XHTML 코드 생성기의 설계와 구현

계승철, 전서현
동국대학교 컴퓨터공학과

Design and Implementation of XHTML Code Generator

Seung-Chul Kye, Suh-Hyun Cheon
Dept. of Computer Engineering, Dongguk University
E-mail : ksch75@dongguk.edu, shcheon@dongguk.edu

요 약

XHTML은 HTML의 요소와 XML의 문법을 가진 마크업 언어이다. XML과 HTML의 장점을 결합하여 발표되었으며, HTML에서 XML로 가는 중간단계로, HTML을 대체할 언어로, 또는 유무선 통합을 위한 마크업 언어로 보고 있다. XHTML 언어를 이용하기 위해 텍스트나 기존에 널리 쓰이고 있는 HTML을 규칙에 맞는 HTML로 바꾸고, 간단한 조작으로 쉽게 XHTML로 바꿀 수 있도록 하는 XHTML 코드 생성기를 설계·구현하였다.

1. 서론

XHTML(eXtensible Hyper Text Markup Language)은 W3C(World Wide Web Consortium)에서 차세대 웹 문서의 표준인 XML(eXtensible Markup Language)과 네티즌의 공용어라 할 수 있는 HTML(Hyper Text Markup Language)의 장점만을 결합하여 발표한 마크업 언어이다[1]. XHTML도 SGML(Standard Generalized Markup Language)의 기반을 두고 설계되었다. SGML은 마크업을 규정하는 것이 복잡하여 실용적이지 못하고, HTML은 구조적이지 못한 문서구조를 포함함으로써 구조적인 데이터를 표현하기에 부적합하고, 문서구조의 확장이 불가능하며, XML은 대부분의 사람들이 사용하고 있는 웹 브라우저가 XML을 표현할 수 없다는 어려움이 있다. 이러한 단점들로 인해 XHTML이 등장했다. XHTML은 SGML의 복합구조와 일반화된 마크업등의 특성을 지원하고, XML의 문법을 따르기 때문에 규칙에 어긋나는 문서는 표현하지 않아도 되며, HTML의 요소들을 사용하므로 구식 HTML 브라우저에도 표현이 가능하다.

HTML은 마크업 구문이 느슨하여 규칙에 어긋나는 마크업 문서도 처리해 주어야 하기 때문에 브라우저는 잘못된 마크업을 원활하게 하기 위한 코드를 추가해야 했다. 최근의 브라우저들은 부적절한 마크업들을 최대한 그럴듯하게 처리하는데 많은 시간과 자원을 소비한다. XHTML은 XML의 규칙을 따르기 때문에 XHTML 브라우저는 규칙에 맞지 않는 표시하지 않는 것을 원칙으로 한다[2]. 시스템 자원이 풍부한 데스크탑 컴퓨터 등에서는 브라우저가 커지고 많은 시간과 자원을 소모하더라도 큰 불편 없이 사용할 수 있다. 이것은 잘못된 마크업을 견딜만한 처리능력을 가지지 못하는 새로운 사용자 에이전트(휴대폰, 핸드헬드 컴퓨터, PDA등등)들이 등장하면서 무선통신 등으로 웹을 표현하는데 XHTML이 데스크탑 컴퓨터에서의 HTML을 대신하는 데 있어서 효과적임을 의미한다.

XHTML의 호환성 때문에 WAP Forum에서 XHTML 표준을 WAP(Wireless Application Protocol) 2.0을 위한 기본 기술로 채택하기로 하였다[3]. XHTML을 WAP 2.0에 적용하기로 한 목적은 유무선 통합을 하기 위함에 있다. XHTML은 HTML 브라우저에서도, XML 브라우저에서도 마크업을 잘 표현해 주기 때문에 호환성과 유효성이 높은 편이며 무선통신장비(휴대폰, 핸드헬드 컴퓨터, PDA등등)에서 XHTML 브라우저가 탑재된다면 유무선 통합은 가능하다[4].

XHTML을 HTML에서 XML로 가는 중간과정으로 보거나, HTML을 대체할 새로운 마크업 언어로 보거나, 무선통신 쪽에서 말하는 유무선 통합으로 본다면, 지금까지의 인터넷에 존재해 왔던 HTML문서들은 XHTML문서로 대체되어야 한다. 하지만 현재의 인터넷에 존재하는 수많은 HTML문서들을 XHTML로 다시 작성하는 것은 많은 시간과 자원이 소모될 것이다.

본 논문은 HTML문서를 규칙에 맞는 HTML로 바꿔주고, HTML문서를 XHTML문서로 바꿔주는 변환기 겸 코드 생성기를 설계하고 구현하였다. 본 시스템은 마크업 언어의 계층구조를 적절히 사용함으로써 XHTML 코드 생성기의 환경에 적합한 구조정보를 유지 할 수 있었다.

논문의 구성은 2절에서는 XHTML에 대해 설명한다. 3절에서는 XHTML 코드 생성기의 설계와 구현을 설명한다. 마지막으로 4절에서 결론은 맺는다.

2. XHTML

XHTML은 HTML4를 재생성하고, 서브셋으로 확장하는 SGML 계열의 마크업 언어이다[5].

XHTML이 등장한 이유는 HTML의 한계 때문이다. HTML은 부적절하게 작성된 HTML 마크업들도 그럴듯하게 처리해준다. HTML 브라우저가 잘못된 마크업에 관대하고, HTML 마크업을 정보의 정의 수단이기보다는 문서 포맷을 위한 명령어로 간주했기 때문이다. 이는 전문가 이외의 보통사람들이 쉽게 웹 페이지를 만들 수 있어 대중화에 큰 역할을 했다. 하지만, 새로운 브라우저들이 나올 때마다 대충 만든 웹 페이지를 적절히 처리할 수 있는 기능이 강화되었으며, 이로 인해 브라우저의 크기가 점점 커지고, 많은 자원과 시간을 소비하게 되었다. 또, HTML은 비서술적 태그들로 마크업되어 있어서, 대부분의 웹 검색 엔진들은 웹 페이지를 문자열의 집합으로 간주하고 특정 키워드를 검색하는 비효율적 방식을 사용하고 있다. SGML의 기본 개념처럼 웹 페이지가 구조적 또는 양식적 태그들이 아닌 서술적 태그를 사용한다면 정보의 검색은 훨씬 더 쉬워지며, 새로운 종류의 웹 에이전트들이 더욱 더 많이 생겨날 것이며, 웹의 활용도도 지금과는 달라지게 될 것이다[1]. 자원이 풍부하지 못한 새로운 에이전트를 위해 또는 브라우저의 크기를 작게하고 빠른 처리를 위해, 브라우저에서 그럴듯하게 표현해주지 않더라도 정확하고 완벽하게 마크업 되었음을 보장할 수 있는 언어이며, 문서의 표현양식 이상을 표현할 수 있는 언어가 필요하다. 그 해결책으로 XML이 등장했지만, HTML은 매우 널리 쓰이고 있으며, XML은 HTML과 호환되지 않는다. 그래서, HTML을 다른 태그 집합들과 결합할 수 있는 작은 조각들의 집합으로 재정의하기 위해 XML의 한 응용물로 재정규화한 것이 XHTML이다.

XHTML 문서의 타입들은 XML에 기초하고, 궁극적으로 XML에 기초한 사용도구들과 연관하여 작동하도록 설계되었다. 이는 XML에 기초한 사용도구들과 HTML4의 규격에 맞는 사용도구들에 사용하도록 의도하였다는 것을 의미한다. 현재 사용중인 마크업 언어를 XHTML 언어로 변경하는 개발자들은 다음과 같은 이점들을 얻을 것이다.

- XHTML 문서들은 XML 규격에 맞는다. XML은 뛰어난 확장성을 가진 마크업 언어로 누구라도 새로운 태그를 만드는 것이 가능하다[6]. 이러한 특징을 XHTML도 가지고 있는 것이다. XHTML 문서들은 확장성이 뛰어나며, 표준 XML 도구들에서 쉽게 보여지고, 수정되며, 유효성이 점검된다.
- XHTML 문서들은 기존 HTML 규격에 맞는 사용도구들과 새로운 XHTML 규격에 맞는 사용도구들에서 작동 될 수 있다. 웹의 대부분을 차지하고 있는 HTML과 호환이 가능하다[7].
- XHTML 문서들은 Script와 applet등과 같은 HTML Document object나 DOM(Document Object Model)같은 XML Document object들을 활용 할 수 있다.
- XHTML이 발달함에 따라, 여러 XHTML 환경과 규격에 맞는 문서들이 웹 페이지를 만드는데 있어서 적용될 가능성이 높다.

WAP Forum에서 XHTML 표준을 WAP 2.0을 위한 기본 기술로 채택하기로 한 것은 기존의 WAP 1.0 표준이였던 WML(Wireless Markup Language)이 XHTML에 비해 호환성과 확장성이 떨어지고, 게이트웨이를 통해야 하기 때문에 상대적으로 느린 속도를 가지며 텍스트 위주의 웹페이지를 특징으로 하기 때문이다. XHTML은 HTML과 호환이 되기 때문에 유무선 통합을 하기에 적합한 마크업 언어이다. 그리고, 게이트 웨이를 거치지 않기 때문에 빠르며 이미 검증된 HTML을 기반으로 하므로 그래픽을 사용할 수 있다.

WML은 단말기마다 균일한 랜더링이 보장되지 않기 때문에 사이트 운영자가 지원할 단말기들을 이용

해 직접 테스트를 거쳐야 하는 부담이 존재했다. XHTML은 XML과 같이 Well Formed Document이어야 하고 DTD(Document Type Definition)를 준수하는 Valid Document이어야 한다. XHTML은 자체에서 제공하는 균일한 랜더링이 보장되기 때문에 테스트로 인한 부담이 경감되고 다양한 단말기의 지원이 용이하며, 문서의 불완전함을 체크하고 처리하는 불필요한 코드를 브라우저로부터 제거할 수 있어 무선 단말기의 작은 메모리를 효율적으로 사용할 수 있다. XHTML은 무선통신에서도 큰 이점을 가진다.

3. XHTML 코드 생성기의 설계 및 구현

3.1 시스템 구조

XHTML 코드 생성기의 구조는 [그림1]과 같다.

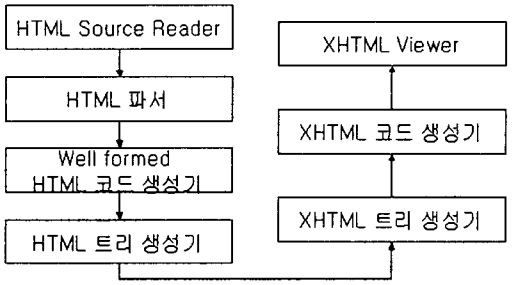

그림 1. XHTML 코드 생성기의 시스템구조

HTML Source Reader는 HTML 파일을 읽어들이는 역할을 한다. 텍스트 파일이나 HTML 파일을 읽어 화면에 출력한다.

HTML 파서는 HTML Source Reader의 결과를 입력으로 받아 읽어들인 HTML 코드를 태그와 텍스트로 분해한다. 각각의 태그와 태그 안의 텍스트를 잘라내어 Well formed HTML 코드 생성기의 입력으로 준다.

Well formed HTML 코드 생성기는 HTML 파서의 결과를 입력으로 받아 잘라낸 태그와 텍스트를 가지고 적격성(well-formedness)을 검사하여 well-formed한 코드로 변환하여 HTML 트리 생성기의 입력으로 준다.

HTML 트리 생성기는 Well formed HTML 코드 생성기의 결과인 변환된 HTML 코드를 가지고 HTML 트리를 생성한다. 그리고, 생성된 HTML 트리를 화면에 출력한다.

XHTML 트리 생성기는 생성된 HTML 트리를 입력으로 받아 XHTML 트리를 생성한다. 그리고 생성된 XHTML 트리를 화면에 출력한다.

XHTML 코드 생성기는 생성된 XHTML 트리를 입력으로 받아 XHTML 코드를 생성한다. 그리고 생성된 XHTML 코드를 XHTML Viewer의 입력으로 준다.

XHTML Viewer는 생성된 XHTML 코드를 화면에 출력한다.

3.2 Well formed HTML 코드 생성기

Well formed HTML 코드 생성기는 읽어들인 HTML Code를 HTML 파서가 잘라내면 그것을 입력으로 받아 적격성 검사를 하고 규칙에 맞는 HTML 코드를 생성하는 데 사용된다.

적격화된 코드를 생성하는 이유는 HTML 코드와 XHTML 코드가 유사한 부분이 많으며 XHTML 코드를 생성할 때 HTML 트리를 XHTML 트리로 변환시켜 효율적인 코드생성을 하기 위함이다.

우선 태그들이 Start Tag와 End Tag로 짝지어져 있는지 판단하여 짝지어지도록 태그를 만들어주는 일을 한다. XHTML요소는 HTML과는 달리 모든 태그들이 Start Tag와 End Tag가 반드시 짝을 이루어야 한다. 그 다음, 반드시 있어야 하는 태그(ex. <HTML>, <HEAD>, <BODY> 등)가 있는지 판단하여 없으면 역시 태그를 맞춰준다. 텍스트 파일을 열어서 Well formed HTML 코드 생성기의 입력으로 주면 <HTML>, <HEAD>, <BODY> 등의 태그를 생성하여 HTML코드로 바꿔준다.

Well formed HTML 코드 생성기는 크게 4개의 모듈로 구성되어 있다.

- StartTag 모듈 : 파싱된 HTML 코드를 분석하면서 Start Tag를 만나거나 적격화된 문서에 반드시 포함되어야 하는 태그가 없으면 실행되며 해당 태그를 Well formed HTML 코드에 포함시킨다.
- EndTag 모듈 : 파싱된 HTML 코드를 분석하면서 End Tag를 만나거나 적격화된 문서에 반드시 포함되어야 하는 태그가 없으면 실행되며 해당 태그를 Well formed HTML 코드에 포함시킨다.
- Text 모듈 : 파싱된 HTML 코드를 분석하면서 Text(태그 안의 내용)를 만나면 실행되며 해당 내용을 Well formed HTML 코드에 포함시킨다.
- ErrorTag 모듈 : 파싱된 HTML 코드를 분석하면

서 HTML 태그에 정의되지 않은 태그를 만나면 실행되며 사용자 정의 태그로 규정되어 Well formed HTML 코드에 포함시킨다.

태그가 전혀 없는 텍스트파일을 읽어 들여 Well formed HTML 코드 생성기의 입력으로 주어도 반드시 사용해야 하는 모든 태그들을 붙여 적격화된 HTML 코드로 바꿔준다. 이 모듈 하나만으로도 사용자가 입력한 텍스트를 HTML 코드로 변환하여 웹 브라우저에서 보여주거나, 일반적인 HTML 코드를 올바른 형식으로 쓰여진 HTML 코드로 변환할 때 사용할 수 있다.

구체적으로 모듈이 가지는 동작은 [그림2]와 같다.

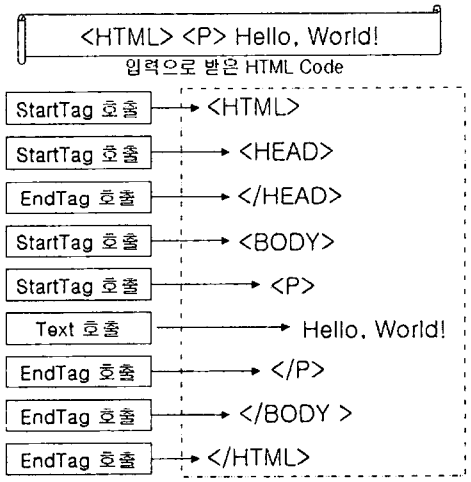

그림2. HTML코드의 적격화를 위한 모듈들의 동작

필요한 태그가 없는 HTML 파일은 각각의 모듈들이 호출되어 올바른 HTML 코드를 생성해 준다.

3.3 HTML 트리 생성기

HTML 트리 생성기는 적격화된 HTML 코드를 가지고 HTML 구조로 트리를 생성한다. Well formed HTML 코드 생성기의 결과인 변환된 HTML 코드를 가지고 HTML 트리를 생성한다.

파일을 읽어들일 때 HTML Source Reader와 HTML 파서, Well formed HTML 코드 생성기가 차례로 수행되어 바로 적격화된 HTML 코드를 만들어 준다. 이것을 받아서 트리로 생성하여 화면에 출력한다.

[그림3]은 [그림2]의 HTML코드를 입력으로 받아 적격화 시킨 HTML 코드를 HTML 트리 생성기가 트리로 생성하여 구조화시켜 출력했을 때의 트리 형태이다.

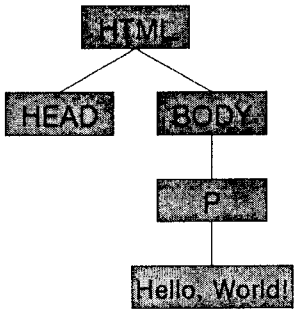

그림 3. 적격화 HTML의 트리

3.4 XHTML 트리 생성기

XHTML 트리 생성기는 Well formed HTML 코드를 바탕으로 만들어진 HTML 트리를 입력으로 받아 XHTML 트리를 생성하고, 생성된 XHTML 트리를 화면에 출력하는 역할을 한다.

태그 부분의 코드는 4가지 수정 사항을 가진다.

- XHTML요소는 모든 태그들이 Start Tag와 End Tag가 반드시 짝을 이루어야 한다. 즉, 종료태그를 생략할 수 없다.
- 태그들의 내포관계가 겹칠 수 없다.
- 태그와 속성은 모두 소문자로 구성되어야 하며, 대소문자를 구별한다.
- 해당 HTML의 태그들은 그에 맞는 XHTML 태그로 바꿔주고 HTML에서는 존재하지 않거나 필요없어도 되지만 XHTML에서는 필요한 태그나 요소들을 포함시킨다.
- 해당 속성의 속성값은 ""(double quote)를 이용하여 표기해야 하며 생략할 수 없다.

첫 번째와 두 번째의 수정사항은 적격화된 HTML 코드를 생성하면서 이미 해결되었다.

세 번째와 네 번째 수정사항을 해결하기 위해 Well formed HTML 코드 생성기에서 출력된 HTML 트리의 노드들을 탐색하면서 태그와 속성을 모두 소문자로 바꿔주고 XHTML에 맞는 태그와 요소들로 수정·추가해 준다.

다섯 번째 수정사항은 XHTML 코드 생성기에서 수정될 것이다.

XHTML 트리 생성기의 출력은 XHTML 코드 생성기의 입력으로 들어가 트리를 탐색하면서 XHTML 코드를 생성하게 된다.

3.5 XHTML 코드 생성기

XHTML 코드 생성기는 생성된 XHTML 트리를 입력으로 받아 XHTML 코드를 생성한다. 그리고 생성된 XHTML 코드를 XHTML Viewer의 입력으로 주는 역할을 한다.

[그림4]에서와 같이 XHTML 트리 생성기에서 출력된 XHTML 트리의 노드들을 탐색하면서 XHTML 선언을 해주고 해당 태그의 속성을 부여해 준다.

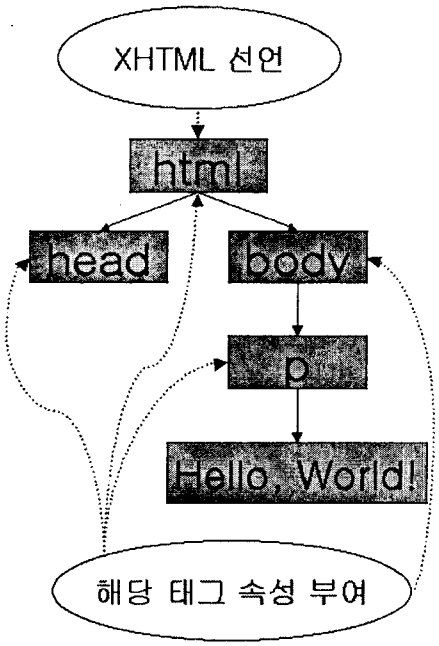

그림 4. XHTML 코드 생성

XHTML은 XML의 요소와 HTML의 요소를 모두 가지고 있으므로 <html>요소에 xml의 속성인 xmlns를 추가하고 XHTML 임을 나타내는 속성값을 준다. 그리고, XHTML을 정의하는 XHTML DTD를 선언한다.

XHTML은 사용언어를 표현하기 위한 언어정의 부분이나 글자 엔코딩 선언 부분이 HTML코드와 다르므로 그런 부분이 있으면 그에 알맞은 XHTML코드로 수정한다.

XHTML의 태그 부분에서 해당 속성의 속성값은 ""(double quote)를 이용하여 표기해야 하며 생략할 수 없기 때문에 HTML 코드에 속성 값이 생략되어 있는 코드가 있으면 속성 자체를 제외시키고, 속성 값이 있는 속성들은 ""를 이용하여 표기한다.

XHTML은 HTML과 스크립트 사용 방법이 다르므로 자바 스크립트등의 스크립트가 사용되어 있으면 XHTML의 형식에 맞게 변경하여 표기한다.

XHTML 코드 생성기의 출력은 XHTML Viewer의 입력으로 들어가 XHTML 코드가 출력되어야 할 탭에 출력되고 XHTML 코드를 보기 위한 탭을 활성화 시켜 XHTML 코드를 볼 수 있게 한다.

3.6 인터페이스의 설계

XHTML 코드 생성기의 인터페이스는 입력받은 HTML 코드와 그 코드를 적격화한 HTML 트리, 그 HTML 트리로 만든 XHTML 트리, 그리고 XHTML 코드를 보여주도록 구성되어 있다. HTML 코드를 읽어들이면 읽어들인 HTML 코드와 그 코드를 적격화한 HTML 트리가 생성되어 출력되고 XHTML 코드를 생성하도록 명령하면 HTML 트리 아래에 XHTML 트리가 출력되고 오른쪽에 XHTML 코드가 출력되도록 설계하였다.

[그림5]는 HTML 코드를 읽었을 때의 화면이다. 오른쪽 HTML Source 탭의 내용은 HTML 파일의 내용이다. 반드시 있어야 할 태그들이 없음을 알 수 있다. 왼쪽의 트리는 읽어들인 HTML 코드에 대한 계층구조를 한 눈에 볼 수 있는 트리이며, 없는 태그들을 포함시킨 것을 볼 수 있다.

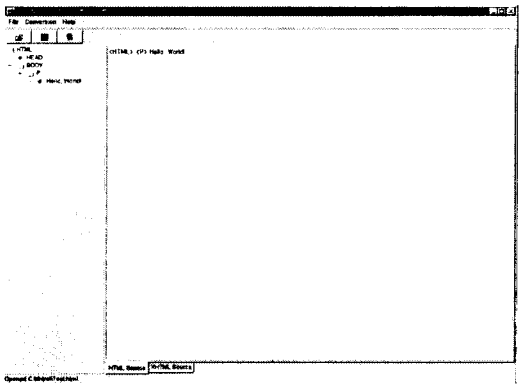

그림 5. HTML 코드 보기 화면

[그림6]은 XHTML 코드로 전환한 후의 화면이다. [그림5]의 상태에서 변환버튼을 누르면 HTML트리를 XHTML 트리로 만들어 왼쪽 아래에 배치해 코드의 계층구조를 보여준다. 그 트리를 탐색하면서 XHTML의 코드를 생성하고 XML요소의 선언과 XHTML DTD의 선언, 그밖에 HTML코드와 다른 내용들을 추가하여 XHTML코드를 생성한다. 생성된 XHTML코드는 탭이 자동으로 전환되면서 출력된다. 오른쪽의 XHTML Source 탭은 전환된 XHTML의 코드를 보여주고 있다. 이 코드를 XML파일이나 HTML파일로 저장하여 각각의 브라우저 또는 새로운 사용자 에이전트가 가지는 브라우저에서 사용할 수 있다.

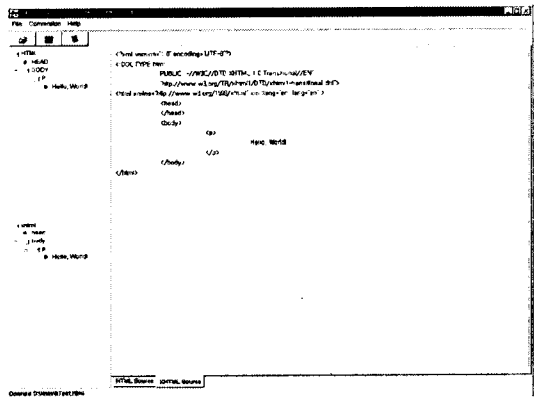

그림 6. XHTML 코드 출력 화면

4. 결론 및 향후연구

본 논문에서는 텍스트파일이나 HTML 코드를 HTML의 규칙과 문법에 맞게 고치고 그 코드를 이용하여 XHTML 코드를 생성하는 변환기 겸 코드 생성기를 설계하고 구현하였다. HTML과 XHTML의 계층구조를 가지는 트리구조와 XHTML 코드가 간단한 조작으로 자동생성 되었으며, 그로 인해 XML 브라우저나 HTML브라우저에서도 잘 표현되는 호환성 높은 코드가 쉽게 만들어질 수 있었다. 향후과제는 이 시스템에 HTML 브라우저와 XHTML 브라우저를 포함시키고 태그를 모르는 사람도 쉽게 편집할 수 있는 WYSIWYG방식의 편집기를 추가하는 것이다.

이 시스템의 XHTML 코드 생성 방식을 이용하여 정확한 표현을 가지는 마크업을 쉽게 이용할 수 있으며, 그로 인해 웹브라우저의 크기를 줄일 수 있고, 나아가 무선통신기기에 탑재되는 브라우저를 이용하여 유·무선 통합에도 적용될 수 있을 것이다.

[참고문헌]

[1] Frank Boumphrey, Cassandra Greer, Dave Raggett, Jenny Raggett, Sebastian Schnitzenbaumer, Ted Wugofski, "Beginning XHTML", wrox, 2000

[2] Michael C. Daconta, Al Saganich, "XML Development With Java 2", SAMS, 2000

[3] Nokia, "Advantages of XHTML for Wireless Data", http://www.nokia.com/press/background/pdf/mar011.pdf, 2001

[4] 권오성, "Xhtml based Integration between Wired Internet and Wireless Internet", Mosca Weekly Newletter, 29호, 2001

[5] W3C, "XHTML 1.0 : The Extensible HyperText Markup Language", http://www.w3.org/TR/xhtml1/, 2000

[6] Hiroshi Maruyama, Kent Tamura, Naohiko Uramoto, "XML and Java Developing Web Applications" Addison-Wesley, 1999

[7] Alexander Nakhimovsky, Tom Myers, "PROFESSIONAL Java XML Programming", wrox, 2000