DSSSL을 이용한 SGML 문서의 처리에 관한 연구

장은영 °, 이경호, 최윤철
연세대학교 컴퓨터과학과

A Study on the Processing of SGML Documents based on DSSSL

EunYeung Chang, KyongHo Lee, YoonChul Choy
Department of Computer Science, Yonsei University

요 약

본 연구에서는 SGML 문서의 포맷팅과 변환을 지원하는 DSSSL 처리기를 개발하였다. 본 논문에서는 DSSSL의 문서 처리 과정과 이를 지원하는 처리기의 개발 방법을 소개한다. 또한 개발 결과를 다양한 문서처리 환경에 적용하여 DSSSL의 우수성과 개선이 요구되는 부분에 대하여 기술한다. 그 결과 DSSSL은 구조 기반 검색을 지원하는 질의 언어와 다양한 연산 기능을 지원하는 수식 언어에 기반하기 때문에 강력한 포맷팅과 변환 기능을 제공한다. 그러나 처리 방식의 특성상, DSSSL은 사용자 인터랙션이 많으며 위지윅한 문서 환경보다는 일괄처리 방식의 문서 처리 분야에 더 적합하다.

1. 서론

SGML(Standard Generalized Markup Language)[1]은 이기종간의 호환이 가능하며 논리적인 구조 정보를 포함한다는 장점 때문에 다양한 분야에서 널리 사용되고 있다[2,3,4,5,6,7]. 이에 SGML 문서의 변환 및 포맷팅 등의 처리에 대한 관심이 증가하고 있으며 이에 관한 연구가 진행 중이다. 기존의 연구 결과는 시스템에 의존적인 방법을 적용한다[8]. 그러나 이는 특정한 플랫폼에 독립적이라는 SGML의 취지에 어긋나는 것이다. 이에 DSSSL(Document Style Semantics and Specification Language)[9]이 구조화된 문서의 처리 정보를 기술할 수 있는 표준으로 제안되었다.

본 연구에서는 SGML 문서의 포맷팅과 변환 과정을 수행하는 DSSSL 처리기를 개발하였다. 본 처리기는 DSSSL 언어의 구조적인 특성상, 하향식 접근 방식과 상향식 접근 방식을 모두 적용한다.

·본 연구는 정보통신부 '97 ~ '98 대학기초연구지원사업의 연구비 지원에 의한 것임.

한편 개발 결과를 문서편집기, 브라우저, 그리고 문서 변환기 등의 다양한 문서처리 환경에 적용하여 그 성능을 분석한 결과, DSSSL의 문서 처리 능력은 강력하였다. 그러나 처리 방식의 특성상, DSSSL은 사용자 인터랙션이 많으며 위지윅한 환경보다는 일괄처리 방식의 문서 처리 환경에 더 적합하였다.

본 논문의 구성은 다음과 같다. 2장에서는 DSSSL에 관한 간략한 개요를 기술하고 3장에서는 DSSSL 처리기의 설계 및 구현에 관하여 자세히 기술한다. 또한 4장에서는 개발 결과 및 이를 다양한 문서 처리 환경에 적용한 후 이를 통하여 DSSSL의 우수성과 개선이 요구되는 부분을 기술한다.

2. DSSSL 개요

일반적으로 문서 처리는 포맷팅과 변환의 두 가지로 나눌 수 있다. 이러한 처리 정보는 SGML 마크업을 이용하여 기술할 수 있다. 그러나 문서 내에 처리 정보를 포함하면, 문서의 재사용 및 재가공이 어렵고, SGML 문서의 가장 큰 특징인 구조적인 정보를 상실할 수 있다. 이에 SGML 문서의 처리 정보

를 기술할 수 있는 표준 안으로 DSSSL이 제안되었다.

DSSSL의 문서 처리 과정은 그림 1에서 알 수 있듯이 포맷팅 과정과 변환 과정으로 구성된다. DSSSL은 두 과정을 기술하기 위하여 각각 변환(transformation) 언어와 스타일(style) 언어를 제공한다. 또한 구조 기반 검색 및 객체의 생성 및 처리를 위하여 질의 언어(standard document query language:SDQL)와 수식 언어(expression language)를 제공한다. 특히 DSSSL은 포맷팅과 변환 정보를 함께 기술할 수 있으며, 여러 개의 스타일 문서를 결합할 수 있는 문서 구조를 정의한다. DSSSL의 문서 모델 및 주요 구성 언어에 대한 간략한 설명은 다음과 같다.

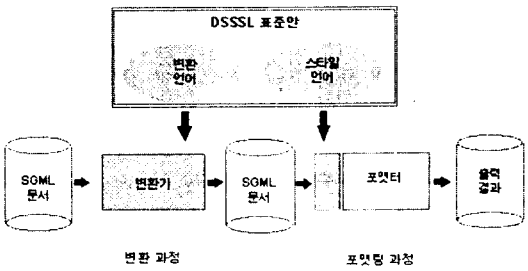

그림 1 DSSSL의 처리 과정

2.1 문서 모델

DSSSL은 문서 모델로 SGML 문서를 그래프 형태로 표현한 Grove(graph representation of property value)를 제공한다. 즉, SGML 문서에 DSSSL 처리 정보를 적용한다는 것은 결국 SGML 문서로부터 Grove를 생성한 후, 이에 DSSSL 처리 정보를 적용한다는 것이다.

구조화된 문서의 범용 모델인 Grove는 크게 실제 문서부와 프롤로그의 두 부분으로 구성되며 엘리먼트, 속성, DTD(document type definition), LPD(link process definition) 등의 트리를 포함하는 트리라고 볼 수 있다. 특히 Grove를 구성하는 각각의 노드는 SGML 특성 집합(SGML property set)에 속하는 클래스의 인스턴스이다. 또한 각각의 노드는 속성 정보와 타 노드와의 관계를 특성(property) 값을 통하여 표현한다. 특히 데이터 값으로 노드나 노드 리스트를 갖는 특성은 트리를 서로 연결하여 최종적으로 Grove를 형성한다.

2.2 변환 언어

변환 과정은 변환 언어에 의하여 기술되며, 한 개 이상의 SGML 문서를 임의의 개수의 문서로 변환한다. 특히 변환 과정의 결과는 SGML 문서이므로, 변환 과정은 포맷팅 과정의 전단계로써 수행될 수 있다. 즉, 보다 효과적인 포맷팅을 위하여 문서의 내용 및 구조를 수정하기 위해 사용할 수 있다.

변환 과정에서 수행 가능한 연산의 종류는 크게 두 가지로 분류할 수 있다. 먼저 DTD를 유지하면서 문서의 구조 및 내용을 수정할 수 있다. 예를 들어 구조적인 요소의 추가, 재배치, 그리고 그룹화를 통해 새로운 구조의 문서를 생성하거나 내용을 수정할 수 있다. 두 번째로 문서를 다른 DTD의 문서로 변환할 수 있다. 예를 들어 TEI(Text Encoding Initiative) 문서를 HTML(HyperText Markup Language) 문서로 변환할 수 있다. 이와 같이 변환 언어는 SGML 문서의 구조 및 내용을 모두 수정할 수 있다.

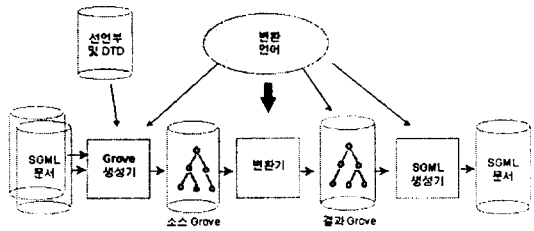

그림 2 DSSSL의 문서 변환 과정

변환 과정은 Grove 생성기(Grove building processor), 변환기(transformer), 그리고 SGML 생성기(SGML generator)의 세 부분으로 구성된다. 특히 변환기의 입력이 되는 변환 명세(transformation specification)는 결합 규칙(association)의 집합으로 구성된다. 각각의 결합 규칙은 소스 Grove의 객체들에 대한 변환 정보를 명시하며 질의 수식(query-expression), 변환 수식(transform-expression), 그리고 생략 가능한 우선순위 수식(priority-expression)으로 구성된다. 변환기의 출력 결과는 결과 Grove이다. SGML 생성기는 결과 Grove로부터 문서를 생성한다.

변환 언어와 스타일 언어는 서로 유사한데 이는 두 언어가 상당 부분 수식 언어에 기반하기 때문이다.

2.3 스타일 언어

스타일 언어는 포맷팅 과정을 기술한다. 포맷팅 과정은 스타일 정보를 문서에 적용하여 출력 매체상의 위치를 결정하고, 문서 내의 특정한 내용을 선택

하여 재 배치할 수 있으며, 새로운 요소를 삽입하거나 특정한 요소를 삭제할 수 있다.

DSSSL 은 문서의 레이아웃을 구성하는 요소로써 다양한 흐름 객체(flow object)를 정의한다. 즉, 문단, 그림, 표, 수식, 그리고 하이퍼텍스트 링크 등의 흐름 객체를 표현하기 위하여 47 개의 클래스를 제공한다. 또한 각각의 클래스마다 포맷팅 관련 정보를 정의할 수 있도록 다양한 특징(characteristics)을 제공한다. 예를 들어 문단의 여백, 그림의 크기, 하이퍼링크의 위치 등을 지정할 수 있다. 그밖에 DSSSL 은 응용 프로그램마다 새로운 흐름 객체 클래스와 해당 특징을 추가로 정의할 수 있도록 한다.

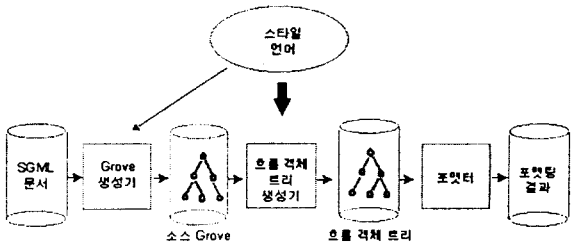

그림 3 DSSSL 의 포맷팅 과정

DSSSL 의 포맷팅 과정은 다음의 네 단계로 구성된다. 먼저 SGML 문서로부터 Grove 를 생성한다. 두 번째는 구성 규칙을 Grove 에 적용하여 흐름 객체 트리를 생성하는 단계이다. 전술한 바와 같이 DSSSL 은 페이지와 온라인 프리젠테이션을 모두 지원한다. 세번째 단계에서 페이지 기반의 경우, 해당 흐름 객체의 특징값을 참조하여 문서의 페이지 레이아웃을 결정한다. 마지막 단계는 흐름 객체 트리의 각각의 흐름 객체와 특징값을 이용하여 영역(area)의 집합을 생성한 후, 이를 출력 매체에 위치시킨다.

3. 설계 및 구현

본 처리기의 구현 환경은 IBM 호환 PC 에서 Visual C++ 5.0 을 개발환경으로 하며 C++ 언어를 사용하였다.

3.1 시스템 개요

본 연구에서는 개발된 DSSSL 처리기는 언어의 구조상 하향식과 상향식의 처리 과정을 모두 적용한다. 여기서 하향식 처리란 DSSSL 명세를 스캐닝하여 정의부(definition), 결합규칙, 그리고 구성 규칙에 대한 리스트를 형성하는 전처리 과정을 의미한다. 또한 상향식 처리 과정이란 정의부에 선언된 변수 및 프로시저의 정의에 관한 수식을 해석하여 해당 정보를 갱신하는 과정과 결합 규칙과 구성 규칙에 포함된 수식을 해석하는 과정에서 LALR 파싱 기법을 적용하는 것을 의미한다.

본 처리기는 SGML 문서의 변환과 포맷팅 과정을 위하여 여러 구성 요소로 이루어진다. 특히 변환 과정은 Grove 생성기, 변환 언어 처리기, 그리고 SGML 문서 생성기로 구성된다. 또한 포맷팅 과정은 Grove 생성기, 스타일 언어 처리기, 그리고 포맷터로 구성된다.

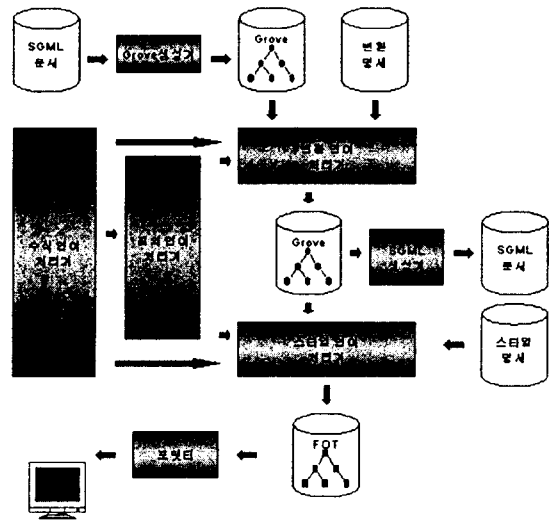

그림 4 DSSSL 처리기의 구성

3.2 시스템 구성

개발된 DSSSL 처리기의 구성 요소에 대한 보다 자세한 설명은 다음과 같다.

3.2.1 Grove 생성기

본 Grove 생성기가 지원하는 Grove 계획은 intrbase(intrinsic base), baseabs(base abstract), prlgabs0(prolog abstract level 0), instabs(instance abstract), prlgabs1(prolog abstract level 1) 등의 모듈에 속하는 모든 클래스 및 특성을 포함한다. 특히 intrbase 는 모든 노드에 대하여 지정 가능한 특성의 집합이며 baseabs, prlabs0, 그리고 instabs 모듈은 모든 Grove 계획에 자동으로 포함되는 모듈이다. 또한 prlabs1 은 변환언어를 사용하는 경우에 자동적으로 포함되는 모듈이다.

SGML 파서[10]의 출력 결과를 이용하여 Grove 를 생성하는 과정은 다음과 같다. Grove 는 크게 프롤로

그와 실제 문서부에 대한 두 부분으로 구성된다. 따라서 Grove 생성기는 먼저 파서가 문서의 프롤로그 부분을 처리한 후 제공하는 정보를 이용하여 Grove의 프롤로그 부분을 생성한다. 마찬가지로 파서가 실제 문서부를 파싱한 후 제공하는 정보를 이용하여 실제 문서부에 해당하는 부분을 생성하여 최종적으로 Grove를 완성한다.

특히 Grove 계획에 속하는 클래스는 C++ 클래스로 구현되었기 때문에 Grove와 관련된 정보는 객체의 형태로 API 함수에 의하여 응용 프로그램에 제공이 가능하다.

3.2.2 DSSSL 언어 처리기

3.2.2.1 전처리기

DSSSL 명세를 스캐닝하여 정의부, 구성 규칙, 그리고 결합 규칙에 대한 각각의 리스트를 생성한다. 특정 엘리먼트에 대하여 여러 개의 구성 규칙이 존재할 경우, 가장 구체적으로 명시된 순서로 정렬한다. 구성 규칙 리스트는 엘리먼트 이름과 생성 수식(construct expression)의 쌍으로 구성된다. 결합 규칙의 경우, 먼저 우선 순위 수식(priority expression)의 크기를 고려하여 질의 수식(query expression)과 변환 수식(transform expression)의 쌍으로 구성된 리스트를 구성한다. 특히 질의 수식을 해석하여 Grove 상의 적용 가능한 노드 리스트를 찾아낸 후, 노드 리스트와 변환 수식의 쌍으로 리스트를 재구성한다. 한편 정의부 리스트에 선언된 변수 및 프로시저에 대한 수식을 해석하여 이를 새로이 추가하거나 갱신한다.

3.2.2.2 스타일 언어 처리기

본 논문에서 개발한 DSSSL 처리기는 스타일 언어의 다양한 구성 규칙 중에서 엘리먼트 구성 규칙(element-construction-rule)과 디폴트 엘리먼트 구성 규칙(default-element-construction-rule)을 지원한다. 또한 스타일 정보를 보다 효과적으로 기술하기 위하여 질의 언어의 일부와 수식 언어의 모든 기능을 지원한다. 먼저 수식 언어를 처리하기 위해 LALR(1) 상향식 파싱 기법을 사용하였다. 또한 스타일 언어를 처리하기 위해서는 언어 구조상 하향식 접근 방법을 적용하여 처리하였다. 즉, 수식 언어 처리 모듈을 기반으로 스타일 언어를 처리하는 상위 모듈을 구현하였다.

DSSSL 처리기가 SGML 문서에 대한 Grove와 스타일 시트를 입력 받아 최종적으로 흐름 객체 트리를 생성하는 과정은 다음과 같다. 즉, 어휘분석기는 스타일 시트로부터 토큰을 분리해내어 스타일 언어 처리 모듈로 넘겨준다. 스타일 언어 처리 모듈은 하향식 파싱 기법을 사용하여 흐름 객체 트리를 생성하면서 필요한 경우 수식 언어 처리 모듈을 사용하여 스타일 시트 내에 포함되어 있는 수식을 처리한다.

수식 언어 처리 모듈은 LALR(1) 파싱 기법을 적용하며 어휘 분석기의 출력인 토큰을 사용하여 적합한 상위 생성 규칙을 찾아 나간다. 이 과정에서 빌트인(built-in) 프로시저를 사용하여 기본 사칙 연산을 비롯하여 리스트 및 문자열 등의 연산을 수행한다. 본 논문에서 구현한 수식 언어 처리 모듈은 DSSSL에 정의된 수식 언어의 모든 기능을 지원한다.

스타일 언어 처리 모듈은 흐름 객체를 생성하기 위하여 Grove를 재귀적으로 탐색하며 Grove 상의 각 노드에 해당하는 구성 규칙을 구성 규칙 리스트로부터 찾는다. 구성 규칙에 명시된 흐름 객체를 생성하고 이들을 트리 구조로 연결하여 최종적으로 흐름 객체 트리를 생성한다. 이 때 자식 흐름 객체는 부모 흐름 객체의 특징값을 상속 받는다.

3.2.2.3 변환 언어 처리기

변환 언어 처리기는 소스 Grove의 특정 노드에 결합 규칙 리스트의 변환 수식을 적용하여 새로운 Grove를 생성한다. 특히 변환 수식에 포함된 생성 프로시저(creation procedure)는 부 Grove를 결과 Grove 상의 생성 기준점(creation origin)에 상대적으로 위치시킨다. 이때 결과 Grove 상의 삽입될 위치는 먼저 생성 시블링(creation sibling)에 상대적으로 결정되며, 특히 의존 시블링은 해당 생성 시블링에 상대적으로 위치한다. 특히 생성 시블링간의 상대적인 위치 및 의존 시블링(dependent sibling)간의 상대적인 위치는 소스 Grove 상의 위치 순서에 따른다. 여기서 생성 시블링이란 동일한 생성 프로시저 및 생성 기준점에 대하여 생성된 부 Grove를 의미한다. 또한 의존 시블링은 특정 노드의 직접 의존 시블링(immediately dependent sibling)과 직접 의존 시블링의 의존 시블링의 집합을 의미한다. 또한 특정 노드의 직접 의존 시블링이란 해당 노드를 생성 기준점으로 하고, create-follow 나 create-preced 프로시저에 의하여 생성된 노드를 의미한다.

특히 생성 프로시저의 결과로 생성된 노드는 소스 Grove 의 해당 노드로부터 화살(arrow)이라고 불리는 링크를 갖는다. 또한 각각의 화살은 이의 시작 노드와 레이블 명칭을 인자로 갖는 생성 프로시저를 호출함으로써 또 다른 화살의 생성을 유발시킨다. 특히 무한 루프를 피하기 위하여, 순차적인 유발 관계를 갖는 일련의 화살들에서 첫번째와 마지막의 시작 노드와 레이블은 서로 상이해야 한다.

3.2.3 출력

3.2.3.1 포맷터

본 논문에서 개발한 포맷터는 흐름 객체 트리의 각종 스타일 정보를 화면에 출력한다. DSSSL 은 흐름 객체 트리를 출력하는 포맷터에 대해서 구체적으로 기술하지 않지만 영역(area)이라는 개념에 기반하도록 명시하고 있다. 즉, 흐름 객체 트리의 루트 노드를 제외한 모든 흐름 객체들의 포맷팅 결과는 영역의 연속적인 집합이다.

영역은 고정된 넓이와 높이를 가진 직사각형 형태이다. 영역은 그 특성에 따라 디스플레이(display) 영역과 인라인(inline) 영역으로 나누어 지고, 전자는 문단과 같이 행의 일부가 될 수 없는 영역을 의미하며 후자는 글자와 같이 행의 일부를 구성하는 영역을 의미한다. 특히 디스플레이 영역은 영역 컨테이너(area container) 안에 위치한다. 각각의 흐름 객체는 해당 클래스의 특징에 따라 디스플레이 또는 인라인 영역에 해당되거나 문맥에 따라 두 가지 경우에 모두 해당될 수 있다.

포맷터는 흐름 객체마다 지정되어 있는 이러한 영역의 속성에 근거하여 적절한 영역 트리를 생성한다. 생성된 영역 트리에는 각 영역의 포맷팅에 관련된 위치와 방향 그리고 크기 등의 속성값이 부여된다. 따라서 본 포맷터는 이러한 속성값에 근거하여 영역 트리를 Windows API 를 사용하여 화면상에 출력한다.

3.2.3.2 SGML 문서 생성기

본 SGML 문서 생성기는 결과 Grove 로부터 SGML 문서를 생성한다. 현재 본 문서 생성기는 검증 매핑(verification mapping)을 지원하지 않는다.

4. 개발 결과 및 고찰

본 연구에서는 개발된 처리기를 에디터, 브라우저, 변환기 등의 문서 처리 응용에 적용하여 DSSSL 처리 능력을 분석하였다.

포맷팅 면에서 DSSSL 은 정교한 페이지 기반의 프리젠테이션과 온라인 프리젠테이션을 모두 지원한다. 또한 DSSSL 은 기본적인 텍스트는 물론이고 그림, 표, 수식, 그리고 하이퍼텍스트 링크 등의 다양한 레이아웃 구성 요소를 표현하기 위하여 총 47 개의 흐름 객체 클래스를 제공하며 새로운 흐름 객체의 사용자 정의가 가능하다. 그리고 여러 스타일 명세를 결합할 수 있는 문서 구조와 프리젠테이션을 구성하는 객체간의 공간적인 동기화를 기술할 수 있는 방법을 제공한다. 이와 같이 다양한 포맷팅 기능과 더불어 질의 언어와 수식 언어를 사용하여 문서의 구조 및 내용에 대한 검색과 연산이 가능하기 때문에 DSSSL 의 포맷팅 기능은 매우 강력하다.

그러나 HyTime 에 기반한 하이퍼미디어 문서의 경우, 2 차원적인 프리젠테이션 기능과 더불어 문서를 구성하는 멀티미디어 객체 간의 시간적인 동기화에 대한 지원이 요구된다[11]. 또한 DSSSL 의 포맷팅 과정은 문서의 내용 및 스타일 정보가 변경되면 Grove 와 sosofo 를 수정한 후, 흐름 객체 트리를 새로이 생성한다. 이러한 처리 방식은 사용자 인터랙션이 많이 요구되는 위지윅(WYSIWYG) 환경보다는 일괄 처리 방식의 브라우징 또는 인쇄 환경에 더 적합하다고 볼 수 있다.

한편 SGML 문서간의 변환에 있어서, DSSSL 변환 언어는 스타일 언어와 마찬가지로 질의언어와 수식 언어에 기반하기 때문에 강력한 문서 변환을 지원한다. 그러나 변환된 문서의 적합성 여부를 검사하기 위하여 별도의 검증 과정을 필요로 한다.

참고문헌

[1] ISO 8879, *Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML)*, ISO(http://www.iso.ch), 1986

[2] Martin Bryan, *SGML : An Author's Guide to the Standard Generalized Markup Language*, Addison-Wesley, New York, 1988.

[3] Eric van Herwijnen, *Practical SGML : Second Edition*, Kluwer Academic Publishers, Boston, 1994.

[4] Charles F Goldfarb and Yuri Rubinsky, *The SGML Handbook*, Clarendon Press, Oxford, 1990

[5] Sean McGrath, *Parseme.1st : SGML for Software Developers*, Prentice Hall, 1998

[6] Martin Bryan, *Sgml and Html Explained*, Addison-Wesley, New York, 1997

[7] Liora Alschuler, *ABCD SGML-A User's Guide to*

Structured information, International Tompson Computer Press, Boston, 1995

[8] Yaron Wolfsthal, "Style Control in the Quill Document Editing System," Software-Practice and Experience, Vol. 21, No. 6, pp. 625- 638, June, 1991

[9] ISO/IEC 10179, *Information technology - Text and office system - Document Style Semantics and Specification Language(DSSSL)*, ISO/IEC, 1996

[10] 고영곤, 최윤철, "한글 SGML 문서처리를 위한 파서의 개발에 관한 연구", 한국통신학회 춘계 학술논문발표회, pp. 535-543,1997

[11] Jacco van Ossenbruggen, Lynda Hardman, Lloyd Rutledge and Anton Eliëns, "Style Sheet Support for Hypermedia Documents", Proceedings of Hypertext 97, April 1997.

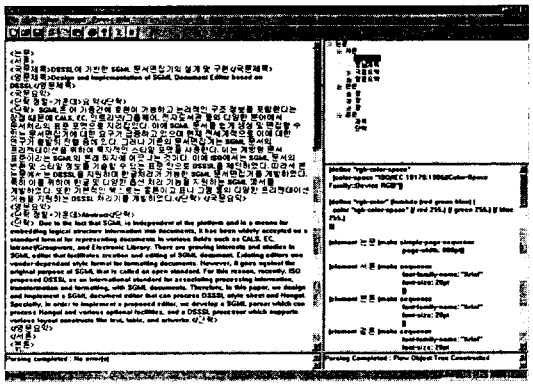

그림 5 DSSSL 기반의 문서편집기

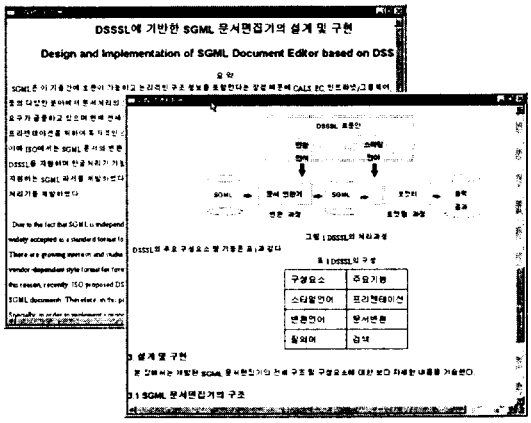

그림 6 문서편집기의 출력 결과

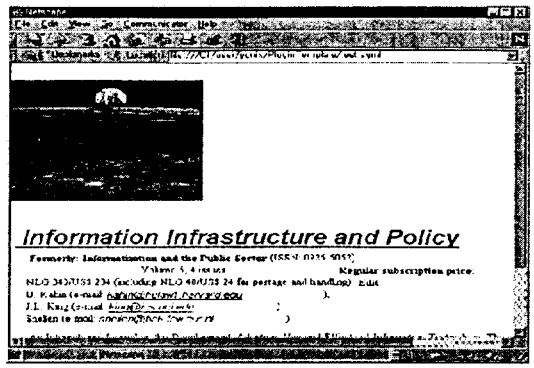

그림 7 DSSSL 기반의 플러그 인 브라우저

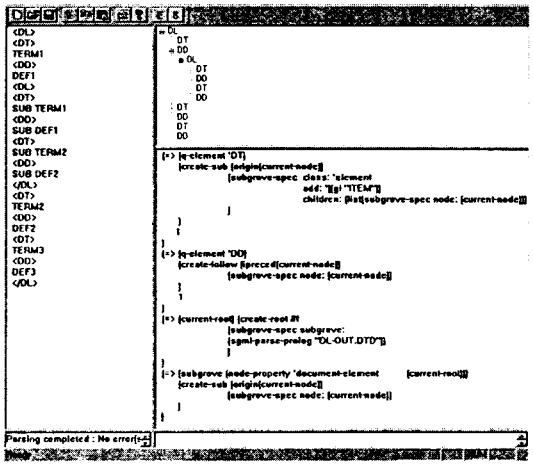

그림 8 DSSSL 기반의 문서변환기

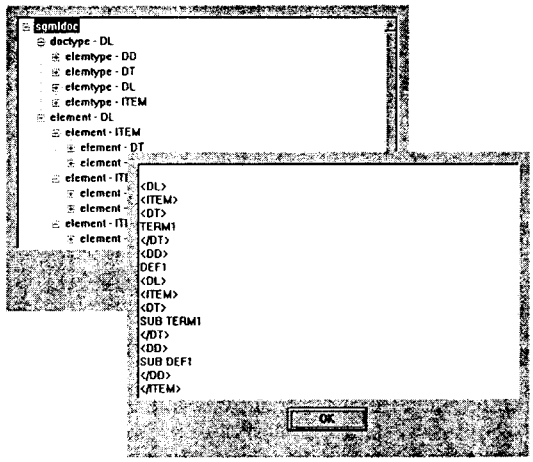

그림 9 문서변환기의 출력 결과