객체모델링 기법을 이용한 XSLT 처리기의 설계 및 구현

인경숙*, 하얀**, 이경환*
* 중앙대학교 컴퓨터공학과
** 경인여자대학 멀티미디어정보전산학부

Design and Implement of an XSLT Processor Using Object -Oriented Modeling Method

Kyung-Sook In*, Yan Ha**, Kyung-Whan Lee*
* Dept. of Computer Engineering, Chung Ang University
** Div. of Multimedia Information Computer Science, Kyungin Women's College

요 약

본 XSLT 처리기는 객체모델링 기법인 UML을 기반으로 설계하였고 핵심적인 부분은 XSL 스타일 시트를 HTML 패스/스타일로부터 구성해 내는 것이다. 즉 문서 작성자는 XML문서를 작성하고 파싱하여 패스 테이블을 형성하고 이 패스 테이블에 작성자는 HTML 패스/스타일을 추가한다. 이를 통해 XSL의 복잡한 스타일에 대한 고려없이 문서구조와 정보에 집중하여 견고한 문서를 만들 수 있다.

즉 구조 문서인 XML과 콘텐츠 타입의 대중적인 문서형인 HTML을 이용함으로써 적은 비용으로 다양한 문서 스타일을 구성할 수 있게 하여 XML 기술을 쉽게 보급시키며 컴포넌트 제작을 통해 다른 산업영역 문서교환에서 활용할 수 있다.

1. 서론

XSLT는 XML 문서를 다른 문서 형태로 변환하거나 워드와 같은 스타일을 지원하는 데 필요한 기술이다. 본 논문에서는 객체 지향 모델링기법을 이용하여 XSLT 처리기를 구현하기 위한 기술환경을 연구하고 응용방법을 제안하고자 한다.

즉 문서 처리 시스템을 구성할 때 XSLT 처리기의 핵심적인 부분인 XML 문서에서부터 HTML로 작성된 패스/스타일 문서를 구성해 내는 것이다. 구조 문서인 XML과 콘텐츠 타입의 대중적인 문서형인 HTML을 이용함으로써 적은 비용으로 다양한 문서 스타일을 구성할 수 있도록 XSLT처리기를 객체모델링 방법을 이용하여 만드는 것을 목적으로 한다.

이를 위하여 XSLT 처리기 내 템플리트 병합기를 두어 문서 스타일을 포함하는 방법으로 실질적인 문서의 디자인은 HTML 패스/스타일로 실시하게 하고 XML 문서의 트리로부터 입력 받은 분석기가 문서의 종류를 파악하고 해당 문서의 필수적인 엘리먼트의 변환은 XSLT처리기내의 템플리트 병합기가 하게 하였다.

2. 관련연구

2.1 XSL와 XSLT처리기

스타일이란 문서의 여러 가지 표현 방법과 효과를 지정하는 것으로 패턴(Pattern)들을 관련시킨다.

XML문서에 대한 사양이 발표된 후에 XML 문서를 보여주기 위하여 문서에 대한 포맷을 정의하는 스타일 시트로 XSL(eXtensible Style Language)이 제안되었다.

XSL에서 문서에 스타일을 적용하기 위한 포맷팅 객체와 문서변환을 위한 트랜스 폼(Transform) 기능을 지원하는데 이를 특별히 XSLT(XML Stylesheet Language Transformation)라고 부른다.

1999년 11월 W3C에서 보편적 사용안으로 제안한 XSLT 1.0표준안은 XML문서들을 필요에 따라 XML 태그명과 트리구조를 변경시키면서, 여러 포맷으로 변환시키는 표준방법을 규정하고 있다.

XSLT가 XML문서를 다른 문서로 변환하는 방법은 두 단계를 거쳐서 이루어 지는데 먼저 문서를 원하는 출력 구조로 변환하고 변환된 출력 구조 문서를 필요한 문서 형식으로 포맷팅을 한다.

XSLT에서 표현된 변환 명세는 원시 트리(Source tree)에서 결과 트리(Result tree)로 변환하기 위한 규칙을 설명하고, 변환은 템플리트(Template)들과 패턴(Pattern)은 원시 트리에 있는 엘리먼트에 일치(Match)시키고, 템플리트(Template)는 결과 트리의 부분을 새로 만들기 위한 것이다. XSLT상에서 표현된 변환을 스타일 시트라고 하며 이 스타일 시트는

템플리트 규칙의 집합을 포함한다.[1]

2.2 UML

소프트웨어 개발 현장에 소프트웨어 위기론이 대두 된 이후로 80년대 후반과 90년대 초반에 다수의 객체지향 분석, 설계기법의 방법론이 소개되었다. 그 중 대표적인 방법 중 Grady Booch, James Rumbaugh, Ivar Jacobson이 의해 고안된 모델링 언어인 UML(Unified Modeling Language)을 표준화 기구인 OMG에서 97년 11월 표준으로 제정했다.

UML의 목적은 객체 지향 시스템을 가시화하며 명세화하고 문서화하는 것이다. UML에는 각종 다이어그램을 포함하는데 대표적인 것으로 순차, 협력, 클래스 다이어그램이 있다.

3. XSLT처리기의 설계

XSLT 처리기의 가장 중요한 템플리트 병합기의 시스템 설계, 시스템 구현 등 일련의 과정은 객체지향 모델링 기법인 UML을 사용하였으며 모델링 도구로써 Rational Rose 2000 Enterprise Edition을 사용하였다.

3.1 XSLT 처리기의 구조

본 시스템은 XML문서를 파싱하여 이를 통해 얻은 트리로부터 모든 경우에 대해 HTML로 작성된 패스/스타일 문서를 만들어서 XSL 문서를 생성해 내고 이를 통해서 XSLT 문서를 HTML 뷰어에서 볼 수 있도록 하였다.

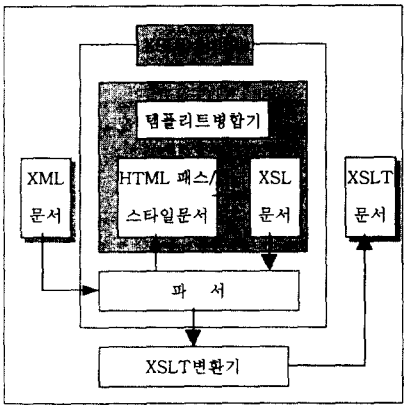

그림 1 XSLT 처리기의 구조

3.2 XSLT 처리기의 컴포넌트 설계

XSLT 처리기는 XML문서로부터 XSLT 문서를 생성해 내는 과정을 설계한다. 본 XSLT 처리기는 전체적으로 두 개의 인터페이스와 네 개의 클래스로 되어있다.
Rational Rose는 논리적 뷰에서 후보 클래스를 선정하며 이를 통해서 클래스 다이어그램을 생성하고 유스케이스 뷰를 통해서 순차 다이어그램과 협력 다이어그램을 생성하고 컴포넌트 뷰에서는 컴포넌트 다이어그램을 설계하였으며 전 과정은 상호 연관성이 있다.

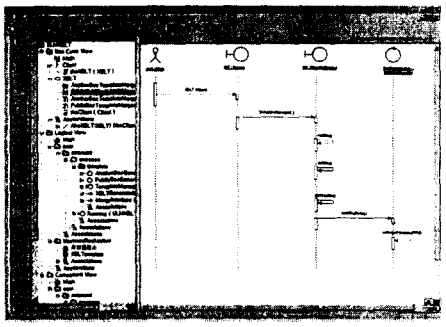

그림 2 XSLT처리기 순차다이어그램

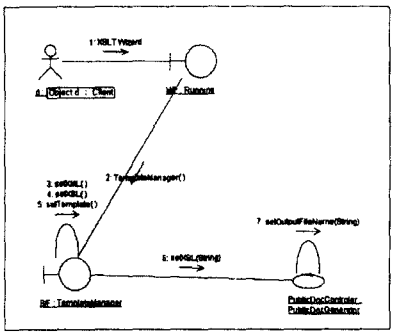

그림 3 XSLT처리기 협력다이어그램

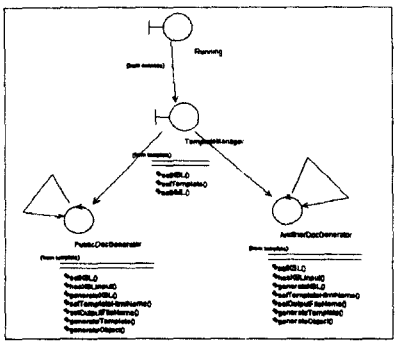

그림 4 XSLT처리기 클래스다이어그램

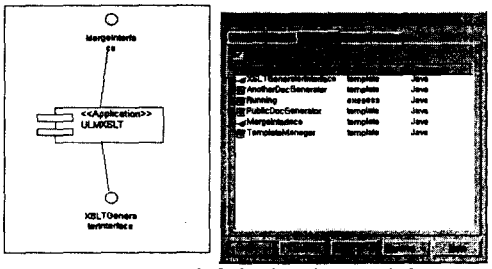

그림 5 XSLT처리기 컴포넌트/스펙창

표 1 XSLT처리기의 주요기능

인터페이스	**XSLTGeneraterInterface** XML문서와·템플리트 병합기를 통해서 나온 XSL 문서를 가지고 XSLT 문서를 생성한다
	MergerInterface XML문서를 입력 받아 문서의 종류를 파악하고 HTML패스/스타일을 지정하여 XSL문서를 생성한다
클래스	**Running** : 사용자 인터페이스이다
	TemplateManager Running의 의해서 시작되며 문서의 종류에 따라 PublicDocGenerator와 AnotherDocGenerator 클래스로 간다
	PublicDocGenerator 특별히 공문서인 경우 대부분의 문서가 스타일이 동일하므로 자체적으로 일부 XSL스타일 문서를 내장하고 있고 특정 경우에는 HTML패스/스타일을 입력받아 XSL문서를 생성한다
	AnotherDocGenerator 기타 다른 형식의 문서인 경우를 의미하며 PublicDocGenerator와 동일하다

3.3 템플리트 병합기의 시나리오

템플리트 병합기는 XSLT 문서를 만들기 위한 XSL의 템플리트를 XML 문서로부터 사용자가 HTML 패스/스타일 문서를 설계할 수 있게 해주고 이를 통해 XSL 스타일 시트로 변경하는 처리기이다.

본 템플리트 병합기는 크게 두 부분으로 나뉘어 졌다.

첫번째 부분은 템플리트 병합기에서 먼저 XML문서를 파싱하여 트리를 생성한다. 이때 생성된 트리를 바탕으로 각각의 노드를 루트에서 단노드까지 모든 경우의 수를 찾아서 각각의 패스를 자료구조에 저장시켜 놓는다.

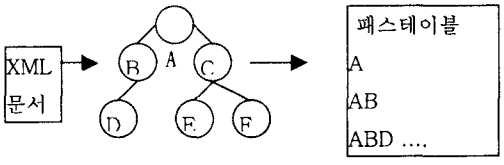

그림 6 템플리트 병합기의 패스 테이블 생성과정

다음은 패스 테이블을 생성하는 시나리오이다.

1. XML 파일을 선택한다.
2. 파서가 파싱한다.
3. 트리가 생성된다. // 파서가 수행해 준다
4. 트리에서 DOM Document를 얻는다. // DOM Document는 트리의 Root 노드를 의미한다
5. 깊이우선 탐색으로 가능한 경로의 경우의 수를 만들어낸다. // 모든 경우의 수는 노드가 엘리먼트 노드일 경우에만 의미가 있다.

두번째 부분은 발생된 패스 테이블의 경로에 대하여 HTML로 패스/스타일 테이블 문서를 지정하여, 사용자가 선택한 패스/스타일을 가지고 XSLT 문서를 생성하게 된다. 즉 파싱한 XML 소스 트리에서 사용자가 의미 있는 패스를 선택할 수 있도록 전체 패스를 테이블 형태로 제공하게 되고, 사용자는 이 테이블에서 의미 있는 패스에 대하여 스타일을 지정하게 된다.

3.4 구현

본 논문에서 개발한 시스템의 구현환경은 IBM 호환 PC로 운영체제는 Windows NT 2000 환경에서 개발하였으며, 구현 언어는 JDK1.3을 사용하여 개발하였다.

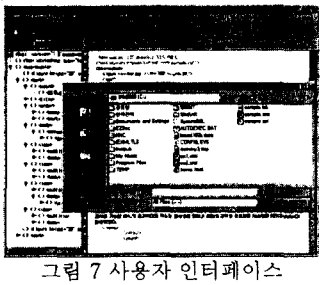

그림 7 사용자 인터페이스

4. 결론 및 향후 연구과제

기존 시스템이 복잡한 입력을 전제로 하고 있는데 하나의 XML 문서로부터 XSLT 문서를 만들기 위하여 XML 문서, DTD, XSL 문서 입력이 이루어 진다. 이는 일반 사용자가 XML 문서 처리에 대한 접근을 어렵게 하였다.

이를 개선하고자 본 시스템에서는 사용자 측면에서 어려운 XSL 문서를 XML 문서에서 HTML 패스/스타일로 대치 하였고 DTD와 일부 정형화된 XSL문서를 내장하여 처리하게 함으로써 사용자가 XSL에 대한 사전지식이 없어도 XSLT 문서를 생성할 수 있게 하였다. 기존에 HTML 태그에 대한 지식이 있으면 쉽게 접근할 수 있도록 한다.

즉, 본 템플리트 병합기에서 XML 문서로부터 패스테이블을 생성해 내고 이를 바탕으로 스타일을 규정하는 시나리오를 제안하였다.

앞으로, 본 논문의 템플리트 병합기 시나리오에서 제시된 모든 경우의 수를 포함한 패스 테이블에서 스타일 지정에 의미가 있는 엘리먼트를 추출하는 방법이 더 연구 되어져야 할 것이다.

5. 참고문헌

1. W3C, Extensible Stylesheet Language (XSL) Version 1.0 http://www.w3.org/Style/XSL , 18 October 2000
2. 박진우 " XSL패턴을 이용한 XSL문서 편집시스템 설계 및 구현" , 배재대학교 컴퓨터공학과 대학원 석사논문, 1999
3. W3C, XSL Transformations (XSLT) Version 1.0 http://www.w3.org/TR/xslt, 16 November 1999
4. W3C, Namespaces in XML http://www.w3.org/TR/REC-xml-names/,14 January 1999
5. Document Object Model (DOM) Level 1 Specification Version 1.0 http://www.w3.org/TR/REC-DOM-Level-1/, *1 October, 1998*
6. Grady Booch " Visual Modeling With Rational Rose 2000 and UML" 도서출판 인터비젼, 서기원 · 김효수 공역, 2000