

XML기반의 모바일 문서 변환 시스템에 관한 연구

김창수* · 정희경**

*청운대학교 인터넷학과 · **배재대학교 컴퓨터공학과

A Study of XML-based on Mobile Document Conversion System

Chang-su Kim* · Hoe-kyung Jung**

*Dept. of Internet, Chungwoon University · **Dept. of Computer Engineering, Paichai University

요 약

최근 모바일 인터넷기술과 인터넷의 발달로 다양한 통합된 서비스가 이루어지고 있다. 하지만 양질의 통합된 서비스를 제공하기 위해서는 많은 모바일 인터넷 콘텐츠가 개발되어야 한다. 기존의 인터넷 콘텐츠는 HTML 기반의 콘텐츠로 구성되어 있어 모바일 인터넷 기기의 제약으로 모바일 인터넷 환경에서 사용하는데 어려움이 있다.

이에 본 논문에서는 기존의 인터넷 콘텐츠와 모바일 인터넷에서 호환하여 유무선 콘텐츠간의 변환과 다양한 모바일 콘텐츠간의 변환을 할 수 있는 문서 변환 시스템을 설계하였다.

ABSTRACT

Recently, various integrated service comes true by a development of a mobile internet technology and Internet. However, much mobile internet contents should be developed to provide integrated service of good quality and does. Existing Internet contents consists of contents of a HTML base and uses in a mobile internet environment by restriction of mobile internet device, but there is a difficulty.

In this paper, we was compatible with existing Internet contents in mobile internet, and wire Internet and wireless Internet contents came , and a conversion was able and designed a XML-based on mobile document conversion system that it was possible with various mobile contents.

키워드

Mobile Internet, XML, XSLT, Document Conversion System

I 서론

최근 정보화 사회로 변화하면서 인터넷의 빠른 발전으로 인터넷을 통한 다양한 콘텐츠 및 데이터 교환의 중요성이 증가되고 있다. 또한 무선 통신기술과 인터넷의 통합으로 사용자는 다양한 무선 디바이스를 사용하여 시간과 장소에 제한 없이 인터넷 서비스를 이용할 수 있는 무선 인터넷 서비스도 빠르게 발전하고 있다.

하지만 무선 인터넷 서비스를 이용하는데 있어 기존의 인터넷 콘텐츠를 제대로 활용하는데 어려움이 있다. 양질의 통합된 서비스를 제공하기 위해서는 많은 모바일 인터넷 콘텐츠가 개발되어야 한다. 기존의 인터넷 콘텐츠는 HTML 기반의 콘텐츠로 구성되어 있어 모바일 인터넷 기기의 제약으로 모바일 인터넷 환경에서 사용하는데 어려움이 있다.

이에 본 논문에서는 다양한 무선 디바이스에

서 정보접근이 가능한 기존의 인터넷 콘텐츠와 모바일 인터넷에서 호환하여 유무선 콘텐츠간의 변환과 다양한 모바일 콘텐츠간의 변환을 할 수 있는 문서 변환 시스템을 설계하였다.

II. XSLT

XSLT는 XML 문서를 표현하기 위해 사용하는 언어로서 구조적 변환 단계와 XSL 포매팅의 두 가지 부분으로 구성되어 있다. XSLT가 XSL 포매팅을 이용하여 XML 문서를 다른 문서 형식으로 변환하는 문제를 해결하도록 고안되었지만 서로 다른 구조를 같은 독립적인 XML 문서로의 변환에도 사용할 수 있다[1,2,3].

이러한 XSLT는 원본 문서의 구조를 결과 트리로 변환하는 패턴과 서식이 결합된 규칙으로 설명하고 있다. XSLT는 서식과 함께 구조화하며, 서식은 문자열의 조합으로 된 형식으로 문서

의 서식 요소를 추가한다. 또한 문서의 구조 요소를 새로이 생성하여 지정할 수도 있다. 이러한 서식은 각각 패턴으로 탐색된 XML 구조 요소를 선택하고 대응 요소를 서식 규칙으로 처리하여 그 결과를 문서로 생성한다. 결국 패턴 생성 결과로 XML 구조의 요소를 선택하고, 서식 내의 포맷 처리 언어로 선택된 구조 요소를 XSLT 처리기로 보내어 문서를 정의된 형식으로 변환한다. 적용하려는 서식을 검색하는 방법에서 주어진 XML 구조 요소에 대응하는 패턴을 갖는 서식은 다양하게 존재할 수 있지만 XSLT 처리기에서 처리되는 서식은 한 서식만 적용된다. 그리고 서식은 XML 이름공간(XML Namespace)에 지정된 영역 집합을 가지는데 각 접두어 'xsl' 을 사용하여 XSLT 서식 요소를 사용할 수 있으며 XSLT 서식 요소는 크게 다음의 세가지로 나눌 수 있다[4, 5].

- 기본 제어 요소 : XML 문서의 내용을 처리하기 위한 XSLT 서식 요소들
- 수정 및 처리 요소 : XML 문서의 구조 요소를 수정하고 처리하는 기능을 제공하는 요소들
- 의사결정 요소 : XSLT 문서에서 의사결정 요소들

III. 시스템 설계

XML 기반의 모바일 문서변환 시스템은 HTML, XML, XSLT 문서를 입력 받아 각 문서의 구조를 생성하고 무선 인터넷 콘텐츠를 위한 표현 및 변환 정보를 정의하는 문서변환 처리 엔진과 문서에 대한 구조 편집 및 디스플레이 편집을 위한 렌더링 엔진, XML에 XSLT를 적용하여 모바일 콘텐츠형식으로 변환하는 모바일 콘텐츠 변환기로 구성된다. 그림 1은 전체 시스템 구성도 이다.

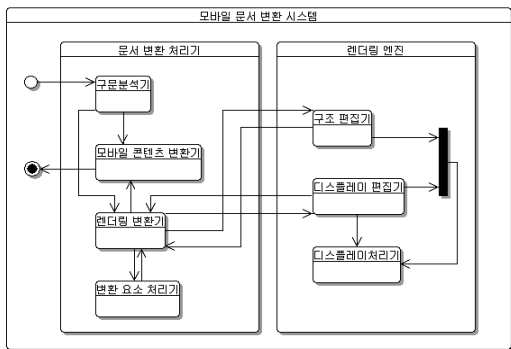


그림 1 전체 시스템 구성도

3.1 문서 변환 처리기

문서 변환 처리기는 입력 문서의 구문 정보처리를 위한 구문 분석기, 입력 문서를 모바일 콘텐츠 표현을 할 수 있도록 문서로 변환하는 모

바일 콘텐츠 변환기, 문서구조 정보를 변환하는 렌더링 변환기, XSLT 요소 처리를 위한 변환 요소 처리기로 구성된다.

3.1.1 모바일 콘텐츠 변환기 설계

모바일 콘텐츠 변환기는 원본 문서인 HTML, XML 문서와 결과 문서인 XSLT 문서를 입력받아 모바일 콘텐츠 HTML 문서로 변환한다. 그림 2는 모바일 콘텐츠 변환기에서 처리되는 흐름을 보여준다.

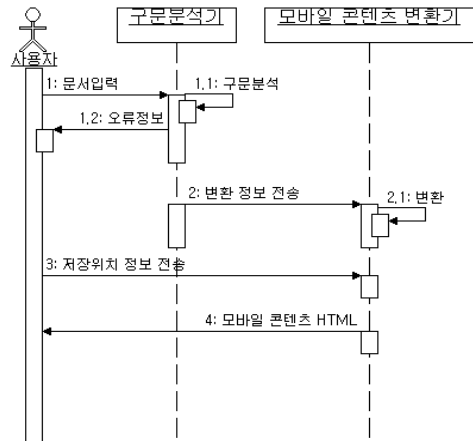


그림 2 모바일 콘텐츠 변환기의 순차 흐름도

바일 콘텐츠 변환기는 구문 분석기에서 분석된 변환 정보를 입력 받아 변환 모듈을 생성하여 사용자에게 지정받은 저장 위치에 변환된 모바일 콘텐츠 HTML 문서를 저장시키게 된다.

모바일 콘텐츠 HTML 문서로의 변환을 위한 파라미터로 원본 문서인 XML 파일과 결과 문서인 XSLT 파일 그리고 변환된 후 생성될 모바일 콘텐츠 HTML 파일이 입력된다. 모바일 콘텐츠 HTML 문서로의 변환을 위한 준비 단계로써 TransformerFactory 인스턴스를 생성한다. 그다음으로 실제 변환을 수행하게 되는 Transformer 인스턴스를 생성하여 파라미터로 받아온 XSLT 파일을 읽고, SAXSource 인스턴스를 생성하여 XML 문서를 읽어 들이는 것으로 변환을 위한 전체적인 준비 단계를 끝내게 된다. 마지막으로 Transformer 인스턴스의 멤버인 transform 메소드가 실행되어 모바일 콘텐츠 HTML 문서로 변환되도록 설계하였다.

3.1.2 렌더링 변환기

노드 정보 관리기는 DOM 객체를 이용하여 XSLT 문서의 루트 요소 및 자식요소까지 탐색하면서 트리 구조를 생성하며, 트리를 구성하는 각 노드에 대한 정보를 관리한다. 이렇게 생성된 트리 구조는 렌더링 엔진 내부의 구조 편집기의 입력 데이터가 된다. 그리고 XSLT 문서의 xsl:stylesheet 요소를 루트 요소로 하여 XSLT 문

서의 구조 요소를 패턴으로 정의한 결과를 표현하는 xsl:template 요소들 중에서 HTML 문서의 최상위 노드인 html 요소를 포함하고 있는 xsl:template 요소를 검색한다. 이렇게 검색된 최상위 xsl:template를 기반으로 xsl:template 내부에 패턴 결과를 적용하기 위해 관련 서식을 탐색하여 처리하는 요소인 xsl:apply-template 요소의 match 속성 값(value)을 기반으로 관련 서식을 탐색한 후, 임시 구조를 생성함으로써 모바일 콘텐츠 HTML 문서의 구조를 생성한다.

HTML 요소 추출기는 렌더링 변환기에서 생성한 임시 구조를 기반으로 모바일 콘텐츠 HTML 요소를 구조에 맞게 추출하여 렌더링 엔진 내부의 디스플레이 편집기에서 사용할 수 있는 입력 데이터를 생성한다.

3.1.3 변환 요소 처리기

렌더링 변환기에서 보내어진 XSLT 요소를 처리하기 위한 변환 요소 처리기는 각 XSLT 요소에 대한 특징적인 처리를 한다. 기본 제어 요소, 수정 및 처리요소, 의사 결정 요소 그리고 확장 제어 요소 등 4개의 요소 처리기와 각 요소 처리기에서 보내지는 속성 처리를 위한 속성 처리기로 구성된다. 그림 3은 변환 요소 처리기의 처리 흐름도 이다.

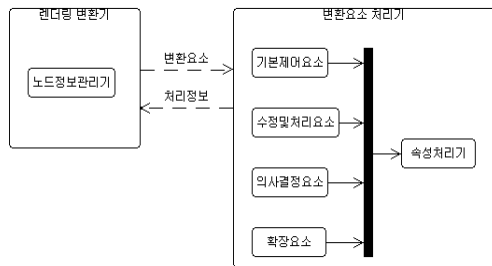


그림 3 변환 요소 처리기의 처리 흐름도

기본 제어 요소는 원본 문서의 내용을 처리하기 위한 XSLT 기본 제어 요소인 xsl:stylesheet, xsl:template, xsl:define-template -set, xsl:apply-templates, xsl:copy, xsl:value -of 요소들에 대한 처리를 수행한다. 특히 선택된 원본 문서의 요소를 패턴으로 정의하는 서식 요소인 xsl:template 요소와 패턴 결과를 적용하기 위해 관련 서식을 탐색하는 xsl:apply-template 요소 그리고 패턴의 처리 결과 값을 표현하는 xsl:value 요소는 XSLT 요소 가운데 가장 빈번히 사용되는 요소으로써 XSLT 문서의 구조를 이루는 근간이 되는 요소들이다.

수정 및 처리 요소는 xsl:element, xsl:attribute, xsl:attribute -set, xsl:comment, xsl:processing-instruction 요소들의 기능에 대한 처리를 하며, 각 요소들은 변환되는 결과 문서에서 요소 및 속성, 주석 등으로 생성된다.

의사 결정 요소는 일반 프로그래밍 언어들에

서 사용되는 조건, 반복 등의 의사결정 기능을 수행하는 요소들에 대한 처리를 하며, 의사 결정 요소들은 xsl:foreach, xsl:if, xsl:choose, xsl:when, xsl:otherwise 등이 있다.

확장 제어 요소는 위의 기본 요소들에 속하지 않는 요소인 xsl:output, xsl:import, xsl:include, xsl:sort 등의 요소들을 포함하고 있으며, 그에 대한 처리를 수행한다. xsl:output 요소는 변환되는 문서에 대한 정보를 설정하는 요소이며, xsl:import 요소와 xsl:include 요소는 다른 XSLT 문서를 포함하기 위한 요소들이다. 그리고 xsl:sort는 선택된 원본 문서의 요소들을 정렬하는 방법을 설정하는 요소이다.

또한, 4개의 요소 처리기에서 처리되는 요소의 속성들을 위한 속성 처리기를 따로 두었다. 요소 처리기와 속성 처리기를 분리함으로써 각 요소들이 갖는 중복되는 속성을 단일화 하여 처리할 수 있도록 설계하였다.

3.2 렌더링 엔진

렌더링 엔진은 HTML 요소를 표현하는 디스플레이 편집기와 문서구조를 처리하는 구조편집기로 구성된다.

3.2.1 디스플레이 편집기

디스플레이 편집기에서 생성되는 텍스트, 테이블, 이미지 등의 요소들은 HTML 4.0 명세에 정의된 요소와 속성을 따르도록 설계하였다. 그림 4는 디스플레이 편집기에서 처리되는 흐름을 보여준다.

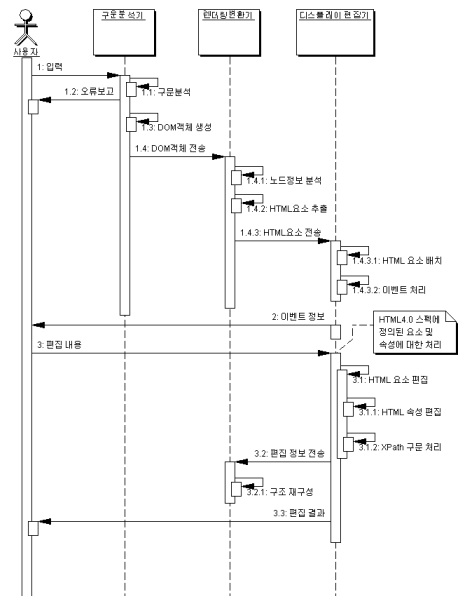


그림 4 디스플레이 편집기의 순차 흐름도

입력된 XSLT 문서를 구문 분석기에서 분석하여 생성되는 DOM 객체를 렌더링 변환기로 보내어 노드 정보를 분석하게 되고, HTML 요소를

추출하기 위하여 XSLT 문서의 루트 요소를 검색하고 xsl:template 요소 가운데 html 요소를 포함하는 최상위 xsl:template를 탐색하여 HTML 요소를 추출하기 위한 임시 구조를 생성한다.

이렇게 추출된 HTML 요소를 디스플레이 편집기에서 HTML 문서의 순서에 맞추어 요소를 배치하고 사용자의 편집 요구에 따라 HTML 요소에 대한 편집과 HTML 속성에 대한 편집을 수행한다. 그리고 원본 문서의 특정 노드를 가리키는 XPath 구문을 처리한 값을 삽입할 수 있다. 편집된 정보는 렌더링 변환기에 보내져 구조를 재구성하고, 디스플레이 편집기에서 재구성된 구조를 다시 배치하여 사용자에게 보이도록 한다.

3.2.2 구조 편집기

구조 편집기는 문서의 구조를 보면서 편집을 할 수 있도록 트리 구조 형태로 보여준다. 이렇게 구조 편집기에서 처리되는 정보는 요소와 속성에 대한 편집과 XPath 구문의 삽입에 대해 처리한다. 그림 5는 구조 편집기에서 처리되는 흐름을 보여준다.

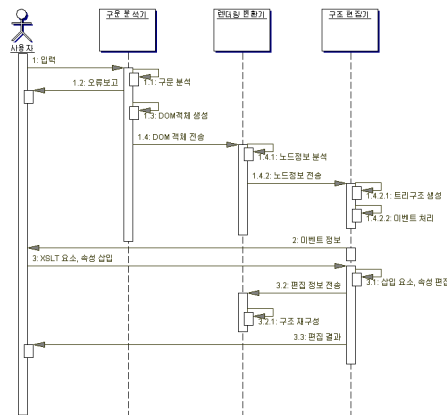


그림 5 구조 편집기의 순차 흐름도

구분 분석기에서 생성된 DOM 객체를 렌더링 변환기에서 노드 정보에 대한 분석을 하며, 분석된 노드 정보를 구조 편집기로 전송한다. 이 노드 정보를 구조편집기에서 트리 구조로 보여준다. 트리 구조로 표현된 노드들은 요소, 속성, 주석이며, 트리 구조에서 각 노드들에 대한 편집은 선택한 노드에 따른 마우스 이벤트를 통해 처리한다. 선택된 노드에 대한 적합한 편집 처리 이후에 렌더링 변환기로 편집 정보를 전송하여 노드의 정보를 갱신하며, 갱신된 노드 정보를 통하여 다시 구조 편집기에서 트리 구조로 보여진다.

트리 구조의 생성은 구분 분석기에서 생성된 DOM 객체에서 자식 요소 목록의 루트 노드를 검색하여 요소 처리 메소드를 호출하여 트리 구조의 최상위 노드에 삽입한다. 요소 처리 메소드는 현재 요소 노드를 트리 구조에 삽입하고, 현재 노드가 갖는 속성을 탐색하여 하위 자식 노드로 삽입시킨다. 그리고 현재 요소 노드의 하위

자식 요소의 유무를 검사하여 최하위 자식 노드 위치까지 탐색을 하게 되면 형제 요소 노드를 검사한다.

이렇게 형제 노드를 트리 구조에 삽입하고 자식 노드의 유무를 검사하고 다시 현재 노드에서 검색된 자식 노드로 위치를 옮기는 반복적으로 재귀 호출(recursion)되는 방법으로 트리 구조를 생성하도록 설계하였다.

IV 결론

최근 모바일 인터넷기술과 인터넷의 발달로 다양한 통합된 서비스가 이루어지고 있다. 하지만 양질의 통합된 서비스를 제공하기 위해서는 많은 모바일 인터넷 콘텐츠가 개발되어야 한다. 기존의 인터넷 콘텐츠는 HTML 기반의 콘텐츠로 구성되어 있어 모바일 인터넷 기기의 제약으로 모바일 인터넷 환경에서 사용하는데 어려움이 있다.

유무선 인터넷 콘텐츠간의 정보 전송이나 변환에서는 문서의 매핑 정보를 이용하거나 불필요 요소를 제거하는 필터링 방법을 사용한 변환을 제공하기 때문에 변환의 정확성이나 다른 서비스 제공하는데 있어 많은 제약이 있다. 이에 본 논문에서는 변환 모듈을 XSLT 기반으로 설계하여 무선 디바이스에서 정보접근이 가능하고 기존의 인터넷 콘텐츠와 모바일 인터넷에서 호환하여 유무선 콘텐츠간의 변환과 다양한 모바일 콘텐츠간의 변환을 할 수 있어 확장성 및 다양한 변환을 할 수 있는 XML 기반의 모바일 문서 변환 시스템을 설계하였다.

향후에는 WAP, ME기반의 보다 다양한 모바일 콘텐츠를 표현할 수 있고 모바일 콘텐츠의 데이터를 고려한 객체표현 기술을 적용한 시스템을 구현해야 할 것이다.

참고문헌

- [1] W3C, Extensible Markup Language (XML) Version 1.0, <http://www.w3.org/TR/REC-xml>, Oct. 6, 2000
- [2] W3C, XSL Transformations (XSLT) Version 1.0, <http://www.w3.org/TR/xslt>, Nov. 16, 1999
- [3] Eric M. Burke, "Java and XSLT", O' Reilly, 2001
- [4] S.Saba, "Bringing the Wireless Internet to Mobile Devices", Computer 2001 IEEE, pp.54-58, 2001
- [4] 성길용, "데이터 매핑을 이용한 XML 구조 변환 시스템", 배재대학교, 2001
- [5] H.Ouahid, A. Karmouch, "Converting Web Pages into Well-formed XML Document", IEEE, pp.676-690, 1999