DTD/SGML 문서 저작 도구의 설계 및 구현

현 득 창† · 이 수 연††

요 약

본 연구는 인터액티브한 방식을 사용하여 문서의 구조를 나타내는 문서 형 정의(DTD)를 저작할 수 있고, 동시에 문서 형 정의에 기반한 범용적인 SGML 문서는 물론 HTML문서를 편집할 수 있는 한글 SGML 문서와 문서 형 정의 저작 도구의 설계 및 구현에 관한 것이다. HTML의 문서 형 정의는 SGML의 구문에 따라 정의된 것으로 그에 따르는 HTML 문서는 SGML의 한 응용이다. 그러므로 HTML의 어떤 버젼의 문서도 그에 대응하는 문서 형 정의와 본 개발 저작 도구만 있으면 저작이 가능하다. 본 시스템은 그래픽 사용자 인터페이스로는 X 윈도우 시스템의 Motif와 UIL을 사용하였고, 그 밖에 기능 모듈은 C-언어를 이용하여 구현하였다.

The Design and Implementation of an Editor Composing DTD and SGML Document

Deuk Chang Hyun† · Soo Youn Lee††

ABSTRACT

This study addresses the design and implementation of Korean SGML(Standard Generalized Markup Language) editor capable of generating DTD(Document Type Definition)s, which can be used at the same time to generate HTML documents as well as SGML documents using interactive method. HTML is an application of SGML and HTML DTD is defined according to the syntax of SGML. Therefore it is possible to generate HTML documents of any versions by replacing the corresponding DTDs and using the implemented editor. This system has been implemented using GUI such as Motif and UIL(User Interface Language) in X-window system and C-language for common modules of functions.

1. 서 론

현재 고도화된 정보화 사회가 빠른 속도로 이루어지고 있다. 이러한 고도의 정보화 작업에 있어서 가장 먼저 선행해야 하는 필수적인 작업이 기존 및 앞으로 발생되는 문서 정보를 전자 문서로 만들고 관리하는 일이다[1, 2]. 이러한 측면이 가장 극명하게 나타나는 것이 공공기관의 각종 서류 및 자료들, 또한 기업에서 처리되는 각종 서류 등이 지금까지는 하드카피(hardcopy)된 문서들로 서로 교환하거나 보관하였기에 그 시간적, 비용적인 손실이 매우 컸다. 따라서 각 공공기관이나 기업에서는 이러한 비효율성을 감소시키고자 효율적이고 비용 절감할 수 있는 방안으로 모든 문서 처리의 전산화를 서둘러 실시하고 있는 실정이다.

미국에서도 이러한 일환으로 미 국방성이 군사 지

※본 연구는 97년도 교내연구비 지원으로 수행된 과제임.
† 준 회 원:현대정보기술 기술자문
†† 정 회 원:광운대학교 컴퓨터공학과, 신기술연구소
논문접수:1996년 4월 9일, 심사완료:1997년 3월 24일

원 과정에 소요되는 시간과 비용을 절감하기 위해 군수 조달 및 관리 정보 시스템(Computer-aided Acquisition Logistic Support : CALS) 전략을 수립하였고, 현재는 점차 민간, 특히 제조업 분야에 응용되고 있으며 따라서 의미도 경영 지원 통합 정보 시스템(Continuous-Acquisition and Life-cycle Support)으로 현재는 광속 전자 거래(Commerce at Light Speed)로 변하고 있다. 그런데 이 CALS 전략이 바로 서류와의 전쟁에서 파생된 것이라고 말 할 수 있다[3, 4].

하지만 단순한 문서 처리의 전산화만이 문제 해결이 아님을 인식하게 되었다. 즉 문서 정보교환의 문제점을 발견하게 된 것이다. 따라서, 문서 정보를 전자화하고 이 기종 시스템간에 서로 상호교환하기 위해서는 문서를 표준적인 형태로 저장하는 것이 중요한데[5], CALS에서의 문서 정보 표현에 관한 표준을 독자적으로 제정한 것이 아니라 지금까지의 표준을 선택적으로 채용하고 있다. 즉 이미 널리 채택된 표준을 CALS에서는 추구하고 있다. 따라서 CALS에서는 문서양식 표준으로 현재 전세계적으로 가장 널리 채택되어 사용되고 있는 국제 표준 ISO 8879 : SGML (Standard Generalized Markup Language)을 공식적으로 문서 표준 SGML-28001로 채택하고 있다[1, 3]. 또한 현재 인터넷상에서 활용되고 있는 WWW(world wide web)의 HTML 역시 SGML의 한 응용이며, 그 이용 증가가 기하급수적으로 늘고 있고, 계속적인 HTML 버전업이 이루어지고 있는 실정이다.

현재 SGML 문서를 저작하고 관리할 수 있는 상용 도구로는 ArborText의 ADEPT, Xsoft의 SGMLEditor, Frame의 FrameBuilder, nissho iwai의 SGML-plus, opentext의 opentext5, xerox의 DocuBiuld1.2, 기타 수많은 저작 도구들이 있다. 이들 사용 도구의 형태를 살펴보면, 기존 워드프로세서나 DTP(Desk Top Publishing) 시스템에 SGML 저작 기능을 부가하는 형태와 SGML를 기초로 한 저작 도구 형태로 분류할 수 있다. 하지만 이러한 상용 도구들은 현재 한글처리를 지원하지않고 있는 시스템이 대부분이며, 또한 한글로 정의된 문서 형 정의(document type definition : 이후로는 DTD로 명함)를 처리할 수 있는 시스템은 없다. 국내에서는 기초적인 형태의 저작 도구가 발표되었으나 프로토타입 시스템이었다[6].

따라서 본 논문에서는 위에서 언급한 프로토타입 시스템을 더 발전시킨 설계 방법을 통해 SGML 한글처리 및 사용자 인터페이스를 보강하고, 동시에 DTD를 저작할 수 있는 DTD/SGML 문서 저작 도구를 설계, 구현한다.

2. SGML

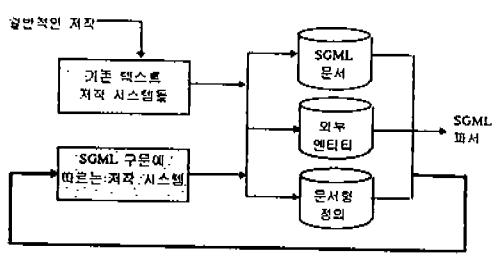
IBM에서 GML(Generalized Markup Language)을 구축하는 작업을 해왔던 Goldfarb는 국제 표준 기구(ISO)의 지원하에서 범용 마크업의 개념에 따른 표준을 작성하게 되었고, 그래서 만들어진 SGML은 다시 수년간의 노력 끝에 1986년 10월 ISO 표준 8879로 공식 채택되었다[5].

이 SGML 표준은 이 기종 시스템 간의 SGML 문서 전송을 위해서는 하나의 완성된 SGML 문서를 준비하여야 한다[1, 5, 7, 8]. ISO 8879에서는 한 완성된 SGML 문서를 SGML 문서 엔티티(entity)라 하며, 이 SGML 문서 엔티티는 SGML 선언(SGML Declaration), DTD, 그리고 문서 실례(Document Instances) 3가지 주요 구문으로 구성된다.

3. DTD/SGML 문서 저작 도구 설계 및 구현

본 3장에서는 2장에서 언급한 DTD를 그래픽 도형으로 표시하고 편집할 수 있고, SGML 문서를 작성할 수 있는 저작 도구의 설계 및 구현에 대하여 설명한다.

3.1 개 요



(그림 1) SGML 구문 작성 시스템
(Fig. 1) SGML syntax editing system

ISO 권고 안에서는 SGML 문서를 작성하는데 (그림 1)과 같은 메카니즘을 권장하고 있다. 즉 SGML 문서를 만드는데는 기존의 일반 워드프로세서를 이용하거나 아니면 SGML 전용 저작 도구를 이용할 수 있다. 하지만 저작자가 기존의 일반 텍스트 저작 시스템을 이용하여 직접 SGML 문서를 작성하는데는 어려움이 있다[9, 10, 11]. 즉 문서를 작성하는 저작자가 문서의 논리구조를 모두 기억하고 작성해야 한다는 것이다.

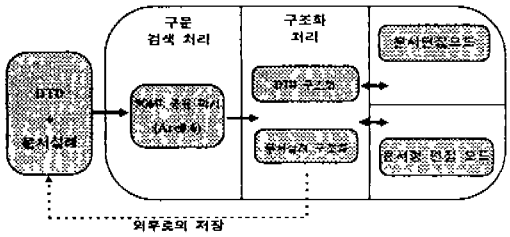
따라서 SGML 전용 구문 작성 시스템에서는 한 SGML 문서를 작성하는데 있어서 텍스트 및 마크업의 입력과 수정하는데 대화적인 수단으로 제공되어야 한다.

또한 DTD를 공용으로 등록된 것을 사용하지 않을 때, 독자적인 DTD를 만들어 사용해야 한다. 이러한 경우 DTD를 그래픽적으로 손쉽게 만들 수 있는 환경을 제공하는 도구가 필요하다[12].

3.2 구 성

본 연구에서 제시한 DTD/SGML 문서 저작 도구의 설계는 다음의 4부분으로 구성하였다(그림 2).

1) 구문 검색 처리
2) 구조화 처리
3) 문서 편집 모드
4) DTD 편집 모드



(그림 2) DTD/SGML 문서 저작 도구의 구성
(Fig. 2) Diagram of the DTD/SGML document editor

3.2.1 구문 검색 처리

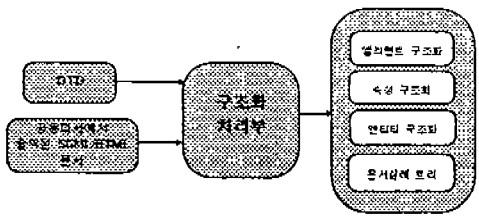
구문 검색 처리는 공용 SGML 구문 분석기를 이용하여 처리한다. 공용 SGML 구문 분석기는 (그림 3)과 같이 DTD와 SGML 문서 실례를 입력받아서 구문 및 구조에 관한 검색을 하고 만약 오류가 검출되면 오류의 종류에 대한 메세지를 알리고, 오류가 발견되지 않으면 출력으로 완전한 문서를 출력하게 된다.

(그림 3) 공용 SGML 구문 분석기의 기능
(Fig. 3) Function of public SGML parser

본 개발에서는 SGML 구문 분석기를 직접 구현한 방법을 탈피하였고, 그 이유는 검색 기능을 담당하는 도구를 국제적으로 검증된 것을 사용하기 위해서 이다. 사용한 공용 SGML 구문 분석기는 SGML user's group이 제공하고 있는 ARC 버전 0.6을 한글처리가 가능하도록 수정하여 사용하고 있고 이 구문 분석기는 SGML 처리를 통합 처리하는 구문 분석기이다[13]. 이 공용 구문 분석기 ARC 0.6은 본 구현 시스템에서 단지 SGML 엔티티를 구문 검색하여, 오류가 있는지만을 사용자에게 알려주는 역할을 담당하고 있다.

3.2.2 구조화 처리

공용 SGML 구문 분석기에서 구문 검색 후 DTD와 SGML/HTML 문서 실례를 내부 구조화하기 위해 다시 한번 DTD와 출력되어 나온 SGML 문서를 내부적으로 스캔한다. 스캔된 DTD의 엘리멘트(element) 데이터, 속성(attribute) 데이터 그리고 엔티티 데이터들과 SGML/HTML 문서 데이터를 내부적으로 구조화시킨다(그림 4).



(그림 4) SGML 데이터 구조화 처리
(Fig. 4) Process for SGML data structure

(그림 4)와 같이 엘리멘트, 속성, 엔티티 및 문서 트리에 관한 구조화는 실제 문서 편집 모드와 문서형 편집 모드의 기초 데이터 구조를 제공하게 된다.

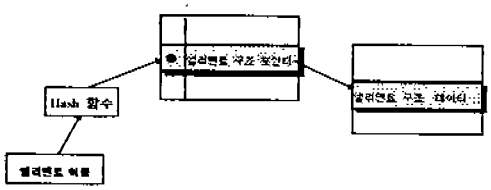
(1) 엘리멘트 구조화

엘리멘트의 구조화는 (그림 5)와 같은 데이터 구조로 저장되며, 후에 SGML 문서 편집 모드와 DTD 편집 모드에서 삽입, 삭제, 감싸기, 변경 등과 같은 처리를 할 때, 엘리멘트들의 데이터 구조를 알기 위한 탐색이 필요하다. 따라서 탐색시간을 단축하기 위하여 해쉬 구조를 이용하였다.

(그림 5) 엘리먼트 데이터 구조
(Fig. 5) Element data structure

(그림 5)의 각 구성 요소들에 대한 정의된 값의 유형을 자세히 살펴 보면 〈표 1〉과 같다.

사용한 해쉬 구조는 엘리멘트 이름을 등록하는 해쉬 테이블을 53개의 구조 테이블로 구성하였고 chaining 방식을 이용하고 있다(그림 6).



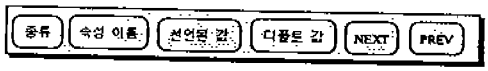
(그림 6) 해쉬구조를 이용한 엘리멘트 구조 찾기
(Fig. 6) Search for element structure using hash function

(2) 속성 구조화

속성 선언은 엘리멘트에 대한 속성을 나타내는 부분으로 이에 대한 데이터 구조는 (그림 7)과 같이 저장되며, 그 구조에 대한 포인터는 엘리멘트 구조에 포함되어 왔다.

〈표 1〉 (그림 1)의 상세 명세
〈Table 1〉 The detail specification of (Fig. 1)

구분	저장 내용		
	표 현		의 미
	매크로 명칭	지정값	
모델形	MCHARS	0X80	#PCDATA 포함
	MGI	0X40	GI를 포함
	MPHARSE	0X20	첫번째 토큰이 #PCDATA
	MANY	0X10	모든 GI 및 #CDATA 포함
	MKEYWORD	0X08	단독 KEYWORD로 정의
	MNONE	0X04	EMPTY로 정의
	MRCDATA	0X02	RCDATA로 정의
	MCDATA	0X01	CDATA로 정의
그룹形 (발생지시자 및 연결자)	TOREP	TOPT + TREP	선택적 + 반복적(발생지시자)
	TOPT	0X80	선택적 발생지시자
	RREP	0X40	반복적 발생지시자
	TTAND	0X08	AND 연결자
	TISEQ	0X04	SEQ 연결자
	TTOR	0X02	OR 연결자
태그생략	TMNSTART	0X80	시작 태그생략
	TMNEND	0X08	종료 태그생략
내용		포인터 값	
예외		포인터 값	
속성		포인터 값	



(그림 7) 속성 데이터 구조
(Fig. 7) Attribute data structure

(3) 엔티티 구조화

엔티티 선언은 매크로 기능을 갖도록 SGML에서 제공하고 있는 선언이다. 엔티티 선언에는 매개변수 엔티티(parameter entity)와 일반 엔티티(general entity) 두 가지 종류로 크게 분류된다.

특히 매개변수 엔티티는 DTD 내에서 쓰이는 매크로 선언으로써 문서의 구조와 매우 밀접한 연관성을

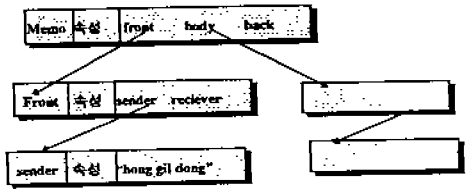
갖고 있어 DTD 편집 모드에서 중요한 역할을 하게 된다. 즉 그룹핑(grouping)의 기능을 이 엔티티를 이용하여 처리한다. (그림 8)에서 데이터 구조를 보이고 있다.

(그림 8) 엔티티 데이터 구조
(Fig. 8) Entity data structure

(4)문서 실례 트리

SGML 구조화 처리에서는 또한 공용 SGML 구문 분석기를 거쳐 나온 SGML 문서 실례를 분석하여 엘리멘트들의 트리 구조를 형성한다. 이 구조는 문서편집 모드에서 실제 화면상에서 텍스트 데이터의 입력 및 삭제와 태그 엘리멘트의 입력 및 삭제를 하기 위한 트리 데이터 구조(그림 9)로 사용된다. (그림 10)에서는 (그림 9)의 구조를 이용하여, 발생 엘리멘트들의 트리 구조를 형성한 것을 보이고 있다.

(그림 9) 엘리멘트 트리 노드의 데이터 구조
(Fig. 9) Data sturcture of element tree nodes



(그림 10) 트리 구조의 연결 형태
(Fig. 10) Connection of tree nodes

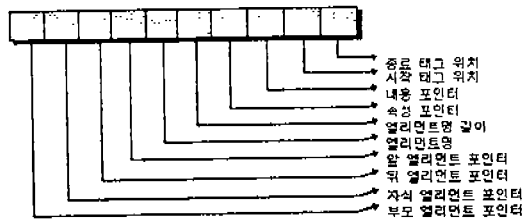
3.2.3 문서 편집 모드

문서 편집 모드에서는 SGML/HTML 문서 실례를 작성하는 모드로서 크게 태그처리와 텍스트 처리, 전체 보기 처리, 3가지 부모들(submodule)로 구분되어 있다.

(1)내부 트리 데이터 구조

SGML 문서를 작성하기 위해서는 문서 내용을 화면으로 WYSWIG하게 보여 주어야 한다. 따라서 문서의 구조 정보 및 내용 정보 외에 관리해야 할 기타 정보들이 많이 요구되므로 이러한 정보를 함께 수용할 수 있는 문서에 대한 내부 트리 데이터 구조를 갖고 있어야 한다. 본 SGML 문서 저작 도구는 구조화 처리부에서 준비된 문서 트리 구조를 내부 트리 데이터 구조로 변환하여 관리한다.

내부 트리 데이터 구조는 (그림 11)과 같이 엘리먼트의 이름과 엘리먼트의 부모, 첫번째 자식, 첫번째 속성, 내용을 가리키도록 하였고 화면 상에서의 출력을 위해서 좌표 값과 이름의 길이를 구조에 포함하고 있다.



(그림 11) 내부 트리 데이터 구조
(Fig. 11) Data structure of internal tree node

(2)태그 처리

SGML/HTML 문서를 작성하는데 있어서 그 문서의 논리 구조를 나타내는 태그를 그 문서에 대해 DTD가 제공하는 규칙 안에서 수정이 가능해야 한다. 이러한 기능을 지원하기 위한 태그 처리부가 필요하다. 본 저작 시스템에서는 태그 처리를 위한 다음 4가지 기능을 가지고 있다.

1) 태크 삽입(Tag Insert): 문서의 특정 위치(현재 커서가 있는 위치)에 삽입될 수 있는 엘리멘트들 중 하나를 선택하여 삽입한다.
2) 태그 삭제(Tag Delete): 특정 위치의 엘리멘트를 삭제한다.

3) 태그 변경(Tag Change): 특정 위치의 엘리멘트를 변경 가능한 엘리멘트들 중 하나를 선택하여 이름을 변경한다.

4) 태그 감싸기(Tag Surround): 마우스를 이용, 일정 영역을 선택했을 때, 그 선택된 부분을 포함할 수 있는 엘리멘트들 중 하나를 선택하여 영역을 감싼다.

이러한 태그 처리를 하기 위해서는 삭제, 삽입, 변경 그리고 감싸기를 특정 위치에서 할 수 있는 엘리멘트들을 엘리멘트 데이터 구조로부터 찾게 된다(그림 12).

(그림 12) 태그 처리 과정
(Fig. 12) Process of tagging

(3) 텍스트 처리

텍스트 처리는 SGML 문서의 데이터를 작성하는데 사용하기 위한 작은 워드 프로세서 기능이다. 일반 워드 프로세서의 기능을 모두 갖추는 것이 이상적이나 본 텍스트 처리는 단순한 텍스트의 입력과 삭제 기능만을 갖는다.

또한 영문은 물론 한글을 처리하기 위한 오토마타 모듈은 조합형 코드로 구성되어 있으나, 최종 글자는 KS-5601 완성형 코드로 만들어지게 구성하였다.

(4) 전체 구조 보기

또한 한 문서의 논리 구조가 크게 되면 사용자가 지금 문서의 어느 부분, 어느 위치에서 작업을 하고 있는지를 모르게 되는 문제가 발생하게 된다. 이러한 문제를 해결하기 위해 문서의 논리 구조를 계층별로 검색할 수 있는 전체 구조 보기 기능을 두었다.

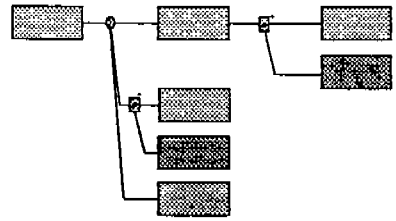
전체 구조 보기는 문서의 논리 구조 계층 중 첫 계층 구조를 처음에 보이고, 사용자가 보고 싶은 계층으로 점점 확장해 나가 한 부분의 계층 구조를 한 눈에 알아 볼 수가 있어 현재 사용자가 작업하고 있는 문서의 위치를 알 수가 있다.

3.2.4 DTD 편집 모드

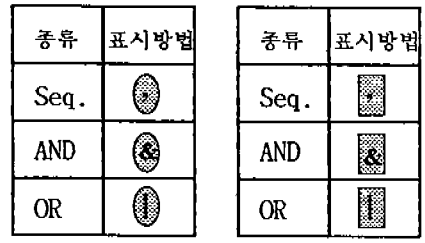
DTD 편집 모드는 사용자가 DTD를 보다 쉽게 편집할 수 있도록 하기 위해 그래픽 화면에서 작업하도록 하였으며, 또한 문서의 논리 구조를 이해하기 쉽게 표현하도록 하기 위해 "DTD 트리"를 사용하였다.

(1) DTD 트리

DTD를 그래픽으로 보여주기 위해 트리 구조로 표현하는데 여기에서 표현되는 트리를 "DTD 트리"라고 정의하였다. 이 DTD 트리는 다른 트리와 구별되는 몇 가지의 특징을 갖고 있는데 (그림 13)에 표현 예를 보이고 있다. 즉 (그림 13)에서 엘리멘트 선언의 내용 모델(content model)을 표현할 때 연결자(connector)와 발생 지시자(occurrence indicator)를 포함하여 표현되고 있다는 점이다.



(그림 13) DTD 트리
(Fig. 13) DTD tree



(그림 14) 연결자의 표시
(Fig. 14) Symbols for connectors

• 연결자의 표현

DTD는 내용 모델의 부엘리멘트 간에는 하나의 연결자(connector)로서 연결되어 표현된다. 연결자의 종류는 ","(SEQ), "|"(OR), "&"(AND) 3가지가 있으며, (그림 14)에서 (a)와 (b)의 두 가지 형태로 분리한 것은 중첩된 모델그룹(model group)을 구분하기 위한 표현 방법이다.

• 발생 지시자의 표현

DTD에서 내용 모델의 부엘리멘트(subelement)들의 발생 빈도를 표현하는 식별자가 있는데 이를 발생 지시자라고 한다. 이러한 발생 지시자는 다음과 같은 특성으로 표현된다.

? :0 또는 1번

+:1번은 반드시 반복 가능

* :반복될 수도 있으나 발생되지 않을 수도 있음.

이러한 발생 지시자의 표현으로는 다음의 (그림 15)에 나타난 것과 같이 표시하였다.

구 분	표 시 방 법
엘리먼트	*
모델그룹	? + *

(그림 15) 발생 지시자의 표시
(Fig. 15) Symbols for occurrence indicators

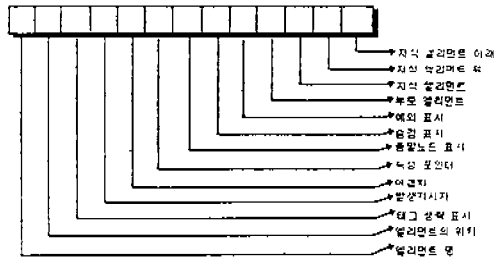
• 중첩된 모델 그룹(Model Group)의 처리

엘리멘트의 내용 모델은 각각의 부엘리멘트로 구성되어 있는데 이러한 부엘리멘트가 다시 중첩된 모델 그룹으로 묶여 표현될 수 있다. 예를 들면 (A, (B|C), D)에서 (B|C)는 중첩된 또 다른 모델 그룹에 해당된다. 이러한 중첩된 모델 그룹의 표현을 (그림 14)와 (그림 15)에 따라 표기한다.

(2)내부 데이터 구조

본 DTD 편집에서는 화면에 DTD를 그래픽적으로 표현하기 위해서는 부가적인 화면 정보와 편집할 때 필요한 정보를 담고 있는 데이터 구조를 필요로 한다. 따라서 구조화 처리부의 정보 데이터를 DTD 저작 도구에서 사용하기 위해 내부적인 변환 처리를 하여야 한다.

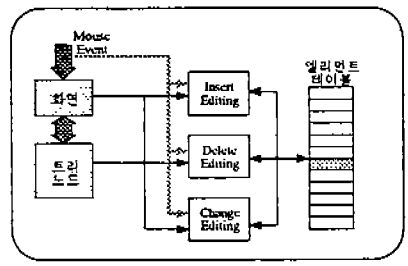
내부 데이터의 기본 구조는 구조화된 포인터(structured pointer)로 되어 있으며 (그림 16)과 같다.



(그림 16) 내부 데이타 구조
(Fig. 16) Internal data structure

(3)편집 기능

문서의 논리 구조를 나타내는 DTD를 SGML이 제공하는 규칙 안에서 수정이 가능해야 한다. 이러한 기능을 지원하기 위해 새로운 엘리멘트의 삽입, 삭제, 그리고 변경의 세 가지의 기본 기능(그림 17)과 편집을 좀 더 용이하게 하는 그루핑(grouping), 엔티티 편집, 엔티티 확장, 확장 취소 등의 특수한 기능을 두고 있다.

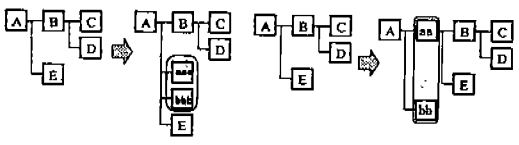


(그림 17) 새로운 엘리멘트의 편집
(Fig. 17) Editing process for new element

• 삽 입

태그의 삽입은 기존의 DTD에 새로운 엘리멘트를 만들고자 할 때 사용하고자 하는 기능이다. 실질적인 삽입은 (그림 18)과 같은 형태로 자매 관계와 자식 관

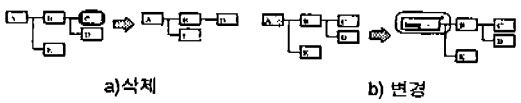
계 삽입으로 이뤄지게 된다. 화면에서 편집한 후 이에 일치하는 엘리멘트 데이터 구조를 새로 만들거나 기존의 엘리멘트가 있는 경우는 데이터 구조의 내용을 수정하게 된다.



(그림 18) 엘리멘트의 삽입
(Fig. 18) Inserting new element

• 삭 제

엘리멘트의 삭제 기능은 기존의 DTD에서 필요 없게 된 엘리멘트를 없앨 수 있는 기능으로 이벤트의 발생에 따라 원하는 트리 노드를 찾아가 화면상에서 삭제(그림 19(a))하며, 일치하는 엘리멘트 데이터 구조에서 그 엘리멘트를 삭제시킨다.



(그림 19) 엘리멘트 삭제 및 변경
(Fig. 19) Inserting and changing for element

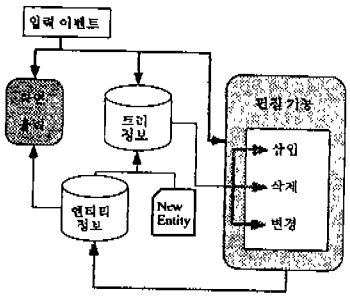
• 변 경

엘리멘트의 변경 기능은 기존의 DTD에서 한 엘리멘트의 이름이나 연결자, 발생지시자를 바꾸고자 할 때 이벤트가 발생된 트리를 찾아 화면상에서 변경이 가능하며, 내부적으로는 엘리멘트의 데이터 구조에서 변경되어 저장되게 된다(그림 19(b)).

• 엔티티의 편집

위에서 설명한 편집 기능은 DTD 중 엘리멘트 자체에 관한 것이다. 그러나 DTD의 엔티티는 독립된 트리 형태로 구성되는데 이를 여기서는 "엔티티 트리"라고 한다. 엔티티 트리는 선언된 엔티티가 어떻게 구성되어 있는지를 명확히 보여줄 수 있다. 또한 엔티티의 편집이 가능하다. 편집 기능은 DTD 편집 기능과 같은 기능을 가지고 있다. 또한 새로운 엔티티를 정의할 수 있도록 설계 되었다. 즉, 새로운 엔티티의 정의 및 기존 엔티티에 삽입, 삭제, 변경이 가능하도록 하였다.

또 엔티티 편집에서는 엔티티 내부에 다른 엔티티가 포함이 되었을 때 엔티티로 표시해주며, 그 부분에서 다시 편집이 가능하도록 설계하였다. 엔티티 편집은 (그림 20)과 같이 구성하였다.



(그림 20) 엔티티 편집
(Fig. 20) Editing for entitys

• 그룹핑

DTD에서 몇 개의 엘리멘트들이 묶어 그룹을 형성하여 하나의 발생 지시자를 부여할 수 있다. 이런 처리를 하기 위하여 그룹핑의 기능을 두고 있다. 몇 개의 엘리멘트들을 화면상에서 마우스를 통해 잡아 끌기(Dragging)를 하면 원하는 영역에 시각형으로 표시가 된다. 이때 그룹이라는 명령을 주게 되면 원하는 부분이 하나의 그룹으로 처리되도록 하였다.

• 그룹 해제

그룹 해제는 그룹이 더 이상 유효하지 않을 때 그룹으로 표시되던 부분만 삭제하는 기능이다.

• 엔티티 확장

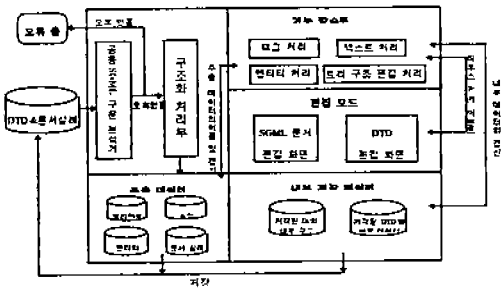
트리 구조를 표현하는 데는 종말 노드의 정의가 필요하다. SGML에서 종말 노드로 사용할 수 있는 것은 CDATA, RCDATA, EMPTY, ANY 그리고 #PCDATA이다. 하지만 본 DTD 편집 모드에서는 엔

티티도 종말 노드로 처리하고 있다. 이것은 화면상에서 DTD를 표현하는데 재순환적인 무한 반복을 피하기 위한 방편이다. 따라서 이러한 엔티티들의 1차 하위 노드를 볼 수 있는 확장 모드가 필요하다. 따라서 모든 엔티티에 대해서는 확장하는 기능을 두어 전체적인 구조를 볼 수 있게 하였다.

4. 실험 및 고찰

4.1 구 현

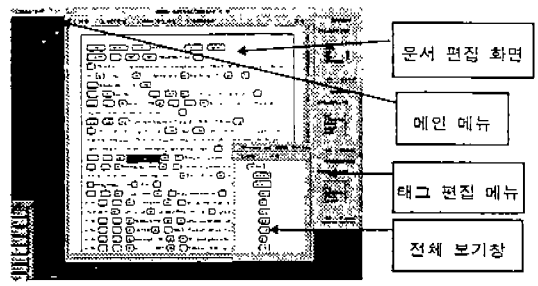
본 연구에서 설계한 SGML/HTML 저작 시스템은 UNIX를 사용하는 Sun SPARC 10을 호스트로 하여 NCD16 그래픽 터미널이 네트워크로 연결된 환경하에서 수행하였다. 윈도우 시스템은 X 윈도우 시스템을, 그래픽 유저 인터페이스(Graphic User Interface: GUI)는 Motif와 UIL(User Interface Language)을 이용하였으며, 폰트는 X 윈도우 시스템의 디폴트 영문 비트맵 폰트를 사용하였다. 본 구현 도구의 전체적인 처리 흐름은 (그림 21)과 같다.



(그림 21) 구현 시스템
(Fig. 21) Diagram of the inplemented system

4.2 실 험

문서 작성 모드에서 HTML 문서의 작성 결과는 (그림 22)에서 보이고 있다. (그림 22)에서 메인 메뉴는 문서 작성 도구와 DTD 도구를 선택할 수 있는 주메뉴이다. 문서 편집화면은 SGML 문서를 태그와 함께 표시하여 저작자가 내용을 키인(key-in)할 수 있는 윈도우이다. 문서 편집 화면에서 태그에 관련된 삽입, 감싸기, 변경 처리를 하려면 문서화면 위에 있는 메뉴중 태그 편집 메뉴를 선택하면 감싸기, 삽입, 삭제,

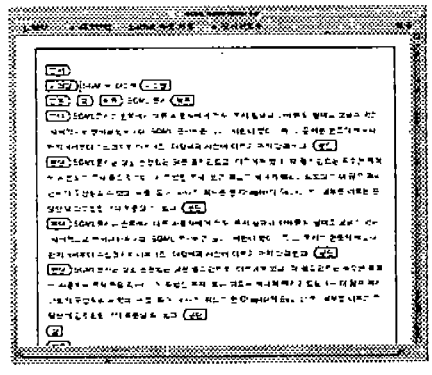


(그림 22) HTML 문서 편집
(Fig. 22) Editing HTML document

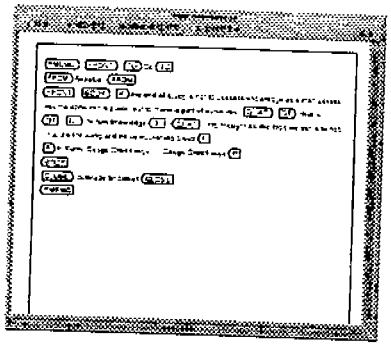
변경 부메뉴가 나타나게 되며, 그중에 하나를 선택하게 되면 (그림 22)에 나타난 것처럼 해당 태그 편집 메뉴 박스가 나타나게 되며, 그 박스 메뉴에는 현재 커서가 있는 위치에서 아니면 태깅된 위치에 들어갈 수 있는 엘리멘트 이름들이 검색되어 나타난다. 또한 전체 보기창은 현재 문서의 부분적인 트리 구조를 피쉬아이(fish-eye) 형태로 보여준다.

그리고 실제 한글 DTD를 이용하여 SGML 문서를 작성한 결과는 (그림 23)에 나타나 있다. 또한 "pmemo" DTD를 이용한 영문 SGML 문서의 작성은 (그림 24)에서 결과를 보였다.

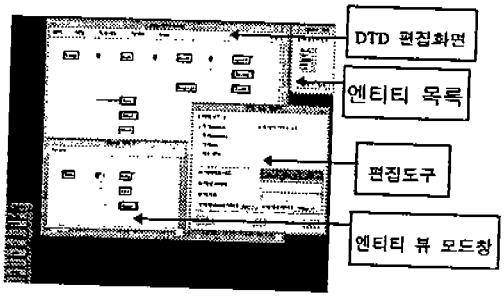
(그림 25)에서는 DTD 편집 모드에서 DTD를 작성하는 예를 보여 준다. (그림 22)의 주메인 메뉴로부터 DTD 도구를 선택하면 (그림 23)의 DTD 편집 화면이 나타난다. 실제 DTD를 저작할 때에 가장 많이 사



(그림 23) 한글 DTD를 이용한 한글 SGML 문서
(Fig. 23) Hangul SGML document using Hangul DTD



(그림 24) "pmemo" DTD를 이용한 "pmemo" 문서
(Fig. 24) "pmemo" document using "pmemo" DTD



(그림 25) DTD 편집
(Fig. 25) DTD editing

용하는 것이 편집 도구이다. 이 도구를 이용하여 하나의 엘리멘트를 정의하게 되고 부 엘리멘트의 갯수 및 부엘리멘트의 발생 지시자, 연결자 등을 모두 지정하게 된다. 그 지정된 값에 의해서 편집 화면에 한 엘리멘트와 그 부엘리멘트들이 서로 연결된 모습으로 나타나게 된다. 엔티티 목록은 매크로로 정의된 모든 엔티티 리스트를 보여주고 있고, 엔티티 뷰 모드창은 선택된 엔티티의 구조를 새로운 창을 여러 보여 준다.

4.3 고 찰

본 논문에서는 SGML/HTML 문서 및 DTD를 저작할 수 있는 DTD/SGML 문서 저작 도구를 구현하였다. 본 저작 도구의 구현은 구문 검색 처리는 공용 SGML 구문 분석기인 ARC 구문 분석기를 사용하였다. 그리고 SGML 데이터를 저작 도구에서 이용하기 위하여 SGML 정보의 추출 및 저장은 구조화 처리부를 따로 설계하였다. 실제 편집 모드는 SGML 문서 편집 모드와 DTD 편집 모드를 두어 구성하였다.

본 저작 도구는 사용하기 편리한 그래픽 사용자 인터페이스를 제고함으로써 사용자가 범용의 SGML 문서와 HTML 문서를 작성할 수 있었고, 또한 DTD를 그래픽 트리로 정의하여 구성할 수 있었다.

그러나 본 시스템에서는 SGML의 구문 검색을 SGML 공용 구문 분석기를 이용함으로써 공용 SGML 구문 분석기의 기능에 의존하는 한계점을 갖고 있다. 또한 문서 편집 모드에서는 태그의 기능에 초점을 두었기 때문에 상대적으로 텍스트 관련 처리 기능이 매우 간단하다. 또한 DTD 편집 모드에서는 DTD에 필요한 속성 선언, 표기법 선언, 단축참조 선언 등의 생성 기능이 없다. 이러한 점들은 SGML/HTML 저작 시스템이 발전되어 감에 따라 개선되어야 할 점이다.

5. 결 론

현재 WWW상에서 HTML이 널리 확산되어 사용되고 있어, SGML 및 HTML 문서를 손쉽게 저작할 수 있는 시스템이 절실히 필요한 때이다. 따라서 본 연구에서는 국제 표준안 ISO 8879(SGML)에 근거하여 DTD와 SGML/HTML 문서를 편집할 수 있는 도구를 개발하였고 그 유효성을 입증하였다.

앞으로는 단순히 SGML/HTML 문서 파일의 이용이 아니라 SGML 문서를 DB화하여 CGI(common gateway interface)를 통해 HTML 문서로 변환하여 이용하려는 쪽으로 기술이 개발되고 있는 움직임이 일고 있다. 따라서 본 개발 DTD/SGML 문서 저작 도구는 앞으로 이러한 기술에 따라 CGI을 이용한 DB와 연계를 모색하고 더욱 발전해 갈 것이다.

또한 국내에 CALS가 도입됨에 따라 SGML 문서 처리 환경의 구축을 필요로 하는 시점에 있다. 따라서 앞으로 좀 더 SGML 관련 기술의 개발이 활성화되어야 할 것이다.

참 고 문 헌

[1] Martin Bryan, "An Author's guide to the Standard Generalized Markup Language", 1988.
[2] 小町祐史, "文書記述言語の標準化 動向-I", 情報處理, Vol.32, No.10, pp.1110-1125, Oct. 1991.
[3] DoD, "MIL-M-28001A", 20 july 1990.
[4] Eric Van Herwijnen, "Practical SGML second edition," kluwer academic publishers
[5] ISO 8879, Information Processing-Text and Office System-Standard Generalized Markup Language(SGML), 1986.
[6] 현득창, "SGML Parser를 이용한 SGML Document Editor의 구현에 관한 연구", 정보과학회 논문지, 제2권 제4호, pp.484-494, 1993.
[7] David Barron, "Why use SGML", Electronic Publishing, Vol.2(1), pp.3-24, 1989.
[8] ISO 9096, Information Processing-SGML Support Facilities-SGML Document Interchange Format(SDIF), 1989.
[9] "DTR 10037 Information Processing-Text and Office System-Guideline for SGML Syntax-Directed Editing Systems", ISO/IEC JTC1/SC18/WG8 N707
[10] ISO/TR 9573 Techniques for Using SGML(Geneva;ISO), Dec. 1988.
[11] "Operational Model for Text Description and Processing Language", ISO/TG97/SC18/WG8 N484
[12] Kiyoshi Toyoda, Eri Kumagai and Tatsuo Bando, "SGML Document Structure Editor", National Technical Report Vol.36, No.5, 1990.
[13] Jos Warmer and Sylvia Van Egmond, "The implimentation of the Amsterdam SGML", Electronic Publishing, Vol.2(1), pp.3-90, 1989.

현 득 창

1992년 광운대학교 공과대학원 전자계산기 공학 전공(공학석사)
1992년~1997년 광운대학교 공과대학원 전자계산기 공학과 박사
1995년~1997년 2월 현대미디어 시스템 IETM 개발팀
1997년 2월~1997년 3월 현대미디어시스템 기술고문
1997년 4월~현재 현대정보기술 및 HCL 기술고문
관심분야:전자도서관, CALS, SGML, HyTime

이 수 연

1969년 광운대학교 전자통신공학과 졸업(공학사)
1977년 연세대학교 전자통신공학과 졸업(공학석사)
1983년 일본 교또대학교 정보공학과 졸업(공학박사)
1977년~현재 광운대학교 컴퓨터공학과 교수
1994년~현재 광운대학교 신기술연구소 연구원
관심분야:하이퍼미디어 문서처리, 디지탈 전자도서관, SGML, HyTime, HTML