

# DSSSL에 기반한 SGML 문서 표현 시스템의 설계 및 구현

(Design and Implementation of an SGML Document  
Presentation System based on DSSSL)

김창수<sup>†</sup> 정회경<sup>\*\*</sup> 윤보현<sup>\*\*\*</sup> 강현규<sup>\*\*\*\*</sup>

(Changsoo Kim)(Hoekyung Jung)(Bohyun Yun) (Hyungyu Kang)

**요약** 본 논문은 SGML(Standard Generalized Markup Language) 문서를 포매팅 처리하여 표현하기 위한 SGML 문서정보 표현 시스템의 설계 및 구현에 관한 것이다. 이를 위해 본 논문에서는 ISO/IEC에서 SGML 문서 포매팅을 위해 정의한 모델인 DSSSL(Document Style Semantics and Specification Language) 표준에 따라 시스템을 설계하였고, SGML 문서를 온라인으로 포매팅 처리하는 시스템을 구현하였다.

본 시스템은 한글 처리를 지원하면서 임의의 DTD(Document Type Definition), SGML 문서, DSSSL 스타일 시트에 대한 파싱 기능을 가지며, 텍스트뿐만 아니라 표, 목록, 그림 등 다양한 명세 표현이 처리 가능한 포맷터를 포함한다. 이는 이 기종간에 포맷 정보를 포함한 SGML 문서 교환에 대한 사용자 욕구를 충족시킬 수 있으리라 보며, SGML 문서처리 환경 구축에 크게 기여하리라 본다.

**Abstract** This paper is for the design and implementation of SGML document presentation system to format SGML document based on DSSSL. ISO/IEC proposed DSSSL as the technological standard for formatting and transforming SGML document. This paper shows the design of this system in accordance with the model defined in DSSSL.

This system, which is able to provide Korean, has parsing function of arbitrary DTDs, SGML documents and DSSSL style sheets, and contains a formatter that can manage various specifications, such as graphs, lists, pictures, as well as text. This will satisfy a user's intent for exchange of SGML documents that involve format information between different types, greatly contribute to make up SGML document processing environment.

## 1. 서론

컴퓨터의 사용이 보편화되고, 정보화 사회가 되어감에 따라 컴퓨터를 이용한 문서처리의 중요성이 증대되어 왔다. 따라서 문서를 전자적으로 처리하기 위해 워드프로세서, 전자출판(DTP : Desk Top Publishing)과 같

은 전자문서 처리시스템의 사용이 필수로 되었다. 그러나 기존의 시스템으로 작성된 문서는 자기 독자적인 문서 구조와 다양한 포맷(format) 정보를 갖고 있으므로 서로 다른 시스템 및 장치를 갖는 문서 처리 환경에서 상호 호환되지 않는다는 문제점을 안고 있었다. 이에 문서 교환 및 공유를 위한 표준 문서구조 모델이 요구되어 ISO에서는 서로 다른 이 기종간의 효율적인 문서정보 교환을 위한 표준 모델로 ISO 8879 SGML을 국제 표준으로 제정하였다[1,2,3]. SGML은 문서의 포맷이나 배치를 제외한 논리적 구조와 내용만을 기술한다. 따라서 논리 구조만을 기술한 SGML 문서를 사용자에게 표현하고, 포맷 정보를 포함하여 이 기종 시스템간에 교환하기 위한 연구가 요구되어 ISO에서는 SGML 문서의 스타일과 요소간의 대응 관계를 기술하기 위한 표준은

† 학생회원 : 배재대학교 컴퓨터공학과  
sungu@markup.paichai.ac.kr

\*\* 중신회원 : 배재대학교 컴퓨터공학과 교수  
hkjung@mail.paichai.ac.kr

\*\*\* 비회원 : 한국전자통신연구원 문서정보연구팀 연구원  
ybh@etri.re.kr

\*\*\*\* 중신회원 : 한국전자통신연구원 문서정보연구팀 연구원  
hkkang@etri.re.kr

논문접수 : 1999년 4월 30일

심사완료 : 2000년 7월 15일

로 ISO 10179 DSSSL을 제안하였다[3,4,5,6]. DSSSL은 SGML 문서를 변환하거나 사용자에게 표현하기 위한 표준 구조와 방법을 제공하고 있다.

이에 본 논문에서는 DSSSL에 기반하여 SGML 문서를 포매팅 처리하기 위한 시스템을 설계 및 구현하였다. 이를 위해 SGML 문서와 DSSSL 처리기의 중간 매개체로 GROVE(Graph Representation Of property ValuEs)를 사용하였고, DSSSL 처리기와 포맷터는 흐름 객체 트리(FOT : Flow Object Tree)를 사용하여 포맷 처리하였다[7,8]. 또한 파서와 DSSSL 처리기를 이용하여 흐름 객체 생성기를 개발하였고, 흐름 객체 트리를 운행하면서 SGML문서의 내용을 포맷팅하는 포맷터를 개발하였다.

2. DSSSL 개요

DSSSL은 플랫폼이나 특정 제품에 독립적인 문서의 변환이나 포매팅 지정을 위한 국제 표준으로, ISO/IEC HyTime (HyperMedia and Time-based Structuring Language)[9]의 PSDR(Property Set Definition Requirements)에 따라 정의된 포맷을 가지는 임의 문서에 사용될 수 있다. 특히 ISO 8879인 SGML에 따라 마크업된 문서 표현을 지정하기 위해 사용되고 있다[7].

DSSSL은 SGML 문서의 스타일이나 배치를 지정하기 위한 표준화된 문법과 의미(semantic)를 제공하므로 엘리먼트 또는 속성에 대한 스타일을 지정할 수 있으며, 또한 스타일 명세도 교환할 수 있다 [10].

DSSSL은 다음 4부분에 대해 표준화하였다[10].

- ① 하나의 SGML 문서를 다른 SGML 문서로 변환하기 위한 언어와 처리 모델(변환 언어)
- ② SGML 문서에 포매팅 특징을 지정하기 위한 언어(스타일 언어)
- ③ SGML 문서의 각 부분들을 구분하기 위한 언어(질의 언어)
- ④ 객체를 생성하고 처리하기 위한 언어(표현식 언어)

표현식 언어는 변환 언어, 스타일 언어, 표준 문서 질의 언어의 기초가 된다. 즉 변환 언어, 스타일 언어, 표준 문서 질의 언어는 표현식 언어를 사용해 기술되며, 또한 표준 문서 질의 언어는 변환 언어와 스타일 언어에서 사용할 수 있다.

DSSSL의 개념적 모델은 변환처리와 포매팅 처리단계로 나눌 수 있다.

변환처리는 하나 이상의 DTD로 작성된 SGML 문서를 다른 DTD 형태로 변환하거나 단일 DTD내에서 문

서의 구조를 변경하는 처리 단계이며, 포매팅 처리는 SGML 문서의 포매팅 특징을 처리하는 단계이다[6]. 변환처리는 포매팅 처리와 함께 또는 독립적으로 사용될 수 있다.

그림 1은 DSSSL 표준에서 정의한 SGML 문서의 처리과정을 보여준다. 그림에서 회색 부분은 DSSSL에서 표준화한 부분을 나타낸다. 변환기는 SGML 문서를 입력받아 변환 명세에 따라 변환 처리하여 새로운 SGML 문서를 생성한다. 이렇게 만들어진 SGML 문서는 포매팅 처리에 대한 입력으로 사용될 수 있다. 포매팅 처리에 입력된 SGML 문서는 포매팅 처리를 거쳐 SPDL(Standard Page Description Language)이나 다른 출력 포맷으로 출력된다.

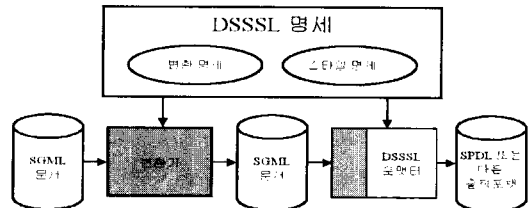


그림 1 DSSSL의 개념적 모델

2.1 DSSSL 처리

DSSSL 처리는 DSSSL 언어에 의해 처리되는데 그림 1에서처럼 변환처리는 변환 명세와 DTD, 선언에 의해 처리되고, 포매팅 처리는 스타일 명세에 의해 처리된다.

2.1.1 변환 처리

변환 처리는 변환명세와 SGML DTD들에 의해 SGML 문서를 다른 SGML 문서로 변환한다. 그림 1의 변환처리에 의해 생성된 SGML 문서는 포매팅 처리의 입력으로 사용될 수 있다. 그러나 변환 처리에서 사용된 모든 처리는 포매팅 처리와 독립적이다. 그림 2는 변환 처리과정을 나타낸다.

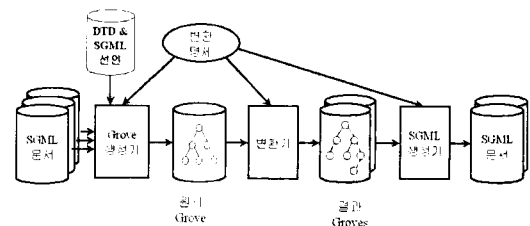


그림 2 변환 처리

변환 처리의 입력은 SGML 문서, DTD, SGML 선언이다. 입력된 SGML 문서들을 파서와 GROVE 생성기에 의해 원시 GROVE로 표현된다. GROVE는 파싱(parsing) 결과에 대한 메모리상의 표현으로써, SGML 처리 시스템에 대해 정렬된 데이터 API인 종합 데이터의 추상적 개념을 제공한다. SGML 응용들이 작업하는데 바탕이 되는 가능한 정보의 총체적인 집합을 정의하고, SGML 도구를 처리하기 위한 추상적 개념을 제공하므로, 본 시스템에서도 이러한 GROVE 개념을 이용하여 DSSSL 처리기가 요구하는 데이터 구조를 만족시키기 위해 GROVE를 DSSSL 처리기의 중간 매개체로 사용하였다. 원시 GROVE는 변환 명세의 정의에 따라 변환 처리되어 결과 GROVE로 변환되고, 결과 GROVE는 SGML 생성기에 의해 유효한 SGML 문서로 출력된다.

위와 같은 처리에 포매팅 정보를 추가하여 변환 처리를 할 수도 있다. 즉, 변환 처리는 포매팅 처리를 쉽게 하기 위해 문서를 변환하는데 사용될 수 있고, 전혀 사용되지 않을 수도 있다.

변환 처리는 다음과 같은 구성요소를 가진다.

① GROVE 생성기(Grove Constructor)

이 처리의 입력은 SGML 문서이고, SGML 문서 또는 부분서(subdocument)는 파싱되어 노드(node)들의 집합인 GROVE로 표현된다.

② 변환기(Transformer)

변환처리의 입력은 SGML 문서와 SGML 선언, DTD, 변환 명세이다. 변환 명세는 관계 지정들의 집합으로 구성된다.

③ SGML 생성기 (SGML Generator)

변환처리 이후에 GROVE는 교환과 검증 또는 포매팅 처리의 입력을 위해 SGML문서로 번역된다. 여기서 SGML 생성기의 출력은 유효한 SGML 문서이어야 한다.

2.1.2 포매팅 처리

DSSSL 표준은 흐름 객체 트리라는 중간적 트리에 포매팅 특징들을 첨부함으로써 포매팅 문서의 시각적 표현을 정의한다. DSSSL에서 의미하는 흐름 객체란 포매팅 정보를 포함하는 특정 영역의 그룹이다. 여기서 특정 영역이란 포매팅하기 위한 최소한의 객체로서 존재하며 블록 단위 또는 인라인 단위로 처리된다. 이에 DSSSL 엔진에서는 이들을 쉽게 처리하기 위해 영역 그룹들의 연속을 트리 구조 형태로 표현하게 하였다. 그러므로 DSSSL 엔진에서 흐름 객체는 매우 중요하다. 그리고, DSSSL 표준은 구성 요소 또는 포매팅 알고리

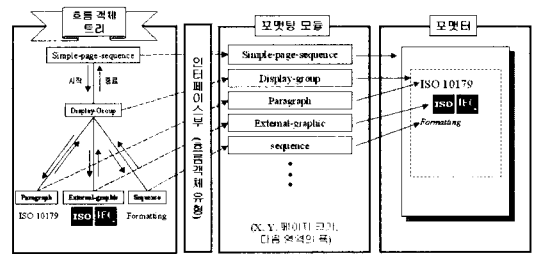


그림 3 포매팅 처리 과정

즘을 제약하지 않아 매우 융통성이 있으며, 또한 포맷터의 출력에 대해서도 정의하지 않으나, 인쇄 또는 이미징에 알맞은 포맷된 문서이어야 한다. 포매팅 처리는 스타일 언어를 사용하여 기술된다.

그림 3은 포매팅 처리 과정을 보인다. 포매팅 처리는 다음과 같은 개념적 단계로 구성된다.

먼저 SGML 문서로부터 GROVE를 생성하고, 흐름 객체 트리를 생성하기 위해 소스 트리의 객체에 구성 규칙(construction rule)을 적용한다. 그리고 적용되는 객체와 관련된 흐름객체 클래스의 의미와 특징(Characteristic) 값에 의해 지정된 규칙에 따라 내용들을 구성하고 배치한 뒤, 쪽(page) 시퀀스 흐름 객체와 컬럼-집합 시퀀스 흐름 객체의 특징에 따라 쪽과 도형을 정의하고, 영역(area) 시퀀스를 생성하기 위해 흐름 객체를 포맷한다.

3. SGML 문서 표현 시스템 설계

3.1 SGML 문서 표현 시스템의 구성

본 논문에서 개발한 SGML 문서 표현 시스템은 SGML과 DSSSL 문서를 파싱하기 위한 파서와 GROVE 생성기, 흐름 객체 생성기, 포맷터로 구성된다. 그림 4는 본 시스템의 전체 구성도 이다.

본 시스템은 입력으로 SGML 문서와 DSSSL 문서를 받아들여 파싱 과정을 거치는데, 이때 파서와 GROVE

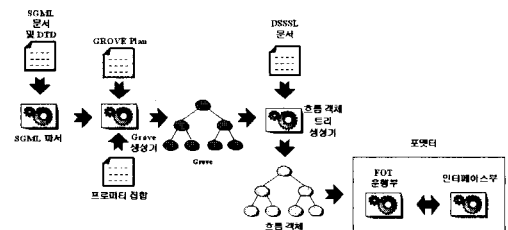


그림 4 시스템 전체 구성도

생성기는 입력된 SGML 문서에 대한 GROVE를 생성한다. 또한 흐름 객체 생성기는 DSSSL 스타일 시트를 파싱해 스타일 시트에 정의된 흐름 객체 클래스의 객체를 생성한다. 흐름 객체 생성기는 생성된 GROVE의 루트 노드를 시작으로 마지막 노드까지 운행하면서 각 엘리먼트 노드를 정의한 흐름 객체를 생성하여 흐름 객체 트리를 구성한다. 마지막 노드를 처리하면 흐름 객체 생성기는 생성된 흐름 객체의 루트 노드와 함께 포맷터를 호출하게 된다. 포맷터는 입력된 루트 노드를 시작으로 흐름 객체 트리를 운행한다. 현재 처리중인 흐름 객체 노드를 인수로 인터페이스부를 호출하면 인터페이스부는 흐름 객체 노드와 디스플레이 되는 영역의 X, Y 좌표, 실제 쪽의 크기, 그리고 흐름 객체가 처리될 영역의 크기를 인수로 해당 흐름 객체 처리 함수를 호출하여 포맷팅 한다.

### 3.2 SGML 파서

본 시스템에서 사용한 파서의 특징을 살펴보면, SGML 문서에 대한 검증뿐만 아니라 DTD와 SGML 선언에 대한 검증도 제공한다. 또한 임의의 구체구문 (concrete syntax)을 지원하며 SHORTTAG, OMITTAG, RANK, SUBDOC, LINK (SIMPLE, IMPLICIT, EXPLICIT)와 같은 SGML 특성의 선택 사항도 지원한다. 그러나 DATATAG와 CONCUR는 지원하지 않는다. 엔터티 관리자는 HyTime[9]의 FSI (Formal System Identifiers)와 SOC(SGML Open catalog)를 지원한다. 또한 내부적으로 16비트 문자집합으로 처리되어 태그 이름이나 다른 태그에 한글이나 일본어와 같은 16비트 문자집합을 사용할 수 있으며 UCS-2와 UTF-8을 사용하는 ISO/IEC 10646 (Unicode)를 지원한다[11].

본 시스템의 파서는 C++와 객체 지향 프로그래밍 방식을 사용하여 개발되었으며, Windows 95와 NT 환경에서 사용할 수 있다.

본 시스템에서 파서는 GROVE 생성기와 흐름 객체 생성기와 함께 처리 호출되면서 GROVE 생성 및 DSSSL 스타일 시트 파싱과 같은 핵심기능을 담당한다.

### 3.3 GROVE 생성기

GROVE 생성기는 SGML 문서의 파싱 결과를 입력으로 받아드리고 GROVE를 출력한다.

GROVE 생성 단계를 살펴보면, SGML 파서는 입력된 SGML 문서의 구문적 구조를 분석하면서 GROVE 생성기가 GROVE에 해당 노드와 프로퍼티를 생성하도록 메시지를 보낸다.

"Grove plan"은 GROVE 생성기에 의해 사용되는 프

로퍼티 집합에 포함되는 클래스와 프로퍼티를 포함시키는 문서이다.

```

<!DOCTYPE메모 [
<!ELEMENT메모 o o (단락*)>
<!ELEMENT단락 (#PCDATA)>
<!ATTLIST 단락 이름 ID #REQUIRED
>]>
<단락 이름=첫번째> 안녕하세요 </단락>
<단락 이름=두번째> 반갑습니다 </단락>

<para name = first> 안녕하세요
<q>정말</q> 반갑습니다.</para>
<para name = second>
안녕하세요.</para>

```

그림 5 SGML 문서

그림 6은 그림 5의 문서에 대한 GROVE 표현 예이다.

그림 6에서 "메모" 엘리먼트가 두개의 "단락" 엘리먼트 노드를 가지고 있고, 각 "단락" 엘리먼트는 서브노드 프로퍼티(SubnodeProperty)를 가지는데, 하나는 속성 노드이고 다른 하나는 내용 노드이다.

이렇게 생성된 GROVE는 흐름 객체 생성기에 입력된다.

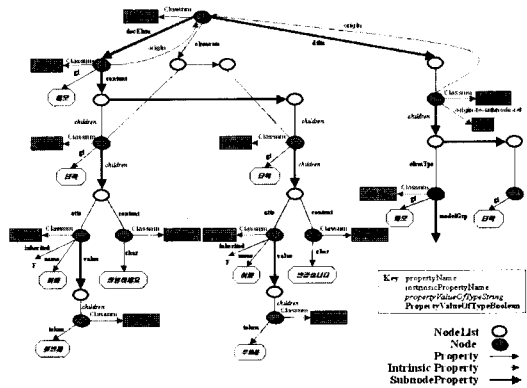


그림 6 생성된 GROVE의 예

### 3.4 흐름 객체 트리 생성기

흐름 객체 트리 생성기는 DSSSL 스타일 언어에 대한 구현이다.

DSSSL 스타일 시트를 파싱하기 위해 흐름 객체 생성기는 파서를 사용한다. 흐름 객체 생성기는GROVE

생성기에서 생성한 GROVE를 입력받아 흐름 객체 트리를 생성하고, 포맷터를 호출함으로써 처리를 종료하게 된다.

흐름 객체 트리 생성기는 DSSSL 문서를 파싱해 스타일 시트에서 정의한 각 흐름 객체에 대한 객체를 생성한다. GROVE와 흐름 객체 클래스의 객체가 생성되면 GROVE의 루트 노드를 시작으로 스타일 시트에서 정의한 노드에 해당하는 흐름 객체 트리의 노드를 생성한다. 마지막 노드를 처리할 때까지 이러한 방식으로 처리하면서, 포맷터에서 처리 가능한 흐름 객체 트리를 완성한다.

흐름 객체 트리가 완성되면 완성된 흐름 객체 트리의 루트 노드와 함께 포맷터를 호출하여 문서를 포맷팅 처리하게 된다.

표 1은 본 시스템에서 처리 가능한 흐름 객체 클래스와 특징들에 대한 목록을 표현한다.

표 1 처리가능한 흐름 객체 클래스와 특징들

흐름 객체 클래스 특징	특징
Simple-page-sequence	page-width, page-height, left-margin, right-margin, top-margin, bottom-margin
Paragraph	line-spacing, first-line-start-indent, font-family-name, font-weight, font-posture, font-size, quadding, start-indent, end-indent, space-before, space-after
line-field	field-width, field-align
External-graphic	display, entity-system-id, notation-system-id, display-alignment, start-indent, end-indent, space-before, space-after
Sequence	구현된 클래스들 중에서 상속되는 특징
Display-group	space-before, space-after
Table	table-width, table-border, display-alignment, start-indent, end-indent, writing-mode, space-before, space-after
Table-row	
Table-cell	n-columns spanned

특징은 DSSSL 표준에서 정의한 각 흐름 객체 클래스의 포맷팅 정보를 저장하기 위한 것이다.

simple-page-sequence 흐름 객체 클래스는 가장 기초가 되는 쪽 모델의 흐름 객체로 쪽 영역을 생성하기

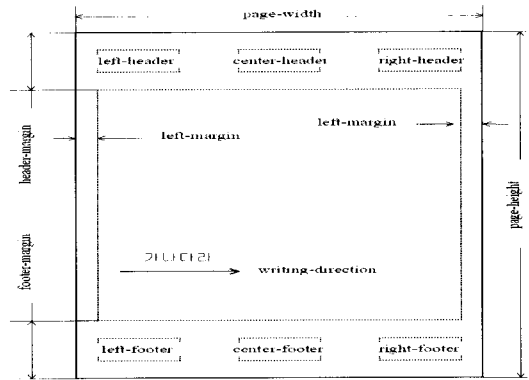


그림 7 simple-page-sequence 흐름 객체 클래스의 특징값

위해 포맷된다. 또한 표현되는 어떠한 흐름 객체라도 가질 수 있으나 다른 흐름 객체 클래스 내에서는 발생할 수 없다[10]. 그림 7은 simple-page-sequence 흐름 객체 클래스의 특징값 사용에 대한 설명이다.

Paragraph 흐름 객체 클래스는 단락을 표현하기 위해 포맷되며, line-field 흐름 객체 클래스는 인라인 되도록 포맷된다. External-graphic 흐름 객체 클래스는 External 엔터티로 포함되는 그래픽을 표현하기 위해 포맷되고, sequence 흐름 객체 클래스와 Display-group 흐름 객체 클래스는 자식으로 생성되는 영역의 연결을 위해 포맷된다. 이 두 클래스의 차이점은 sequence 흐름 객체 클래스의 자식으로 생성되는 클래스는 인라인 되거나 디스플레이 될 수 있으나 display-group 흐름 객체 클래스의 자식은 인라인 될 수 없다.

테이블을 포맷팅하기 위해 사용되는 흐름 객체 클래스로 table, table-row, table-cell 흐름 객체 클래스를 사용하였다. Table 흐름 객체 클래스는 자식을 table-row와 table-cell 흐름 객체 클래스를 가진다. Table-row 흐름 객체 클래스는 테이블에서 한 열에 생성되는 table-cell 흐름 객체 클래스를 그룹 시킨다. 따라서 table-row 흐름 객체 클래스의 자식으로 발생하는 table-cell 흐름 객체 클래스는 하나의 열로 포맷 처리된다. Table-cell 흐름 객체 클래스는 table과 table-row 흐름 객체 클래스의 자식으로만 발생할 수 있으며, 자식으로서는 디스플레이 되는 어떠한 흐름 객체 클래스라도 가질 수 있다[5].

그림 8은 흐름 객체 트리 생성기에서 입력된 GROVE를 운행하면서 흐름 객체 트리를 생성하는 과정과 포맷터를 호출하는 시점을 나타낸다.

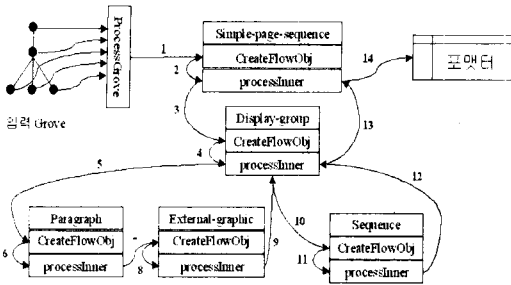


그림 8 흐름 객체 트리 생성 과정

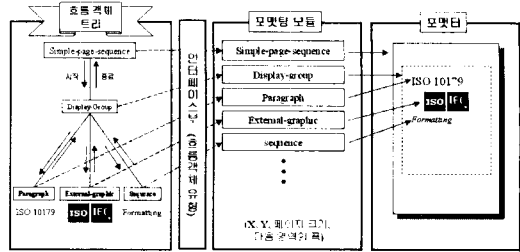


그림 9 포매팅 함수 처리

생성된 GROVE의 루트 노드를 흐름 객체 생성기에 입력하면 흐름 객체 생성기는 그림 8의 1번부터 14번까지의 단계를 거쳐 처리된다. 각 흐름 객체 클래스는 CreateFlowObj함수와 processInner 함수를 가지고 있다. CreateFlowObj 함수는 처리중인 노드의 객체를 생성과 특징값의 할당을 처리하는 함수이고, processInner 함수는 이전 흐름 객체와 다음 흐름 객체의 관계 지정과 다음 흐름 객체의 호출을 담당하는 함수이다. 흐름 객체 트리가 완성되면 흐름 객체 생성기는 흐름 객체 트리의 루트 흐름 객체를 포맷터에 넘겨주어 포매팅 처리를 완성한다.

3.5 포매팅

포맷터는 흐름 객체 트리(FOT) 운행부와 인터페이스 부로 나뉘어 진다.

3.5.1 흐름 객체 트리 운행부

흐름 객체 트리 운행부는 흐름 객체 트리 생성기에서 생성한 흐름 객체 트리의 노드 순회를 처리하는 것으로, 루트 흐름 객체를 시작으로 순회하는 전위순회(pre-order) 방식을 사용한다. 이는 운행과정에서 처리되는 흐름 객체의 유형에 따라 인터페이스부를 호출하여 각 흐름 객체를 포매팅 한다.

3.5.2 인터페이스부

인터페이스 부는 흐름 객체 노드를 중심으로 자식 노드들을 재귀적(Recursion) 호출 방식으로 처리한다.

그림 9는 그림 8의 흐름 객체 트리를 포매팅 처리하기 위해 각 흐름 객체의 종류에 따라 해당 포매팅 처리 함수를 호출하여 처리하는 과정을 보여준다.

인터페이스 부의 재귀적 호출 방식을 살펴보면 인수로 넘어온 노드를 시작으로 이 루트의 자식으로 존재하는 모든 흐름 객체 노드들을 재귀적으로 호출하게 된다. 재귀 호출의 인수로는 흐름 객체 노드와 디스플레이 되어야 하는 X, Y 좌표, 실제 쪽의 크기, 다음 흐름 객체가 처리되어야 할 영역의 폭 크기가 사용된다. 각 흐름 객

체 함수는 입력된 흐름객체의 유형과 특징값에 따라 포매팅 처리를 하고 다음 흐름 객체 노드를 가져온다. 위와 같은 방식으로 다음 흐름 객체의 유형에 따라 포매팅 함수를 계속 처리한다.

4. 구현 및 고찰

본 논문에서 개발한 시스템의 구현환경은 IBM 호환 PC에서 Windows NT 4.0 운영체제 환경에서 개발하였으며, 프로그래밍 언어는 Visual C++ 5.0을 사용하여 개발하였다.

4.1 구현

본 시스템의 특징은 SGML 문서에서 엘리먼트와 속성에 한글을 사용할 수 있고, 또한 DSSSL 문서에서도 한글 사용이 가능하다.

시스템 설계에 있어서 DSSSL 표준에서 정의한 포매팅 처리 과정에 부합되게 설계되었으며, 47개의 흐름 객체 클래스 중에서 가장 많이 사용되는 9개의 흐름 객체 클래스(simple-page-sequence, display-group, paragraph, sequence, line-field, external-graphic, table, table-row, table-cell)를 구현하였다.

이외에 파서 메시지 창과 흐름 객체 트리 창을 두었는데, 메시지 창은 사용자가 작성한 DSSSL 스타일 시트의 오류를 보고해 주어 DSSSL 스타일 시트 오류 수정에 도움을 줄 수 있도록 하였으며, 흐름 객체 트리 창은 SGML 문서와 DSSSL 스타일 시트에 의해 생성된 흐름 객체 트리가 사용자의 요구에 맞게 생성되었는지 확인할 수 있게 하였다.

4.1.1 흐름 객체 포매팅

본 시스템의 구성은 파서와 GROVE 생성기, 흐름 객체 생성기, 포맷터로 구성되는데, SGML 문서를 포매팅하는 과정에서 위의 모듈들을 사용하도록 API 함수들을 개발하였다.

본 시스템은 메시지를 모듈간의 통신 수단으로 사용

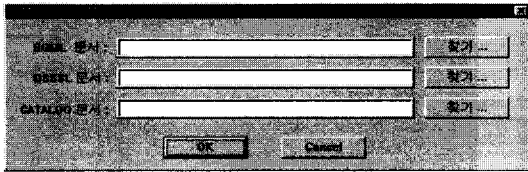


그림 10 SGML 문서 입력 창

하였는데, 정의한 메시지에 맞는 파싱의 시작과 끝, 파서의 오류 메시지, 쪽 영역의 확장 메시지가 있다.

그림 10은 본 시스템에서 데이터 입력을 위한 입력 대화 상자이다.

그림 10에서 사용자가 SGML 문서와 DSSSL 문서, CATALOG 파일을 개별적으로 입력 할 수 있도록 하였다.

CATALOG 파일은 입력된 SGML 문서에 대한 DTD와 SGML 선언 파일 연결을 지정하는 파일이다. 따라서 DSSSL 포매팅 처리의 입력으로 지정해야 하는 DTD와 선언을 직접 입력하지 않아도 CATALOG 파일에 지정해 놓으면 SGML 문서만 입력해도 해당 DTD와 선언 파일을 입력 할 수 있도록 한 파일이다. CATALOG 파일을 사용하는 장점은 하나의 DTD에 따라 작성된 여러 개의 SGML 문서를 처리할 때, DTD와 DSSSL 스타일 시트를 매번 지정하지 않아도 된다는 점이다.

그림 11은 한국정보과학회 논문지에 기 발표된 논문의 처리 결과를 보여주고 있다.

DSSSL 표준은 루트 흐름 객체를 simple-page-sequence, page-sequence, scroll 흐름 객체로 정의하도록 권장하고 있다. 이에 본 시스템에서는 루트를 simple-page-sequence로 정의하여 지원한다.

그림 12는 테이블 및 외부 엔티티로 지정되는 이미지 처리 결과를 보여주고 있다. 테이블과 관련된 흐름 객체

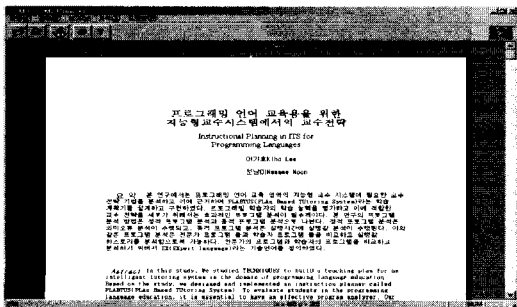


그림 11 한국정보과학회 논문의 처리 결과

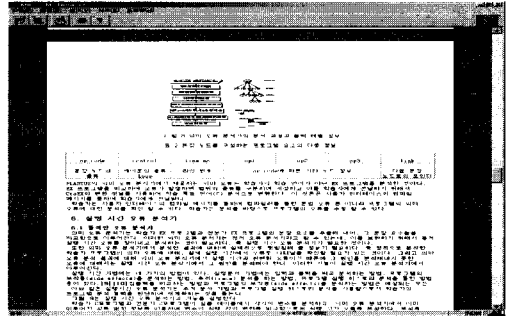


그림 12 테이블 및 이미지 처리 결과

클래스 중 본 시스템에서는 Table, Table-row, Table-cell 클래스를 지원하고 있으며, 그림과 같이 한 Table-row에서 셀 병합을 지원한다. 외부 엔티티로 지정되는 이미지 처리와 관련된 External-graphic 흐름 객체의 처리 결과로 entity-system-id 특징값으로 이미지의 경로명이 입력되고 notation-system-id 특징값으로 이미지의 표기법이 입력되어 처리된다.

4.1.2 오류보고

DSSSL 표준은 처리에 있어 검증된 SGML 문서에 대한 처리만을 보장하도록 하고 있다. 이에 본 시스템에서도 SGML 문서는 검증된 문서라는 것을 전제하여 처리한다. 그러나 본 시스템은 DSSSL 스타일 시트에 대한 오류를 찾아 오류 보고창을 이용해 사용자에게 어느 스타일 시트의 몇 번째 줄, 몇 번째 칸에서 오류 또는 경고가 어떠한 내용으로 발생하였는지를 알려준다. 그림 13은 DSSSL 스타일 시트에서 오류가 발생하였을 때

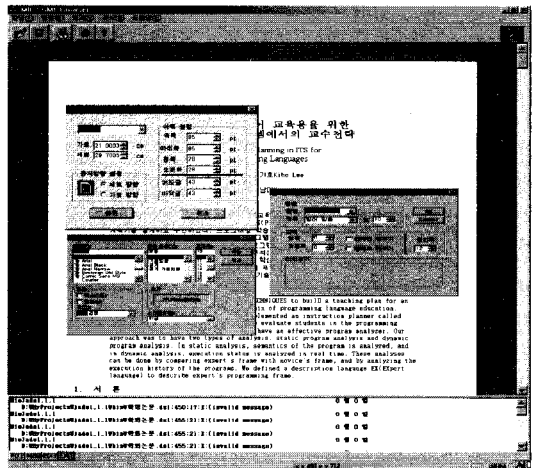


그림 13 오류 보고 및 스타일 편집 예

요류를 보고해 주는 결과와 간단한 스타일 편집기능의 예를 보이고 있다.

4.1.3 흐름 객체 트리 구조

흐름 객체 트리 구조창은 흐름 객체 트리 생성기에서 생성한 흐름 객체 트리의 모습을 보여주는 것으로 그림 14에 결과를 보여준다.

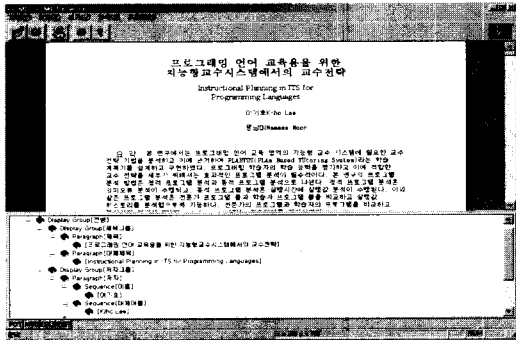


그림 14 흐름 객체 트리창 처리 결과

흐름 객체 트리 구조창의 각 노드는 흐름 객체 클래스 이름과 엘리먼트 이름으로 구성된다. 흐름 객체가 내용 노드를 자식으로 가지고 있으면, "[ ]" 기호 안에 내용을 보여준다. 또한 해당 흐름 객체 트리 창에서 노드를 더블 클릭하면 처리된 디스플레이 위치로 이동하는 기능을 두어 디스플레이 영역으로 이동하여 노드로의 빠른 이동과 지정한 특징값의 포맷 여부를 확인 할 수 있는 방법을 제공한다.

4.2 고찰

본 논문은 SGML 문서를 DSSSL에 기반하여 포맷팅 처리하여 표현하기 위한 SGML 문서 표현 시스템 설계와 구현에 관한 것이다.

본 논문에서 개발한 시스템은 기존의 DSSSL 처리 엔진인 Jade 엔진에[11] 기본하여 다른 프로그램에 종속되지 않고 포맷팅 결과를 표현할 수 있도록 백엔드(backend)를 생성하도록 포맷터를 설계하여 브라우징할 수 있도록 하고 있다. 이를 위해 SGML 파서와 DSSSL 처리기의 인터페이스로 GROVE를 사용하였으며, DSSSL 처리기와 포맷터의 인터페이스로는 흐름 객체 트리를 사용하였다. 이렇게 생성된 흐름 객체 트리는 포맷터 처리에 의해 화면에 디스플레이 하도록 하였다.

본 시스템은 한글 엘리먼트와 속성 등 SGML 문서뿐만 아니라 DSSSL 문서에서 한글 사용이 가능하다. 또한 ISO 10179의 DSSSL 표준과 SGML 브라우저 및 편집기를 위한 포맷팅 명세인 DSSSL-O(Online) 프로

표 2 흐름 객체 구현 난이도

흐름 객체 클래스	구현 난이도
sideline	상
leader	상
rule	상
link	상
box	중
marginalia	중
character	중
scroll	하

파일에서 정의한 스타일 처리 과정에 적합하게 시스템을 설계하였으며, 47개의 흐름 객체 클래스 중에서 가장 많이 사용되는 9개의 흐름 객체 클래스를 구현하였다. 그러나 DSSSL-O에서 sideline, character, leader, rule, box, scroll, link, marginalia 등이 아직 구현되지 않아 향후에 더욱더 연구되어야 할 것인데, 이들 흐름 객체 클래스들의 추가 구현 시에 구현 난이도를 상, 중, 하로 나타내어 표 2에 보였다. 그밖에 다른 흐름 객체 클래스들은 화면 출력을 위한 흐름 객체가 아니거나 선택사항으로 존재할 수 있으므로 표시하지 않았다.

현재 국내외적으로 많은 SGML 문서의 포맷팅 기능을 갖는 편집기들이 존재하지만, DSSSL을 지원하는 시스템은 거의 전무하며 단락, 그래픽과 테이블 처리에 문제점이 노출되고 있는 실정이다. 국외의 경우 DSSSL 처리 시스템으로 Jade, YADE, DSSSL Developer's Toolkit, Kawa, Psgml, panodssl, Jadetex Package 등이 있으나[11] 한글 처리 등의 문제가 있고, 대부분 포맷팅을 위한 결과를 RTF, TeX, XML FOT, MIF, SGML 변환으로의 백엔드로 변환되기 때문에 다른 응용 프로그램에 종속되고 표준에 입각하여 처리되지 않는 제한적인 표현만이 가능하나, 본 시스템에서는 백엔드로 본 시스템의 포맷터를 설계하여 브라우징하므로 다른 응용 프로그램에 종속되지 않고 표준에 따라 처리된다. 국내의 경우 연세대에서 개발한 SGML 편집 시스템은 아직 프로토타입 수준이지만 국내 첫 시스템이라는 점에서 의미가 있다고 사료된다[12]. 이들 시스템에 대한 비교를 표 3에 보인다.

본 시스템은 한글 처리뿐만 아니라 단락과 그래픽, 테이블에 대한 처리가 가능하지만, 완전한 SGML 문서를 표현하고 처리하기 위해서는 sideline, character, leader, rule, box, scroll, link, marginalia와 같은 흐름 객체 클래스들의 처리가 향후에 이루어져야 할 것이다. 그러나 일반적으로 많이 사용하고 있는 흐름 객체 클래스들의 처리가 가능하고 선택사항으로 요구되는 미 처



표 3 기존 시스템들간의 비교

		장 점	단 점	비고
국외	HyBrick	• SGML,XML 문서지원	• 그래픽과 테이블 처리 미흡 • 한글과 같은 멀티 바이트 코드에 대한 처리 미흡 • 자체 처리 엔진으로 타 시스템에의 이용 불가	• JADE 기반 • DSSSL 지원 • 자체표현엔진
	Panorama	• Link 처리 • 향상된 그래픽 처리	• 제한적인 표현만이 가능 • HTML의 Link로 표현	• CSS 지원
국내	YesSGML	• 빠른 처리 • 향상된 사용자 인터페이스	• 그래픽 정렬과 테이블 처리 미흡 • 지원 흐름 객체 클래스 미비	• JADE 기반 • DSSSL 지원
	연세대	• 자체 개발 파서로 최적화	• 테이블 처리 등이 미흡 • 지원 흐름 객체 클래스 미비	• DSSSL 지원
	본 시스템	• 다른 응용시스템에서 이용 가능 • 향상된 테이블 처리 • 멀티 바이트 코드 환경 제공 • 표준 인터페이스인 GROVE 지원	• sideline, character, leader, rule, box, scroll, link, marginalia 등의 흐름 객체 클래스 미 처리	• JADE기반-Backend • DSSSL 지원

리 흐름 객체 클래스를 제외하여도 DSSSL 엔진으로 실제 사용하는데 손색이 없다고 생각된다.

또한 본 연구실에서 개발한 학회논문 DTD를 이용하여 개발된 SGML 문서들을 사용해 검증하였으며, 또한 학회논문 DTD에 대한 DSSSL 스타일 시트를 개발하였다.

**5. 결론**

본 논문은 SGML 문서와 DSSSL 문서를 처리하여 사용자에게 온라인으로 보여 줄 수 있는 표현 시스템 개발에 관한 것이다.

본 시스템은 기존의 시스템인 Jade 엔진을 바탕으로 다른 응용 프로그램에 종속되지 않고 표준에 입각하여 본 시스템의 포맷터를 설계하여 곧바로 브라우징 할 수 있도록 백엔드를 생성하였고, 한글 처리뿐만 아니라 단락과 그래픽, 테이블에 대한 처리가 가능하고, DSSSL 스타일 시트를 처리 가능하며, SGML 파서와 DSSSL 처리기의 표준 인터페이스인 GROVE를 지원한다. 또한 간단한 스타일 정보를 WYSIWYG 방식으로 표현이 가능하다.

향후 연구에서는 보다 나은 포매팅 처리를 위해 수학적 및 멀티 컬럼과 같은 흐름 객체 클래스들을 더 지원할 수 있도록 해야 하며, SGML 편집기와 같은 다른 응용 프로그램에서 본 시스템을 사용할 수 있도록 모듈화 할 계획이다. 또한 스타일 시트 편집기를 개발하여 스타일 시트 작성시 발생할 수 있는 오류를 방지하고, 온라인으로 파싱 및 포매팅을 할 수 있도록 할 계획이다.

본 논문은 국내에서 컴퓨터를 이용한 SGML 문서처리 환경 구축에 크게 기여할 뿐만 아니라, SGML에 기반한 초고속정보망에서 문서처리, 저장, 검색을 위한 기본 응용 개발을 촉진하고 멀티미디어 문서처리 시스템 기술 개발에 중요한 역할을 하리라 본다.

(감사의 글 : 본 논문은 1999년 한국과학재단의 특정 기초 연구과제 및 ETRI 위탁과제 연구비에 의해 연구되었음)

**참고 문헌**

- [1] "SGML 가이드" 정희경, 현득창, 이수연 공저 사이버출판사
- [2] ISO/IEC 8879 Standard Generalized Markup Language(SGML)
- [3] ISO/IEC 10179 Document Style Semantics and Specification Language (DSSSL)
- [4] "문서 스타일 의미 지시 언어 표준개발에 관한 연구 보고서", 배재대학교, 정희경, 1997
- [5] <http://sunsite.unc.edu/pub/sun-info/standards/dsssl/dsssl0/d0960816.htm> "DSSSL-online Application Profile" by Jon Bosak
- [6] <http://itrc.uwaterloo.ca/papresco/dsssl/tutorial.html> "Introduction to DSSSL" by Paul Prescod
- [7] <http://www.oasis-open.org/cover/GROVESKimber1.html> "GROVES: an illustrated example" by Eliot Kimber
- [8] <http://nwalsh.com/docbook/index.html> DocBook Projects by Norman Walsh
- [9] ISO/IEC 10744 Hypermedia/Time-based Structuring Language(HyTime) 2d Edition
- [10] "Semantic Extensions to DSSSL to Handle Tree"

by Matthew Fuchs, GCA SGML'96

- [11] <http://www.jclark.com/jade/> "James' DSSSL Engine" by James Clark
- [12] 이경호 외 5인, "DSSSL에 기반한 SGML 문서편집기의 설계 및 구현", 정보과학회논문지(C), 제4권, 제 6 호, pp. 807-816, 1998.12
- [13] "ABCs of DSSSL" by Sharon Adler, Anders Berglund, James Clark SGML'95
- [14] "practical SGML" Second Edition by Eric van Herwijnen NICE Technologies Veraz, France
- [15] "ParseMe 1<sup>st</sup>" Sean McGrath Prentice Hall
- [16] "Using SGML" by Patrick Henry Winston, Berthoold Klaus Paul Horn, ADDISON WESLEY



김 창 수

1996년 배재대학교 전자계산학과(학사).  
1998년 배재대학교 전자계산학과(석사).  
1999년 ~ 현재 배재대학교 컴퓨터공학과(박사과정). 관심분야는 멀티미디어 문서정보처리, SGML, XML, XML/EDI, XSLT.



정 회 경

1985년 광운대학교 컴퓨터공학과(학사).  
1987년 광운대학교 컴퓨터공학과(석사).  
1993년 광운대학교 컴퓨터공학과(박사).  
1994년 ~ 현재 배재대학교 컴퓨터공학과 교수. 관심분야는 하이퍼미디어/멀티미디어 문서정보처리, SGML/XML,

HyTime, IETM, DSS SL/XSL, XML/ EDI



윤 보 현

1992년 목포대학교 전산통계학과 (학사).  
1995년 고려대학교 컴퓨터학과 (석사).  
1999년 고려대학교 컴퓨터학과 (박사).  
1999년 ~ 현재 한국전자통신연구원 선임연구원 문서정보연구팀. 관심분야는 정보 검색, 자연언어 처리, XMI/SGML,

지식정보 처리



강 현 규

1985년 홍익대학교 전자계산학과 (학사).  
1987년 한국과학기술원 전산학과 (석사).  
1992년 정보처리 기술사 자격 취득. 1997년 한국과학기술원 전산학과 (박사).  
1987년 ~ 현재 한국전자통신연구원 책임연구원 문서정보연구팀장. 관심분야는

정보 검색, 자연언어 처리, XML/SGML, 지식정보 처리