 프로토콜을 이용한 객체 내장(mailto, http, ftp, etc.)

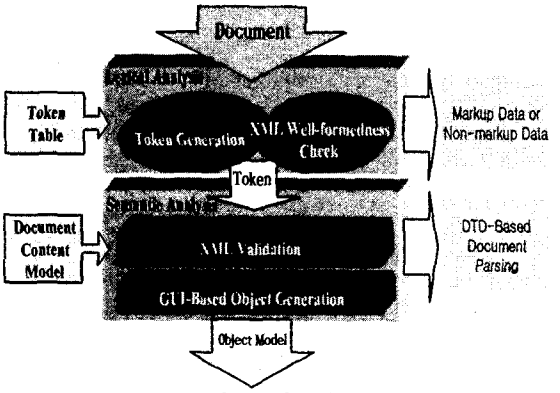
세 번째는 모듈화되어 제공되는 구조이다. 이러한 방식을 택하는 예는 W3C의 XHTML 모듈화[3]가 대표적이다.

- XHTML 모듈화

3.2. 시스템 구조

위에서 분석한 결과, HTML을 제외한 대부분의 문서는 XML(eXtensible Markup Language)을 기반으로 개발된 것들이며, HTML 역시 XML 로 전이하고 있는 추세이다. 따라서, 본 논문에서는 XML을 기반으로 하는 마크업 언어들에 기초하여 파싱 시스템을 설계하였다. 또한 위에서 제시한 문서 구조를 모두 지원할 수 있도록 설계하였다.

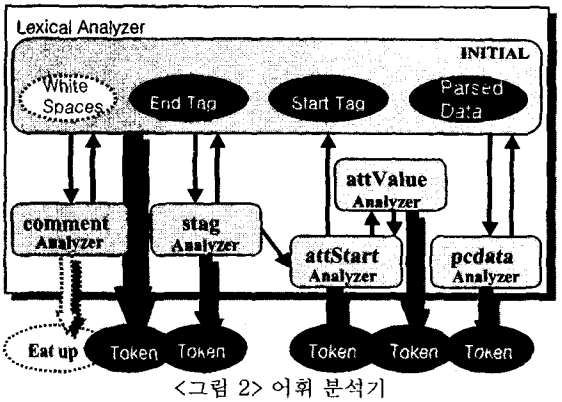
본 논문에서 제시한 방식으로 처리하기 위해서는 <그림 1>의 과정을 거친다. 본 논문에서의 파싱 시스템은 크게 어휘 분석기와 구문 분석기로 구성된다. 어휘 분석기에서는 지원하고자 하는 문서 종류에서 필요로 하는 모든 마크업 데이터를 위한 토큰 테이블을 참조하여 크게 마크업과 non-마크업을 기반으로 토큰을 분리한다. 또한 구문 분석기에서는 각 문서의 DTD(Document Type Definition)를 기반으로 컨텐트 모델을 분석하여 이를 기반으로 구문을 분석한 후, 단말의 그래픽 사용자 인터페이스를 기반으로 트리 기반의 객체를 생성하여 랜더링을 위한 데이터로 제공한다. 이후 과정은 파싱 시스템의 범위를 벗어나므로 본 논문에서 다루지 않기로 한다



<그림 1> 시스템 구조

3.3 어휘 분석

본 논문에서의 어휘 분석기는 XML Well-formedness[2] 표준에 기반하여 토큰을 추출해 내며, 이때 어휘분석기에서는 지원하고자 하는 문서들의 모든 토큰을 테이블로 구성한다. <그림 2>[2]과 같이 XML 구조에 따라 상태 변이를 하면서 토큰을 분리해 내도록 설계하였다. 이때 토큰 상태란 같은 어휘라 할지라도 어휘 분석기 상태에 따라 다른 토큰으로 분리해 내도록 하는 것을 말한다. 어휘 분석기에서의 상태는 주석문(comment), 마크업 시작(statg), 속성(attrStart, attValue), 일반 사용자 데이터(pcdata) 등으로 나뉜다.



<그림 2> 어휘 분석기

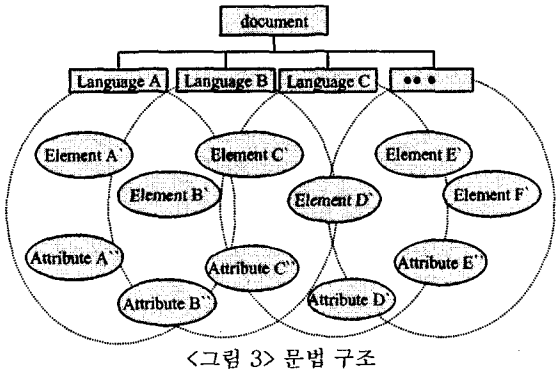
3.4 구문 분석

본 논문의 구문 분석기는 크게 두 가지 기능을 수행한다.

먼저, 각 모든 문서의 마크업을 각 단말에 적합하게 표현할 수 있도록 도와준다. 앞서 어휘 분석기에서는 XML Well-formedness를 기반으로 문서를 분석하여 토큰을 추출해 내었다. 이제 구문 분석기에서 이러한 토큰을 바탕으로 문서가 각각의 DTD에 적합하게 작성되었는지 여부를 검사한다. 또한 이 과정에서 분석된 마크업은 단말의 그래픽 사용자 인터페이스와 일치시킨다. 즉, 마크업 언어의 그래픽 사용자 인터 페이스 모델을 단말에서 지원

가능한 그래픽 사용자 인터페이스로 표현할 수 있도록 하기 위한 매핑을 하는 것이다. 이렇게 하는 이유는 단말들은 각각의 특성에 맞는 그래픽 사용자 인터페이스 요소를 가지므로, 모든 마크업 언어 표준들을 데스크 탑에서와 같이 지원할 수는 없다. 대신에, 마크업 언어의 그래픽 사용자 인터페이스 특성들을 단말의 사용자 인터페이스에 맞게 고쳐야 한다.

다음으로 본 논문에서는 여러 종류의 문서 혹은 다중 문서를 파싱 하기 위해 <그림 3>과 같은 형태의 문법 구조를 설계하였다.



<그림 3> 문법 구조

<그림 3>의 문법 구조에서 파서는 여러 종류의 표준을 지원하는 마크업 언어를 파싱할 수 있도록 하였다. 지원하고자 하는 모든 DTD를 분석하여 각 요소 별로 문법을 설계한다. 이때, 여러 언어에서 공통된 요소나 속성들을 가질 수도 있는 반면, 어떤 요소나 속성들은 특정 언어에 국한되기도 한다.

본 논문에서는 모든 프리젠테이션을 위한 마크업들의 공통 분모를 파싱하는 시스템을 설계하였다.

다음은 <그림 3>의 구조를 가지는 문법을 BNF(Backus-Naur Form) 형식으로 표현한 것이다.

[1] document : Language A | Language B | Language C
[2] Language A: {Element A`|Element B` }* | Language B | Language C
...
[3] Element A`: *attributes contents*
[4] attributes: Attribute A`` Attribute B``
[5] contents: {Element B` | Element C`}*
...
[6] Language B: {Element A` | Element D`}* | Language A | Language C

위 문법을 설명하자면, 라인 [1]은 파싱하고자 하는 하나의 문서는 여러 표준을 지원하는 언어들 중의 하나로 이루어 진다. 라인 [2]에서는 각 언어는 자신의 DTD를 기반으로 구성된 컨텐트 모델을 가진다. 동시에, 다른 언어를 내장할 수도 있다. 라인 [3]~[5]는 각 요소는 속성들과 자신의 컨텐트들을 가질 수 있다. 라인 [6]은 라인 [2]와 마찬가지로 다른 표준을 지원하는 컨텐트 역시 자신의 DTD를 기반으로 한 컨텐트 모델과 다른 언어를 내장할 수 있음을 나타낸다.

4. 결 론

본 논문의 요지는 현재 서비스 되고 있는 다양한 형태의 마크업 언어를 어떻게 하면 휴대단말에서 효율적으로 지원할 수 있는가 하는 것이다. 이를 위해서, 본 논문에서는 다양한 마크업 언어 스펙을 따르는 웹 브라우징 컨텐츠를 휴대 단말에서 파싱할 수 있는 시스템을 설계하였다

웹 브라우징을 위해 설계된 마크업 언어들은 각 단말이나 서비스 제공자에 의해 특화되어 있는 반면, 공통된 기능들을 많이 포함하며 대부분 XML 표준을 따른다. 본 논문에서는 이러한 특성들을 고려하여 각 언어들의 공통 분모 형태로 마크업을 파싱하도록 하였다. 또한, 각 단말에서 지원 가능하지 않는 마크업들은 무시하도록 하여 휴대단말에 적합하도록 설계하였다.

본 논문에서 설계한 파싱 시스템은 다음과 같은 특성을 가진다.

첫째, 각 휴대단말에 최적화시킬 수 있다.
둘째, 다중 마크업 언어의 파싱이 가능하다.
셋째, 다중 문서의 파싱이 가능하다.

본 논문의 파싱 시스템은 상기의 특성을 가지므로 기존의 다양한 마크업 언어를 지원할 수 있으며, 새로운 마크업 언어를 지원하는 것도 훨씬 쉬워지게 된다.

5. 참고문헌

[1] 최은정, 한동원, 임경식, "무선 인터넷 서비스를 위한 WAP 게이트웨이용 WML 컴파일러의 설계 및 구현," 한국정보과학회 논문지: 컴퓨팅의 실제, 제7권, 제2호, pp.165~182, 2001 년 4월.
[2] "Extensible Markup Language (XML) 1.0 Specification (Second Edition)", T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, 6 October 2000. http://www.w3.org/TR/REC-xml
[3] "Modularization of XHTML", M. Altheim et al., 10 April 2001. http://www.w3.org/TR/xhtml-modularization
[4] "Wireless Markup Language ," WAP Forum, 30-April-1998. http://www.wapforum.org/