정보학연구 제2권 제2호 1999년 11월(pp.169-279)
The Studies of Information Technology, Vol.2, No.2

GROVE를 이용한 SGML 문서 저장 관리 시스템 설계

정회경/안성옥*/오일덕**

요 약

정보화 사회에서 많은 문서가 전자화 됨에 따라 효율적인 처리를 위해 구조화된 전자 문서 처리가 요구되고 있다. 이에 SGML은 구조화된 정보를 생성하고 교환하기 위한 문서 표준으로써, 이러한 전자 문서를 보여주고 수정하며 새로운 문서를 생성하기에 알맞다. 이에 따라 대량의 구조화된 SGML 문서 정보의 저장, 관리에 관한 연구가 필요하다.

본 논문은 HyTime(Hypermedia Time-based Structuring Language)에서 정의된 GROVE(Graph Representation Of property ValuEs)를 이용하여 데이터 모델링 설계 및 SGML 문서 저장 관리 시스템 설계에 대해 기술한다.

Ⅰ. 서론

정보의 홍수시대에서 많은 양의 전자 문서 정보를 효율적으로 관리하거나 구축해야 하는 요구가 발생하면서 대량의 전자문서 정보를 저장, 검색, 관리해야 하는 필요성이 증가되고 있다. 특히 전자도서관(Digital library), CSCW(Computer Supported Coorperative Work), CALS(Commerce At the Light Speed) 등에서 이러한 요구가 많다. 이러한 요구에 따라 표준화된 SGML(Standard Generalized Markup Language)[1,11,14]은 구조화된 문서 정보를 체계적으로 생성하고, 전송하기 위한 문서 표준으로서 이러한 전자 문서들의 관리작업들이 가능하게 한다.

본 논문은 SGML 문서를 데이터베이스에 효율적으로 저장 및 관리하기 위한 방법을 설계하고 이에 대한 SGML 문서 저장 시스템에 대해 기술한다.

기존의 SGML 문서 데이터 모델링은 엘리먼트(element), 엔티티(entity), 애트리뷰트(attribute)의 관계만을 정의하는 연구에 중점을 두었다[2][3][4]. 그러나, 본 논문에서 설계한 모델링의 기본 개념은 HyTime에서 정의한 GROVE 개념을 이용한다. GROVE는 문서를 파싱하는데 있어 메모리내의 결과로서 노드들로 구성된 트리이며, 각 노드는 프로퍼티(property)들의 집합으로 구성된다.

본 논문에서 GROVE를 이용하여 SGML 문서의 구조 정보를 모델링함으로써 DTD와 문서 인스턴스의 데이터 정보를 표현하고, 이들 사이의 관계뿐만 아니라 엘리먼트, 엔티티, 애트리뷰트 등의 각 노드에 대한 프로퍼티들을 정의해 준다. 프로퍼티 집합은 노드나 클래스를 설명해 주는 역할을 하기 때문에, SGML 문서의 검색

* 배재대학교 컴퓨터공학과 조교수/부교수
** 대전산업대학교 전자공학과 교수
이 논문은 1998년 한국과학재단의 특정기초 연구과제 연구비에 의하여 연구되었음.

시스템에서 색인에 대한 범위가 넓어지고, 문서의 세세한 구성요소들을 이용해 질의어 작성이 가능하므로 SGML 문서관리가 효율적으로 이루어질 수 있다.

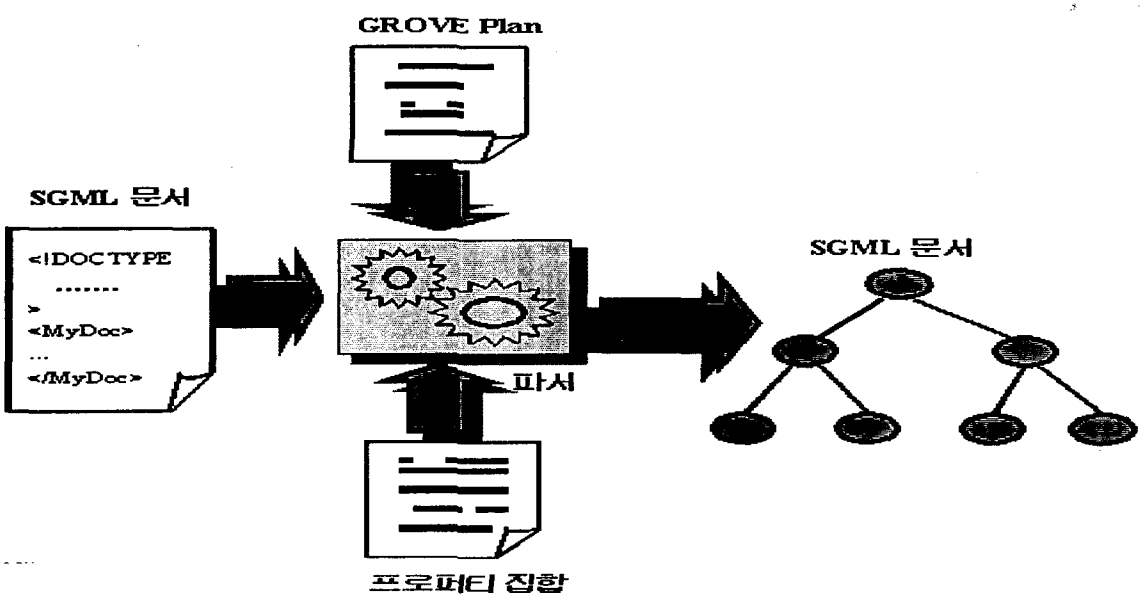
GROVE의 각 노드들은 근본적인 객체 지향 데이터 모델이므로, SGML 문서의 구조 정보를 저장하기 위해 OODBMS(Object-Oriented Data-Base Management System)를 채택한다.

본 논문의 구성은 다음과 같다. 2장은 GROVE의 개략적인 설명, 즉 프로퍼티 집합을 이용하여 GROVE가 구성되는 원리와 이들 사이의 관계를 기술한다. 3장은 GROVE를 이용한 SGML 문서의 데이터 모델링에 대해 기술하고, 4장은 SGML 저장 관리 시스템의 구조와 기능 설계에 대해 기술한다. 5장에서는 결론 및 향후 연구 과제에 대해 알아본다.

II. GROVE와 프로퍼티 집합의 기본개념

본 장에서는 SGML 데이터 모델링의 기반이 되는 GROVE[9,10]와 GROVE를 구성하는 프로퍼티 집합[8,12,13]에 대해 기술한다.

모든 처리 시스템들은 이들이 사용하는 데이터를 그 시스템만이 표현할 수 있는 구조로 메모리에 생성하여야 한다. 실제로 이들 표현은 배열이나 관계 테이블, 객체 등과 같이 서로 다른 형태를 갖으며, HyTime이 분산되고 개방된 네트워킹 환경으로 발전하는 것처럼, SGML과 그 관계 표준들은 여러 가지의 도구들과 시스템들의 상호작용을 가능하게 한다. 그래서, 이들 표준은 메모리 내의 데이터 구조들을 정의하고



(그림 1) SGML 문서 GROVE의 생성 과정

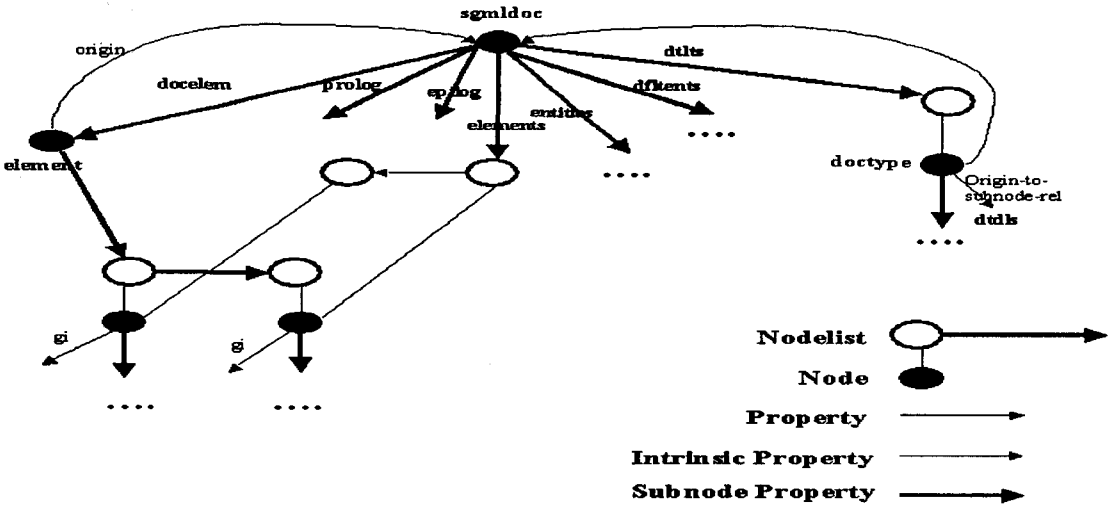
참조하기 위해 일반적이고 독립적인 형식을 필요로 한다.

이 정의의 첫번째 부분이 프로퍼티 집합이고, 두번째 부분이 GROVE이다. GROVE는 SGML 문서를 파싱할 때 메모리 내에 생성되는 트리 표현이고, 프로퍼티 집합은 GROVE 각 노드를 특성화시킨다. GROVE는 객체들을 처리하는 동안 프로퍼티 집합이 어떻게 정의되는지와 서로 다른 GROVE들이 어떻게 관련되는지에 대해 정의한다. 특히 프로퍼티 집합은 주소가 정확하게 지정되므로, GROVE에서는 규칙적이고 예측할 수 있는 데이터 구조를 정의한다. 그림 1에 SGML문서 GROVE의 생성 과정을 보인다[9]. 여기서 프로퍼티 집합으로부터 하나 이상의 GROVE plan을 정의하고, GROVE plan은 GROVE 객체들이 어떻게 도출되는지를 설명한다. GORVE plan은 GROVE에 포함될 수 있거나 포함될 수 없는 객체 형태가 무엇인지를 나타낸다. 또한 모든 프로퍼티 집합에 대한 모든 객체들과 프로퍼티를 포함하는 절대적이고 완전한 GROVE plan도 존재한다.

본 논문에서는 데이터베이스 저장소의 내부 모델로 GROVE를 이용하여 SGML 문서를 모델링하였다. GROVE 형태의 문서 모델은 SGML 문서 관리 시스템의 저장 모델로 효과적이고 여러 가지 응용들 사이에서 훌륭한 API를 제공해 준다.

Ⅲ. SGML 문서의 데이터 모델링 설계

SGML 문서를 저장하기 위해서는 SGML 문서의 논리적 구조를 데이터베이스에 표현하기 위한 논리적 구조 모델링이 필요하다.



(그림 2) SGMLDOC 모델

본 장에서는 효율적인 저장을 위해 GROVE를 이용해 설계한 SGML 문서의 데이터 모델링을 제시하고 설명한다.

3.1 SGML 문서의 논리적 구조 모델링 설계

SGML 문서가 포함하고 있는 데이터 및 특성들을 데이터베이스에 손실 없이 표현하고, 이를 기반으로 문서 관리를 효율적으로 하기 위해서는 SGML 문서의 논리적 구조 모델링이 필요하다.

본 논문의 모델링은 DTD와 DI(Document Instance)를 표현하기 위해 설계된 DOCTYPE 모델과 DOCELEM 모델로 구성되어 있고, SGML 문서에서 ENTITY는 중요한 부분을 차지하고 있기 때문에, 이에 대한 설계 모델을 기술한다. 또한 SGML 문서 모델의 전체 구조를 파악하기 위해 SGMLDOC 모델을 그림 2에 보이고 있다.

SGMLDOC 모델은 DOCELEM 모델, PROLOG 모델, EPILOG 모델, ELEMENTS 모델, ENTITIES 모델, DFLTENTS 모델, DTLTS 모델로 구성되어 있다. DOCELEM 모델은 문서 인스턴스에 대한 실제적인 모델링이고, DTLTS 모델은 SGML 문서의 문서 형태(document type)와 링크 형태(link type)가 어떻게 구성되어 있는지 설명해 주는 모델이다. ELEMENTS와 ENTITIES 모델은 실제 문서 인스턴스에서 사용된 모든 엘리먼트들과 엔티티들을 기술해 주는 모델로서 문서내 모든 엘리먼트와 엔티티에 연결되고, DFLTENTS 모델은 디폴트 엔티티들을 기술해 주어 디폴트 엔티티가 쓰이는 부분과 연결된다. PROLOG와 EPILOG 모델은 SGML 문서에 대한 프롤로그와 에필로그 이고, SGML-DOC에 대한 프로퍼티 성격이 강하므로, 본 논문에서 이에 대한 모델은 제외시켰다. 또한 DTLTS 모델의 링크 형태에 대한 모델링도 향후 HyTime에서 자세히 다루어야 할 문제로 사료되어 본 논문에서는 제외시켰다.

SGML 문서는 DTD와 DI로 구성되므로, 본 모델링에서는 SGMLDOC 루트 노드 아래 DTD를 DOCTYPE 모델로, DI를 DOCELEM 모델로 표현함으로써 두 가지 구성요소를 모두 포함한다.

DOCTYPE과 DOCELEM 모델에는 구체적인 차이가 있다. 두 가지 모델은 SGML 문서에 대한 모델링이기 때문에 엘리먼트, 엔티티, 애트리뷰트 등에 대한 모델링을 포함하고 있지만, DOCTYPE 모델은 SGML 문서의 형태가 어떻게 이루어졌는지, 즉 엘리먼트 형태들이 어떻게 기술되는가, 시작태그가 기술되는가 생략되는가 또는 엘리먼트의 내용 부분이 어떻게 이루어졌는가 등에 관해 파악할 수 있는 모델인 반면, DOCELEM 모델은 실제적으로 SGML 문서 인스턴스의 값이 어떤 것인지 파악할 수 있는 모델이다. 즉, 엘리먼트의 실제 값이 문자 값인지 비 SGML 형태 값인지 또는 외부 엔티티 값인지 등을 기술해 준다.

3.1.1. DOCTYPE 모델링

DOCTYPE 모델은 SGML 문서의 형태론적인 모델이다. DOCELEM 모델이 실제 저장되는 데이터들의 형태 값을 모델링한 것인 반면, DOCTYPE 모델은 SGML의 문서 형태가 어떻게 이루어졌는지에 관한 모델링이다. 즉, SGML에서 사용되는 문법들이 이 모델링의 주류를 이룬다. 이 모델은 DOCELEM 모델과 마찬가지로 엘리먼트와 엔티티, 애트리뷰트가 기술되지만,

그들이 어떤 형태를 가지고 있는지 나타낸다.

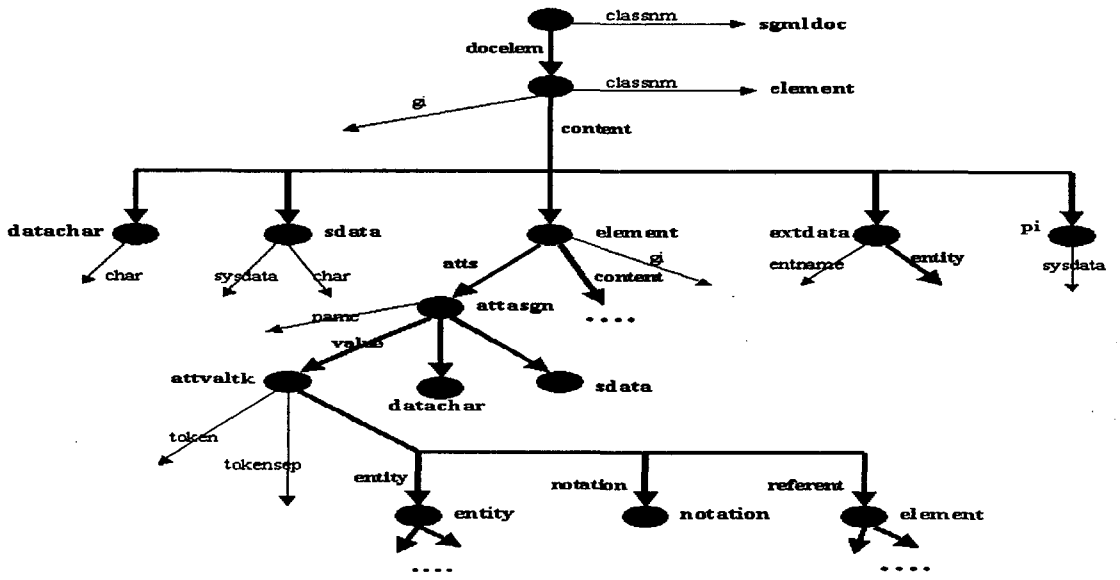
엘리먼트 클래스는 클래스 이름으로 elemtype을 갖고, 이에 대한 프로퍼티들로 gi, 엘리먼트의 태그(tag)가 생략되었는지 아닌지에 대한 여부인 omitstrt(omit start tag), omitend(omit end tag), 또 엘리먼트의 내용 모델이 무엇으로 구성되어 있는지에 관한 contype(content type)이 있다. contype이 modelgrp(model group)인 경우, 모델 그룹안에 있는 토큰별로 엘리먼트나 pcdata 토큰 노드에 토큰들을 저장하고, 이 모델 그룹에 나타나는 발생 지시자(occurrence indicator)를 기술한다.

엘리먼트가 애트리뷰트를 갖는 경우, 그 애트리뷰트의 이름과 토큰들을 기술하고 선언되는 값의 형태와 디폴트 값의 형태를 기술하는데, 이 때 디폴트 값이 CURRENT일 때 curgrp (current group)과 curattix(current attribute index)를 포함한다. 애트리뷰트 정의가 동일하고 curattix 프로퍼티 값이 같으면 애트리뷰트들은 동일한 CURRENT 값을 공유한다.

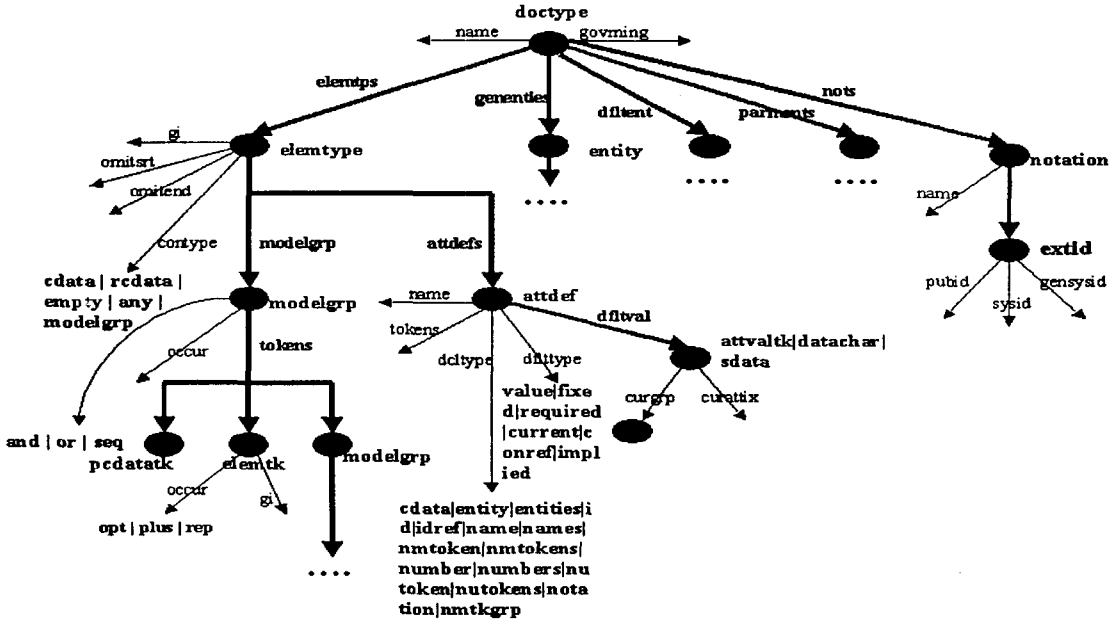
그 밖에 DOCTYPE 모델에서는 엔티티들의 종류에 따라 각각의 클래스를 만들고, 표기법(notation)에 대한 클래스를 포함하여 이에 대해 기술한다.

3.1.2. DOCELEM 모델링

DOCELEM 모델은 엘리먼트 모델에서 시작된다. DOCELEM는 문서 형태를 대표하는 문서 엘리먼트라 할 수 있고, DOCELEM로 지정되는 엘리먼트가 DOCELEM 모델의 최상위 클래스이다. 이 클래스는 classnm과 gi 프로퍼티로 식별되고, 여기서 classnm은 모든 클래스에 대해 공통으로 사용되는 클래스명이며, gi는 인스턴스에 명명된 엘리먼트 이름에 관한 일반 식별자이다.



(그림 3) DOCTYPE 모델



(그림 4) DOCELEM 모델

파서가 식별하는 엘리먼트의 순서는 같은 형제 노드에서 왼쪽에서 오른쪽의 순서이다.

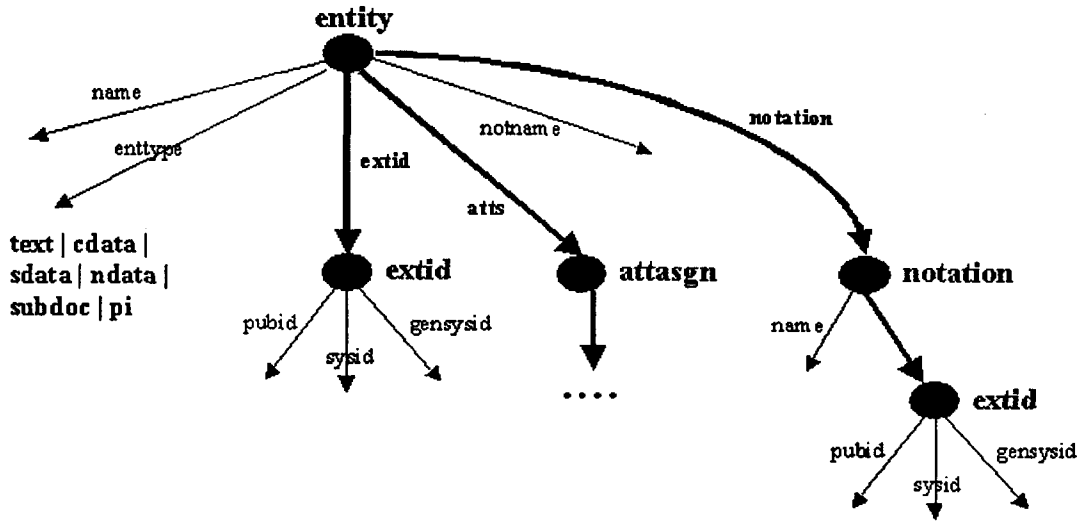
엘리먼트의 내용 모델은 문자 데이터(datachar), 특정 데이터(sdata), 다른 엘리먼트, 외부 데이터(extdata), 그리고 처리 명령(pi)이 온다. 내용 모델에 다른 엘리먼트가 이어질 경우, 엘리먼트의 내용 모델에는 또 다른 엘리먼트가 올 수 있고, 그 엘리먼트에 대한 애트리뷰트가 있을 수도 있다. 이 모델에서는 데이터 애트리뷰트 때문에 attasgn 클래스를 두어 여기에 애트리뷰트 이름을 명기하고, 그 애트리뷰트 값이 토큰 값이라면 attvaltk 클래스에, 다른 데이터 애트리뷰트는 datachar 클래스나 sdata 클래스에 각각 지정한다.

3.1.3. 엔티티 모델링

엔티티 모델은 doctype 모델과 docelem 모델에서와 같은 방식을 갖지만, docelem 모델에서는 extdata(external data) 클래스의 하위 클래스로 존재한다. 이것은 docelem 모델이 실제 데이터에 대한 모델링이기 때문이다.

Ⅳ. SGML 문서 관리 시스템 설계

SGML의 큰 특징은 문서의 논리적 구조를 갖는다는 것이다. 이 특징은 SGML 문서의 논리 구조와 저장 구조를 완전히 분리할 수 있게 한다. 즉, 문서의 저장 체제가 문서의 논리 구조에 반드시 영향을 주지 않고도 최적화 될 수 있다



(그림 5) 엔티티 모델

는 것을 의미한다.

본 논문에서는 이 특징을 이용하여 엔티티 계층을 두어 추상적인 저장 모델을 설계하였다.

4.1. SGML 문서 관리 시스템의 구조

SGML 문서들은 엘리먼트와 엔티티들의 트리 구조이므로, SGML 문서 관리 시스템들은 시스템의 작업 단위로써 완전한 엘리먼트와 엔티티 트리 또는 이들 트리의 일부분을 가지고 작업할 수 있는 기능을 제공한다. 본 논문의 문서 관리 시스템은 그림 6.과 같이 4 가지 추상 계층으로 SGML 문서를 관리한다.

가장 낮은 단계는 SGML 데이터 스트림(stream)으로 이는 상위의 추상 단계 엘리먼트 계층과 엔티티 계층에서 SGML 파서가 해석하는 문자들의 스트림이다. 엘리먼트 계층은 SGML 엘리먼트 뿐만 아니라 하이퍼링크(hyperlink) 관계를 표현하는 다른 계층일 수도 있다.

SGML 문서 관리 시스템에서 각 계층의 기능은 다음과 같다.

n 데이터 스트림 계층

Ø 파일 시스템이나 데이터베이스 시스템과 같은 저장 시스템에 SGML 데이터 스트림이 저장 관리되도록 하는 계층

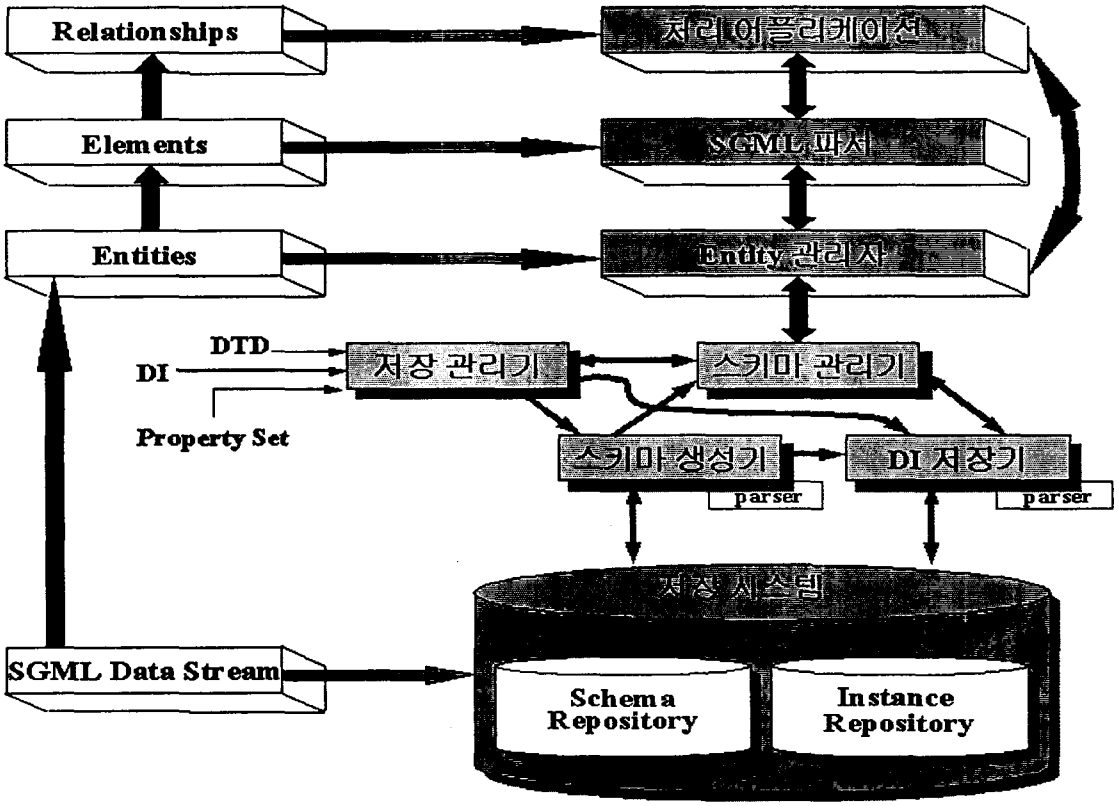
Ø 문자들의 스트림이 SGML 파서에 의해 해석된다

n 엔티티 관리자 계층

Ø 엔티티들을 관리

Ø SGML 문서 엔티티를 시작으로 엔티티들을 해석하고 엔티티 선언과 참조를 인식하는 파서에게 SGML 데이터 스트림을 전달

Ø 엔티티를 해석한 정보(엔티티 ID)를 가지고



(그림 6) SGML 문서 관리 시스템의 추상 계층

실제 저장 객체들로 엔티티 참조들을 분해하기 위해 저장 시스템과 통신

Ø SGML 파서에 의해 파싱된 파싱 정보를 받아 실제 저장 객체와 맵핑한 후 다시 파서에게 이 객체를 전달

n SGML 파서(엘리먼트 계층)

Ø 엔티티 관리자에게 SGML 데이터 스트림을 전달받아 이 데이터를 파싱

Ø 엔티티 선언과 참조를 인식하여 엔티티 관리자에게 전달

Ø 편집기, 브라우저, 검색 시스템 등의 처리 응용들에게 엘리먼트들의 표현, 데이터 내용, 그리고 다른 SGML에서 정의된 구조 등을 전달하는 SGML 엘리먼트 마크업을 인식

n 처리 응용(관계 계층)

Ø SGML 파서에 의해 분석된 엘리먼트들과 그들의 데이터를 해석하여 처리 응용의 목적에 맞게 사용

SGML 문서 관리 시스템에서 SGML 데이터 스트림 계층의 저장 시스템의 기능은 다음과 같다.

n 저장 관리기

새로운 스키마 생성, SGML 인스턴스 저장, DTD나 인스턴스에 대한 정보 반환 등의 기능을 수행할 때 발생하는 부수적인 수행에 대한 투명성을 제공

n 스키마 관리기

Ø SGML DTD, SGML 인스턴스, 분산된 저장소에 대한 포괄적인 정보를 관리. 즉, 특정 스키마에 대한 저장소가 어디에 있는지, 특정 인스턴스가 어떤 저장소에 있는지 등에 대한 정보를 관리

Ø 각 DTD의 스키마 이름, 각 엘리먼트에 대응되는 클래스 이름, 저장소 ID 등을 할당, 관리

Ø 새로운 DTD에 대한 스키마를 생성하려 할 때, SGML 문서 관리 시스템에 이미 존재하는지의 여부를 판단

Ø 색인되지 않은 인스턴스에 대한 정보 관리

n 스키마 생성기

Ø 입력된 DTD를 파싱해서 스키마 생성에 필요한 정보를 추출하여 스키마 생성

Ø 생성된 스키마에 대한 정보를 스키마 관리기에 등록

Ø DBMS에 의존적

n DI 저장기

Ø SGML 인스턴스를 미리 생성되어 있는 DBMS의 스키마에 복합객체 형태로 저장

Ø 인스턴스 저장기가 DBMS에 객체 생성시 객체 식별자(Object ID) 부여

Ø 인스턴스에 대한 정보를 스키마 관리기에 등록

V. 결론

일반적으로 SGML 문서의 데이터 모델링은 SGML 문서의 효율적인 저장과 관리에 기본이 되는 것으로, 저장소로 사용할 데이터베이스의 특성과 기능에 따라 나누어진다.

지금까지 제시되어 온 모델링을 보면 관계 모델, 확장된 관계 모델, 복합 객체 모델, 객체 기반 모델, 엘리먼트 기반 모델 등으로 나눌 수 있는데, 본 논문은 객체 기반 모델에 해당한다.

본 논문에서는 HyTime에서 정의된 GROVE 개념을 이용해 SGML 문서 구조 정보를 DTD에 대한 정보를 유지하기 위한 DOCTYPE 모델과 인스턴스에 대한 정보를 유지하는 DOCELEM 모델로 크게 나누어 설계하였으므로, 기존의 모델과 비교하여 DTD에 대한 형태와 논리적 구조를 연계시켜 인스턴스에 대한 정보를 효과적으로 관리할 수 있다. 또한, 본 논문에서는 SGML 문서를 효율적으로 저장하고 관리할 수 있는 SGML 저장 관리 시스템을 설계하여 DBMS에서 제공하는 다양한 기능을 기반으로 대량의 SGML 문서 처리 및 공유와 부분적인 SGML 객체의 추출 및 관리가 가능하다.

따라서, 본 연구 결과는 CALS 및 전자 도서관 등의 데이터 저장 시스템 개발에 사용될 수 있으리라 생각된다.

이를 바탕으로 향후 연구 방향은 본 논문의 모델에서 제외시킨 링크 형태에 대한 모델링과

HyTime, DSSSL, XML(eXtensible Markup Language) 등 다양한 구조화된 문서들에 대한 데이터 모델링, 특히 HyTime의 링크에 대한 모델링 등에 대한 연구가 요구되며, 설계된 내용의 구현에 대한 연구도 진행되어야 될 것이다.

참고문헌

[1] 정회경, 현득창, 이수연, SGML 가이드, 사이버 출판사, 1997

[2] 이원석, 대량의 구조화 문서 관리를 위한 SGML 저장 관리기의 설계 및 구현, 충남대학교, 1998

[3] Patricia Francois, Generalized SGML repositories : Requirements and modelling, Computer Standards & Interfaces, 1996.

[4] Extending the scope of document handling : The design of an OODBMS application framework for SGML document storage. Publication (GMD-IPSI). K.Bohm, K. Aberer and C. Huser (Dec.1993)

[5] K.K. Bohm and C. Huser, The Prospects of publishing using advanced database concepts, Electronic Publishing 6(4) (1993) 469-480.

[6] V.Christophides, S.Abitetoul, S.Cluet and M.Scholl. From structured documents to novel query facilities, 13rd ACM SIGMOD Conf. (May 1994).

[7] W. Eliot Kimber, SGML Document Management, ISOgen International Corp., 1995

[8] W. Eliot Kimber, ISOGEN DSSSL Specification Architecture(Dslspec), http://www.isogen.com/demos/dslspec/dslspec.html

[9] W. Eliot Kimber, An Excerpt from Practical Hypermedia : An Introduction to HyTime : Property sets and GROVEs, 3rd international HyTime conference, http://www.hightext.com/IHC96/ek8.html

[10] W. Eliot Kimber, Managing SGML Architectures and Object Models with GROVEs, W. Eliot Kimber and ISOGEN International Corp., 1997, http://www.isogen.com/papers/GROVEmng/GROVEmng.html

[11] International Organization for Standarization.Information processing text and office system standard generalized markup language(SGML), 1986, ISO/IEC 8879 : 1986.

[12] International Organization for Standarization. Information technology Processing Language - Document Style Semantics and Specification(DSSSL), 1996, ISO/IEC 10179 : 1996.

[13] International Organization for Standarization. ISO/IEC 10744 Hypermedia/Time-based Structured Language(HyTime) 2nd Edition, annex, 1992, ISO/IEC 10744 : 1992.

[14] Eric van Herwijen, *Practical SGML Second Edition*, NICE Technologies Varaz, France.

[15] Jasmine Online Document, Computer Associates International, Inc. & Fujitsu LIMITED, 1996-1998, Japan.

Design of SGML Document Storage Management System using GROVE

Hoe-Kyung, Jung*/Sung-Ok, Ahn*/Eel-Deok, Oh**

Abstract

SGML(Standard Generalized Markup Language) is proper to view, modify and create new electronic document as documentation standard to create and interchange the structured document information. Accordingly, a study on efficient storage and management of very large structured SGML document information is need.

This paper proposes design of data modeling based on GROVE(Graph Representation Of property ValuEs) defined in HyTime(Hypermedia Time-based Structuring Language) and describes design of SGML document storage management system.

* Dept. of Computer Eng., Paichai Univ.
** Dept. of Elec. Eng., Taejon Nat'l Univ. of Tech.