

# 문서병합 기능을 갖는 XML 문서 편집기의 설계 및 구현

김규평<sup>○</sup> 이종하  
 대구효성가톨릭대학교 컴퓨터정보통신공학부  
 (g9521005, jhlee11@cuth.cataegu.ac.kr)

## Design and Implementation of XML Document Editor for the Document Merging

Kyu-Pyong Kim<sup>○</sup> Jong-Hak Lee  
 School of Computer and Information Communications Engineering, Catholic Univ. of Taegu-Hyosung

### 요 약

최근 XML(eXtensible Markup Language) 문서의 효율적인 작성과 유지관리를 위해 다양한 XML 문서 편집기들이 개발되고 있다. 하지만, 기존의 편집기들의 경우 하나의 XML 문서에 대한 편집기능은 제공하고 있으나 다수의 XML 문서들에 대한 병합(merge)기능은 제공하지 못하고 있다. 본 논문에서는 다수의 XML 문서를 병합할 경우 발생할 수 있는 이질종복과, 이질종복들의 해결 방법을 제안하고, 문서병합 기능을 갖는 XML 문서 편집기를 설계하고 구현하였다. 문서병합 기능을 갖는 XML 문서 편집기는 여러 곳에 산재하고 있는 동일한 부류의 여러 XML 문서들을 병합해야 할 경우에 용이하게 사용될 수 있다.

### 1. 서론

XML(eXtensible Markup Language)은 1996년 W3C(World Wide Web Consortium)의 후원으로 형성된 XML Working Group에 의해 개발되었다. 1998년 2월에 최종 권고안이 발표된 XML 1.0[1]은 SGML(Standard Generalized Markup Language)의 부분집합이며, SGML과 HTML(Hyper Text Markup Language) 양쪽 모두와의 상호운용이 가능하고 웹 문서의 용이한 구현을 위해 설계되었다[2].

XML 문서는 문서의 구성 요소인 엘리먼트(element)간의 논리적인 계층구조를 가지고 있는 DTD(Document Type Definition)와 DTD에 선언된 엘리먼트에 따라 기술된 XML 문서내용(Document Instance)을 가진다. XML 파서(parser)는 DTD를 이용하여 XML 문서의 유효성(validity) 검사를 시행한다. 그것은 XML 문서안에 기술된 내용이 DTD에 선언된 엘리먼트 타입의 논리적 계층구조에 일치하는지를 검사하는 것이다. 그리고 유효한 XML 문서내용은 웹 에이전트와 같은 XML 어플리케이션(application)에 제공되어 사용된다.

사용자가 직접 작성한 XML 문서는 오류를 범할 가능성이 있으므로, 현재 많은 XML 문서 저작 및 편집도구가 개발되고 있다. 또한 XML 문서의 구조 검색이나 내용 검색에 사용되는 다양한 XML 어플리케이션들이 제작되고 있다. 이 도구들은 사용자가 특별한 지식이 없이도 웹 문서를 쉽게 생성할 수 있도록 해야하며 접근하기 쉬운 사용자 인터페이스를 제공하여야 한다[3].

본 논문에서는 문서병합(merge) 기능을 갖는 XML 문서 편집기를 제안하고 이를 구현한다. 기존의 편집기들[4, 5, 6, 7, 8]은 하나의 XML 문서를 대상으로 하는 편집 기능만을 제공하며, 다수의 XML 문서를 병합하는 기능은 지원하지 않고 있다. XML 문서병합 기능은 여러곳에 산재하고 있는 동일한 부류의 여러 XML 문서들을 병합하여 사용할 경우 매우 필요한 기능이다.

본 논문의 구성은 다음과 같다. 제 2절에서는 XML 문서병합에 필요한 기본 개념을 설명하고, 제 3절에서는 문서를 병합하는 과정에서 발생하는 문제점과 그 해결 방법을 제시한다. 그리고, 제 4절에서는 병합기능을 갖는 XML 문서 편집기의 설계와 구현에 대한 내용을 기술한다. 마지막으로, 제 5절에서는 결론 및 향후 연구과제를 기술한다.

### 2. XML 문서병합을 위한 기본 개념

XML 문서에서 XML 문서형의 선언은 하나의 문서 형태를 제공하는 DTD를 명시하는 역할을 한다. 문서형 선언은 문서 외부(별도의 파일)에 정의되어 있는 마크업 선언들을 가리킬 수도 있고 문서

내부에 직접 포함할 수도 있다. DTD란 문서형 선언에서 정의되는 문서 형태에 대한 논리적인 계층구조를 나타내는 일종의 문법이라 할 수 있다[9]. DTD는 엘리먼트들의 집합으로 내용 모델(Content Model)[10]에 따라 기술한다. 예를 들어, 보고서에 대한 문서를 작성할 경우 그 엘리먼트들은 제목, 계층인, 목차, 본문, 초록 등으로 생각할 수 있다. 작성하고자 하는 문서의 형태에 해당하는 것이 보고서이고, 보고서의 세부적인 구조 규칙들은 보고서의 DTD가 된다.

내용 모델은 DTD의 엘리먼트형 선언에서 어떤 엘리먼트의 내용이 어떤 형태로 구성되어야 하는가를 정해주는 모델이다[10]. 하위에 올 수 있는 엘리먼트들, 그 엘리먼트들의 반복 횟수, 순서, 텍스트 사용여부 등이 내용 모델로서 정해 줄 수 있다. ANY(엘리먼트내에 어떠한 것이든 허용됨), EMPTY(어떤 것도 허용되지 않음), #PCDATA(텍스트 입력 가능함)의 세 가지 예약어와 엘리먼트명, 그리고 표 1과 같은 연결자(Sequence[,], Choice[|]), 발생 지시자(?, \*, +)등을 적절히 조합하여 내용 모델을 구성한다.

표 1: 내용 모델에서 사용되는 기호

기호	용도
-	순서지정
	선택
+	반복(최소 1회)
*	반복(최소 0회)
?	생략 가능
∅	그룹화

표 2는 A, B, 그리고 C와 같은 이름을 사용하는 세 개의 엘리먼트와 내용 모델에서 사용되는 기호들과의 조합에 의한 내용 모델의 예를 보여준다.

표 2: 내용 모델에서 사용되는 기호들의 조합 예

내용모델	설명
A	엘리먼트 A가 한 개만 나와야 함
A B	A와 B 엘리먼트 중 한 개의 엘리먼트만 나와야 함
A, B	A 엘리먼트가 나온 뒤, B 엘리먼트가 순서대로 한 개씩만 나와야 함
A, B?	A 엘리먼트가 한 개 나온 뒤, B 엘리먼트가 나오지 않거나 한 개만 나와야 함
(A, B, C)	A, B, C 세 엘리먼트가 순서대로 한 개씩 나와야 함
A(B C)	A 엘리먼트가 반드시 한 개 나온 뒤, B나 C는 순서에 상관없이 여러개 나올 수도 있고 아무 것도 나오지 않을 수도 있음
A(B C)+	A 엘리먼트가 반드시 한 개 나온 뒤, B나 C는 순서에 상관없이 한 개 이상 나올 수 있다.
A(B, C)	A 엘리먼트가 한 개 나오거나, B와 C가 순서대로 한 개씩 나와야 함

3 DTD 병합시 발생될 수 있는 문제점과 해결방법

XML 문서를 병합하기 위해서는 먼저 DTD를 병합하여야 한다. 이는 DTD가 XML 문서에 대한 일종의 스키마이기 때문이다. XML 문서의 DTD에서는 같은 엘리먼트 이름을 가지는 엘리먼트가 존재해서는 안 된다. 만약 병합하기 전 두 XML 문서의 DTD에서 같은 이름을 가진 엘리먼트가 있을 경우에는 유효성 검사에서 오류를 발생하게 된다. XML 문서병합의 경우 발생할 수 있는 스키마 충돌 문제로는 엘리먼트 이름충돌과 이름중복이 있다. 엘리먼트 이름충돌은 병합하려는 두 개의 XML 문서가 있을 경우 DTD의 엘리먼트에 포함되어 있는 태그(tag)에 기록되는 이름, 즉 엘리먼트 이름이 같은 종류의 엘리먼트 내용에 대하여 서로 다른 경우이다. 그리고, 엘리먼트 이름중복은 서로 다른 종류의 엘리먼트 내용에 대하여 같은 엘리먼트 이름들이 주어지는 경우이다. 병합 기능을 갖는 XML 문서 편집기는 이런 이름충돌과 이름중복을 자동으로 감지하고, 편집기능을 조작하는 사용자에게 자동으로 알려 줌으로써 사용자가 인터페이스로 조작을 해야 하는 부분만 편집하여 유효한 XML 문서를 생성해 줄 수 있도록 한다.

3.1 동일한 종류의 XML 문서의 병합

그림 1은 서로 동일한 종류의 XML 문서에 대한 동일한 DTD의 예를 보여준다. 즉, 엔진1과 엔진2의 DTD 트리 구조는 동일하다. 두 DTD에서 엔진 엘리먼트는 일련번호, 이름, 배기량 엘리먼트를 가진다. 그리고, 그림 1에서 #PCDATA는 문자 데이터를 나타내며 각 엘리먼트가 문자 데이터를 가짐을 나타낸다. 그러므로 XML 문서는 두 DTD 중 하나를 선택하여 병합되어 질 수 있다.

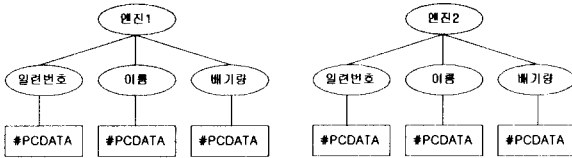


그림 1: 엔진1과 엔진2의 DTD 트리 표현

그림 2는 동일한 종류의 XML 문서에 대해 서로 다른 이름의 엘리먼트 이름을 가진 두 개의 DTD 트리를 나타낸다. 엔진의 DTD에서 엔진 엘리먼트는 일련번호, 이름, 배기량 엘리먼트를 가지고 있고, 자동차엔진의 DTD에서 자동차엔진 엘리먼트는 고유번호, 명칭, 엔진배기량 엘리먼트를 가지고 있다. 엔진 DTD의 일련번호 엘리먼트와 자동차엔진 DTD의 고유번호 엘리먼트는 엘리먼트의 이름이 다르지만 같은 종류의 엘리먼트 내용을 가지므로 문서병합시 이름충돌이 발생한다. 이런 경우 어느 한 쪽의 DTD를 기준으로 다른 쪽의 엘리먼트 이름을 같은 엘리먼트 이름으로 변경시켜서 이름충돌 문제를 해결하여야 한다.

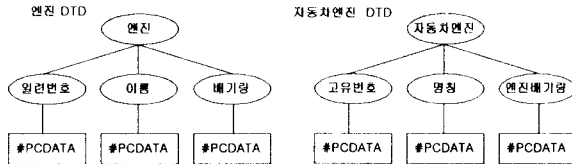
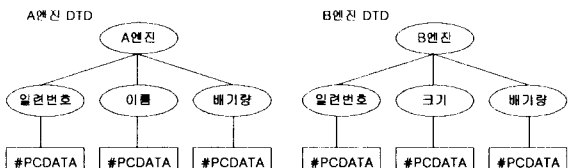


그림 2: 엔진과 자동차엔진의 DTD 트리 표현

3.2 서로 유사한 XML 문서의 병합

그림 3은 서로 유사한 XML 문서를 기술한 DTD의 예를 트리로 보여준다. A엔진의 DTD에서는 A엔진 엘리먼트는 일련번호, 이름, 배기량 엘리먼트를 가지고, B엔진의 DTD에서는 B엔진 엘리먼트는 일련번호, 크기, 배기량 엘리먼트를 가진다. 그리고 일련번호, 이름, 배기량, 크기 엘리먼트들은 모두 문자 데이터를 가진다. 두 DTD에서 이름 엘리먼트는 A엔진의 DTD에만 들어있고 크기 엘리먼트는 B엔진의 DTD에만 들어 있으며, 일련번호 엘리먼트와 배기량 엘리먼트는 공통적으로 들어있다. 그림 4는 XML 문서 파일속의 A엔진과 B엔진의 DTD 코드를 나타내고 있다.



- A엔진의 DTD  
 <!ELEMENT 엔진 (일련번호, 이름, 배기량)>  
 <!ELEMENT 일련번호 (#PCDATA)>  
 <!ELEMENT 이름 (#PCDATA)>  
 <!ELEMENT 배기량 (#PCDATA)>
- B엔진의 DTD  
 <!ELEMENT 엔진 (일련번호, 크기, 배기량)>  
 <!ELEMENT 일련번호 (#PCDATA)>  
 <!ELEMENT 크기 (#PCDATA)>  
 <!ELEMENT 배기량 (#PCDATA)>

그림 4: XML 문서 파일속의 A엔진과 B엔진의 DTD 코드

그림 5는 A엔진과 B엔진의 두 DTD를 C엔진의 DTD로 통합할 경우 그 통합된 결과를 보여준다. 일련번호 엘리먼트와 배기량 엘리먼트는 이름중복 때문에 자신의 상위(부모) 엘리먼트의 이름을 사용해 A엔진-일련번호, A엔진-배기량, B엔진-일련번호, B엔진-배기량 엘리먼트 이름으로 변경하여야 한다. 그러나, C엔진의 DTD에서 A엔진-일련번호 엘리먼트와 B엔진-일련번호 엘리먼트는 같은 종류의 내용을 가지고, A엔진-배기량 엘리먼트와 B엔진-배기량 엘리먼트도 같은 종류의 내용을 가진다. 이와 같이 두 DTD에서 같은 종류의 엘리먼트가 많은 경우에는 두 DTD를 통합하기 보다는 병합하는 것이 바람직하다. 그림 6은 A엔진과 B엔진의 두 DTD를 C엔진의 DTD로 병합한 결과를 나타낸다. 그림 6의 C엔진 DTD에서 C엔진 엘리먼트는 일련번호, 이름, 크기, 배기량의 엘리먼트들을 가지며, 이름과 크기 엘리먼트는 둘 중에 하나만 선택적으로 나타남을 알 수 있다. 그리고, 그림 7은 XML 문서 파일속의 C엔진 DTD 코드를 나타내고 있다.

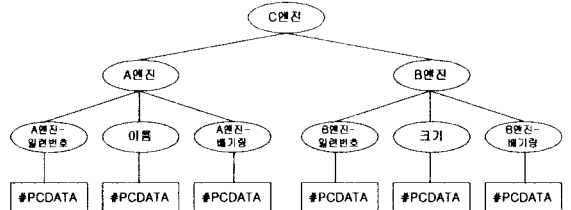


그림 5: A엔진과 B엔진의 두 DTD를 C엔진 DTD로 통합한 트리 표현

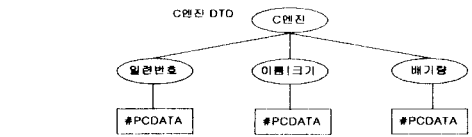


그림 6: A엔진과 B엔진의 두 DTD를 C엔진 DTD로 병합한 트리 표현

- C엔진의 DTD  
 <!ELEMENT C엔진 (일련번호, 이름|크기, 배기량)>  
 <!ELEMENT 일련번호 (#PCDATA)>  
 <!ELEMENT 이름 (#PCDATA)>  
 <!ELEMENT 크기 (#PCDATA)>  
 <!ELEMENT 배기량 (#PCDATA)>

그림 7: XML 문서 파일속의 C엔진의 DTD 코드

3.3 서로 다른 XML 문서의 통합

그림 8은 서로 다른 종류의 XML 문서를 통합하기 위한 두 문서의 DTD 트리를 나타낸다. 엔진의 DTD에서 엔진 엘리먼트는 일련번호, 이름, 배기량 엘리먼트를 가지며, 브레이크 DTD에서 브레이크 엘리먼트는 일련번호, 등급, 제철 엘리먼트를 가진다. 즉, 그림 8에서 엔진과 브레이크에 대한 DTD는 서로 다른 종류의 XML 문서를 나타내는 것이다. 그리고, 그림 9는 XML 문서 파일속의 엔진과 브레이크에 대한 DTD의 코드를 나타내고 있다.

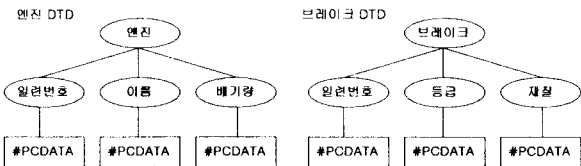


그림 8: 엔진과 브레이크의 DTD 트리 표현

그림 10은 엔진과 브레이크의 두 DTD를 자동차의 DTD로 통합해야 될 경우 그 통합된 결과를 보여준다. 자동차의 DTD에서 자동차 엘리먼트는 엔진과 브레이크 엘리먼트를 가지며 그 하위 엘리먼트도 가지게 된다. 그림 11은 자동차의 DTD 코드를 보여준다. 엔진과 브레이크의 DTD에서 엘리먼트 이름은 같지만 서로 다른 종류의 내용을 가지는 일련번호 엘리먼트에 대해 엘리먼트 이름중복 문제가 발생한다. 따라서, 자동차의 DTD로 통합할 경우 서로 다른 엘리먼트 이름을 가지게 하여 엘리먼트 이름중복 문제를 해결하여야 한다. 그림 11에서와 같이 엔진 엘리먼트의 일련번호 엘리먼트를 엔진-일련번호로, 브레이크 엘리먼트의 일련번호 엘리먼트를 브레이크-일련번호로 이름을 변경해 줌으로써 엘리먼트 이름중복 문제를 해결할 수 있다. 즉, 중복이 일어나는 엘리먼트가 있을 경우 엘리먼트 이름 앞에 그 엘리먼트의 상위 엘리먼트의 이름을 식별자로 추가함으로써 DTD를 오류 없이 통합시킬 수 있다.

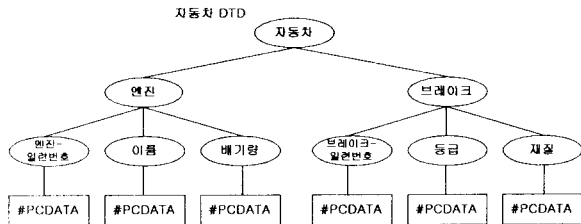


그림 10: 엔진의 DTD와 브레이크의 DTD를 자동차의 DTD로 통합한 트리 표현

● 자동차의 DTD

```

<!ELEMENT 자동차 (엔진, 브레이크)+>
<!ELEMENT 엔진 (엔진-일련번호, 이름, 배기량)>
<!ELEMENT 브레이크 (브레이크-일련번호, 이름, 배기량)>
<!ELEMENT 엔진-일련번호 (#PCDATA)>
<!ELEMENT 이름 (#PCDATA)>
<!ELEMENT 배기량 (#PCDATA)>
<!ELEMENT 브레이크-일련번호 (#PCDATA)>
<!ELEMENT 등급 (#PCDATA)>
<!ELEMENT 재질 (#PCDATA)>
    
```

그림 11: XML 문서 파일속의 자동차 DTD 코드

4. 병합 기능을 가진 XML 문서 편집기

XML 문서의 병합 기능을 가지는 편집기의 전체 구성은 그림 12와 같다. DTD 파일이나 XML 문서파일을 읽은 후 XML 문서 편집기는 XML 문서병합을 실행하고 그 결과를 파일로 출력해 준다. 사용자의 목적에 따라 XML 문서 편집기의 편집 기능을 사용할 수도 있고 병합 기능을 사용할 수도 있다.

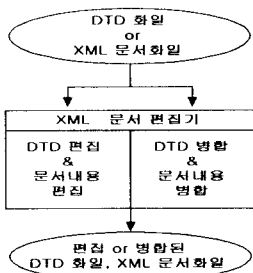


그림 12: 문서병합 기능을 갖는 편집기의 전체 구성

그림 13은 XML 편집기에서 문서병합 시스템의 구성을 보여 준다. 병합하기 위한 XML 문서들에 대해 유효성 검사를 먼저 시행한 후, XML 문서내용은 XML 문서내용 트리로 변환된다. 그리고 DTD는 DTD 트리로 변환되어 DTD 병합 편집기와 DTD 병합 처리기에 입력된다. DTD 병합 편집기에서 사용자가 목적으로 하는 XML 문서병합의 구조를 지정해 주면 엘리먼트 이름중복과, 엘리먼트 이름중복을 검사한다. DTD 병합 처리기는 DTD 병합에서 발생하는 정보를 사용자에게 인터페이스를 통하여 제공함과 동시에 DTD 병합 처리기와 XML 문서내용 병합 처리기에 제공된다. DTD 병합 처리기는 병합된 DTD 파일을 생성하고 XML 문서내용 병합 처리기는 XML 문서내용 트리와 DTD 병합정보를 입력받아 통합된 XML 문서내용 파일을 생성한다.

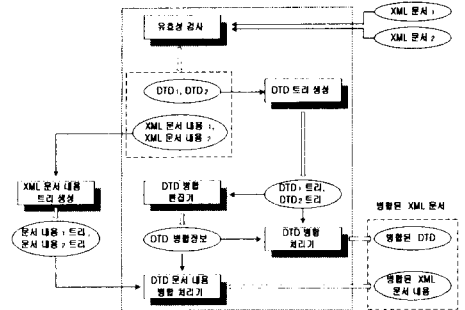


그림 13: XML 문서병합 시스템

그리고, 그림 14는 그림 13과 같은 XML 문서병합 시스템을 사용하기 위한 사용자 인터페이스이다. 사용자 인터페이스에는 병합할 두 XML 문서의 DTD를 트리로 출력해 주고, 그것을 보고 직접 편집할 수 있는 두 개의 편집창과, 병합의 결과를 DTD 트리로 보여주는 결과창이 있다. 그리고, XML 문서병합에 필요한 정보를 표시하는 메시지 표시창과 여러 기능선택에 필요한 선택버튼들이 있다.

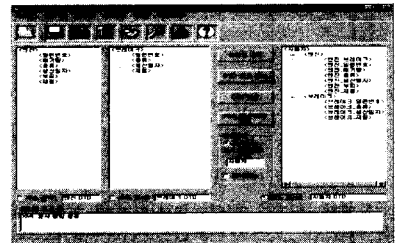


그림 14: XML 문서병합 시스템의 사용자 인터페이스

5. 결론

XML 문서가 증가함에 따라 기존의 문서를 쉽게 재사용하면서 웹 애플리케이션과 같은 XML 어플리케이션들이 XML 문서를 쉽게 접근 하도록 하기 위한 연구가 많이 진행되고 있다. 본 논문에서는 XML 편집기가 여러 XML 문서를 병합할 수 있는 기능을 가짐으로써 XML 문서의 재사용성과 XML 어플리케이션의 접근성을 향상시킬 수 있음을 보인다. 따라서, 사용자는 기존의 XML문서를 재사용함으로써 작성하고자 하는 다양한 문서를 쉽고 빠르게 작성할 수 있다.

향후 연구과제로는 엘리먼트가 가지는 애트리뷰트(attribute), 엔티티(entity) 등의 효율적인 병합에 대한 연구이다. 이를 기반으로 웹 상에서 유용한 정보를 추출하여 XML 문서로 통합함으로써 새로운 XML문서를 만들 수 있는 XML 문서 편집기를 개발하고, 이를 지원 하는 어플리케이션을 개발하는 것이다.

6. 참고문헌

- [1] W3C, Extensible Markup Language(XML) 1.0, <http://www.w3.org/TR/1998/REC-xml-19980210.html>, Feb. 1998.
- [2] C. Baru, et al., "XML-Based Information Mediation with MIX," exhibition program, ACM Conf. on Management of Data (SIGMOD'99), Philadelphia, USA. 1999.
- [3] W3C, Authoring Tool Accessibility Guidelines 1.0, <http://www.w3.org/TR/2000/REC-ATAG10-20000203/atag10.txt>, Feb. 2000.
- [4] SoftQuad Software Inc., XMetal, ver. 1.0, 1999.
- [5] 한국정보과학(주), XML DESIGNER, ver. 1.0#, 1998.
- [6] Icon Information-system Inc., XML SPY, ver. 2.5, 1998.
- [7] Microsoft co., XML Notepad, ver. 1.0, 1998.
- [8] Techno2000 Project Inc., CLIP, ver. 1.0#, 1998.
- [9] 이계춘 외, "XML문서의 검색을 위한 효율적인 색인 기법과 질의," 한국정보과학회 추계학술발표논문집, pp.57-59, 1999. 10.
- [10] Frank Boumphrey 외, PROFESSIONAL XML APPLICATIONS, 정보문화사, pp.127-133, 1999. 9.