

# 일반 텍스트 문서를 XML로 변환하기 위한 T2XG 시스템 설계 및 구현 (T2XG System Design and Implementation for General Text To XML Document Translation)

최 유 순\*      김 변 곤\*\*      김 정 옥\*\*\*      한 성 국\*\*\*\*      박 종 구\*\*\*\*\*  
(Yue-Soon Choi) (Byun-Gon Kim) (Jeong-Ok Kim) (Sung-Kook Han) (Jong-Goo Park)

## 요 약

HTML이 웹 페이지를 만드는 데 사용되는 가장 흔한 언어이지만, HTML은 정보 공유에는 제한된 능력을 가지고 있다. 확장형 마크업 언어라고 하는 XML은 현재 웹 상에서 정보를 전달하고 저장하는 데 있어 가장 주목을 받고 있는 언어이다. XML로 표현된 정보는 의미가 부여되므로서 보다 정확하고 빠른 검색을 제공하게 된다. 전자상거래나 가상교육 등의 시스템을 개발하는 최근의 경향은 XML로 구현되고 있는 추세이다. 따라서 기존의 일반 텍스트 문서를 XML로 변환하기 위한 도구가 요구된다. 본 논문에서는 일반 텍스트 문서를 XML로 변환하기 위한 알고리즘을 제시하고, 시스템을 구현하였다.

## ABSTRACT

HTML, a very ordinary language for making web pages, as a restricted ability to share information. XML is what we call 'extension mark-up language'. It is being watched with keen interest for the communication and saving of information. Information represented in XML provides more accuracy and a higher-speed of reference after the process of being implication. For that reason, an instrument which can convert existing general text documents into XML is in great demand. In this thesis, I will describe an algorithm for converting general text documents into XML and create a system to implement this algorithm.

---

\* 정회원 : 원광대학교 BK21 겸임교수  
\*\* 정회원 : 한국표준과학연구원 Ph.Dr  
\*\*\* 정회원 : 정인대학 겸임교수  
\*\*\*\* 정회원 : (주)비트컴퓨터 기술고문  
\*\*\*\*\* 정회원 : 원광대학교 컴퓨터공학과 교수

논문접수 : 2001. 12. 27.  
심사완료 : 2002. 2. 23.

## 1. 서론

최근 인터넷의 급속한 보급은 정보처리 사회에 인터넷 혁명을 불러일으키며 웹을 기반으로 하는 지식사회로 만들었다. 웹은 우리에게 정보 표현 능력과 정보 가공 능력을 제공하면서 대중성을 확보하게 되었다. 웹에서 표현할 수 있는 정보의 형태는 텍스트, 음성, 이미지, 동영상 등의 멀티미디어 데이터이다. 이러한 데이터들은 플랫폼에 독립적이고 정보의 다양한 표현이 가능하며 전송 및 교환이 가능하다. W3C(World Wide Web Consortium)는 HTML(Hyper Text Markup Language)의 단점을 보완하고 SGML(Standardized Generalized Markup Language)의 복잡성을 제거한 XML(eXtensible Markup Language)을 웹 문서의 표준으로 지정하였다.<sup>1)2)</sup>

확장형 마크업 언어라고 하는 XML은 현재 웹 상에서 정보를 전달하고 저장하는 데 있어 가장 주목을 받고 있는 언어이다. HTML이 웹 페이지를 만드는 데 사용되는 가장 흔한 언어이지만, HTML은 정보 공유에는 제한된 능력을 가지고 있다.<sup>3)</sup> 상품 정보를 테이블 형태로 출력할 경우, <table>...</table>로 그 형태를 정의하고, <tr>...</tr>로 줄을 구분하여 한 제품 정보를 구성하며, <td>...</td>로 제품명, 가격 등의 정보를 열로 구분하여 표현한다. 이러한 정보는 사람이 분석하여 의미를 파악할 수 있으나, 태그들을 바탕으로 컴퓨터가 그 의미를 해석하는 것은 불가능하다.

반면에 XML은 간단한 메모에서부터 복잡한 데이터베이스까지 특정 목적에 맞게 사용할 수 있도록 모든 종류의 정보를 기술하는 데 사용할 수 있는 유연한 문법을 제공한다. XML 문서는 스타일시트나 HTML 페이지와 함께 연동하여 웹 브라우저에 쉽게 출력할 수 있다. 데이터의 의미와 구조를 표현하기 위한 태그들을 정의할 수 있어 자신이 담고 있는 정보를 효과적으로 구조화하고 레이블링하기 때문에 브라우저는 유연한 방법으로 정보를 검색하고, 추출하고, 정렬하고, 필터링하고, 재배치하고, 조작할 수 있다. 그래서 빠르게 늘어나고, 복잡해지고 있는 정보들을 다루는 데 이상적인 해결책을 제시한다. 또한 지식의 관리와 보급, 그리고 더 큰 규모의 웹 사이트를 관리할 때 기본 기술로서 중요하게 사용되고 있다.<sup>4)</sup>

XML은 데이터 구조에 대한 정보를 DTD(Document Type Definition)로 기술하고 있다.<sup>5)</sup> DTD는 XML 문서에서 사용할 태그를 정의하고, 이들이 어떤 순서로 동작하는지를 정의한다. DTD는 유효한 XML 문서의 프롤로그에 추가해야만 하는 XML 마크업의 블록이다. 이 DTD는 XML 선언의 다음이라면, 프롤로그 내의 어떤 곳이라도 넣을 수 있으며, 문서의 구조를 정의한다. 만약 인터넷 익스플로러 5에서 DTD가 없는 문서를 열었다면 인터넷 익스플로러 5 프로세서는 문서가 잘 구성되어 있는지만을 체크하게 된다. 그러나 만약, 인터넷 익스플로러 5에서 DTD가 있는 문서를 열었다면, 프로세서는 문서가 잘 구성되었는지 뿐 아니라, 유효성도 검사할 것이다. 따라서 XML 문서는 DTD내의 모든 선언을 따라야 한다. 예를 들면, 문서 타입에 아직 선언하지 않은 요소나 속성을 문서에 포함할 수 없게 되는 것이다. 또한 문서에 포함시킬 모든 요소와 속성은 그에 상응하는 선언에 표현된 스펙과 일치해야 한다.<sup>6)</sup> 이처럼 XML은 훌륭한 특징과 전망이 있다. DTD는 “잘 구성된(well-formed) 문서”의 조건을 만족하는 것보다 더 엄격한 조건을 만족하는 “유효한(valid) 문서”를 만들기 위해 XML에 추가된다.

본 논문에서는 XML 문서의 모델링 개념을 활용하여 일반 문서를 XML 문서로 변환하고, XML 문서에서 DTD를 생성하는 알고리즘을 개발하여 구현하였다. 2장에서는 XML 문서의 구성 요소를 살펴보고, 문서를 모델링하기 위한 과정과 방법을 기술한다. 3장은 XML 문서를 생성하기 위하여 문서의 선형 구조와 트리 구조를 비교하고, XML이 갖추고 있는 트리 구조를 생성하기 위한 알고리즘을 개발한 다음, DTD를 생성하도록 한다. 4장은 3장에서 제시한 알고리즘을 사용하여 DTD문서와 XML 문서를 생성하는 시스템을 구현하였다. 5장에서는 본 논문에서 제시한 시스템의 유용성을 설명하고, 향후 연구 과제에 대하여 논의한다.

## 2. DTD를 생성하기 위한 모델링

이기종 시스템들간의 구조적인 문서 교환을 위한 XML 문서의 사용이 증가함에 따라 이에 대한 모델링 방법들이 연구되고 있다.<sup>7)8)</sup> 그러나 기존의 모델

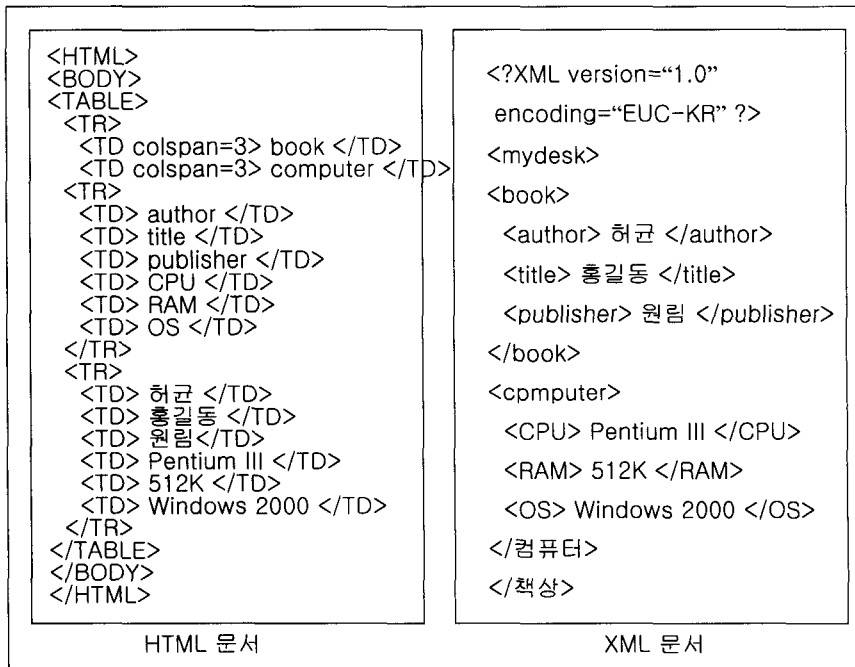
링 방법들은 인위적으로 DTD를 생성하기 위하여 일일이 요소들을 설정해주어야만 했다. 따라서 일반 문서를 모델링하기 위한 자동화된 방법이 요구된다.

## 2.1 XML 문서

대량의 정보 속에서 사용자가 원하는 정보를 빠르게 찾아낼 수 있도록 하기 위해서는 정보 검색 시스템이 정보를 얼마나 빠르게 찾을 수 있는가에 관한 정보 탐색 요소와 더불어 정보 검색 시스템 자체의 정보표현 능력이 어느 정도인가에 관한 정보표현 요소가 갖추어져야 한다.<sup>9)</sup> 정보 표현 영역을 넓혀 사용자가 원하는 정보를 정확하게 표현할 수 있게 하도록 하기 위해서 정보의 구조화는 필요하다. [그림 1]은 HTML 문서와 XML 문서로 정보를 표현한 경우를 보이고 있다.

두 개의 문서는 도서에 관한 정보를 HTML과 XML로 기술하였다. 이 문서들간의 두드러진 차이는 HTML 문서는 구조화 되어있지 않고, XML 문서는 저자와 책명이 구조화되어 있다는 것이다.

사용자가 책명인 홍길동을 검색하고자 할 때, 위의 두 개의 문서 모두 검색이 된다. XML로 표현된 문서는 <title> 태그에서 값이 홍길동을 검색하게 되고, HTML 문서에서는 태그 내에 의미가 부여되어 있지 않으므로 저자인지, 책명인지 구분이 안되는 홍길동을 검색하게 된다. 이는 사용자에게 변별력이 전혀 없어 불필요한 문서를 검색결과에 포함시키는 결과를 가져오게 된다. 홍길동이 책명인 문서를 검색하라는 질의를 수행할 수 있도록 정보가 구조화되어 있고, 검색 시스템이 이러한 정보 구조화를 표현하고 색인할 수 있다면 홍길동에 대한 가장 적절한 정보를 사용자에게 제시할 수 있다.



[그림 1] HTML 문서와 XML 문서

[Fig. 1] HTML and XML Document

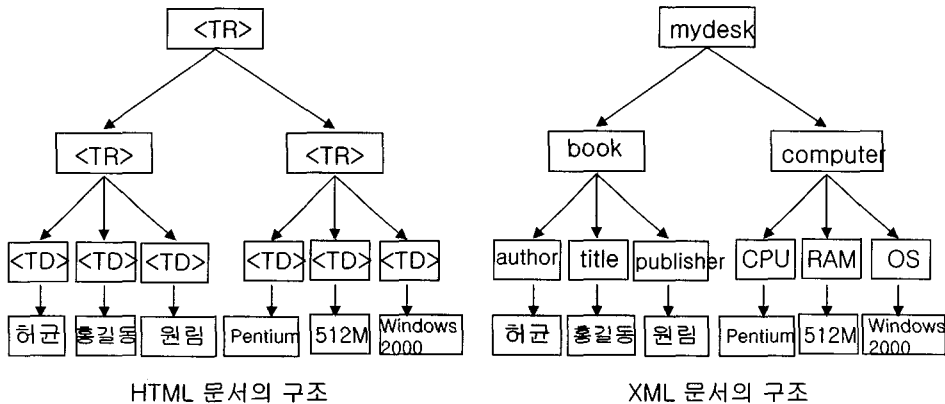
즉 사용자 하여금 자신이 원하는 정보를 세분화하여 검색할 수 있는 정보표현의 확장성을 제공하여 대량의 정보에서 원하는 정보만을 빠르게 찾을 수 있도록 해 준다. 필요한 정보를 찾기 위해 불필요한 시간과 비용을 현저히 감소시킬 수 있기 때문이다.

정보를 구조화하기 위해 적합한 언어로 XML이 많이 사용되고 있고, XML로 표현된 문장은 [그림 2]에서 보는 것과 같이 트리형식으로 구조화된다. 일반적인 문서는 계층구조를 갖는 트리구조로 표현될 수 있으며<sup>10)</sup>, 표현된 구조는 프로그래밍을 통하여 쉽게 핸들링이 가능하다. XML의 경우는 구조화된 문서로 작성되므로 문서 일부에 대한 접근, 데이터베이스의 저장, 구조화된 질의 및 검색 등이 용이하게 된다. 그러나 HTML은 문서의 논리적 구조를 정의하는 구조이지만 노드에 표현된 값에는 의미가 결여되어 있으므로 데이터베이스의 저장이나, 구조화된 질의 및 검색에 어려움이 따른다. XML은 사용자가 원하는 형태로 정보를 구조적으로 구성할 수 있도록 기능을 제공하고 있기 때문에 확장성과 융통성이 풍부하다는 장점을 가지고 있다.

## 2.2 DTD

유효한 XML 문서는 DTD를 갖는데 XML 프로세서는 문서가 구문상 정확한지를 검사하며, DTD에 따라 구조가 기술되어 있는지를 검사한다. XML은 데이터 구조에 대한 정보를 DTD로 기술하고 있다. DTD는 XML 문서 모델링을 위한 가장 기본적인 지식 중의 하나이다. DTD는 문서를 구성하는 요소, 요소의 속성값, 요소간의 관계 등 문서의 형태를 지정하는데 필요한 규칙을 지정한다. DTD에서 정의한 문서 조직 규칙을 만족하는 XML 문서를 유효한 XML문서라 한다.<sup>11)</sup>

XML을 기반으로 원문정보를 제공하고자 하는 도서관에서는 원문을 제공하려는 모든 유형에 대한 DTD를 개발하여야 한다. 문서의 구조 규칙인 DTD는 문서의 종류에 따라, 적합한 문서 구조 규칙을 작성할 수 있다. DTD는 기본적으로 관련된 XML 문서의 논리적 구조를 위한 템플릿을 형성하는데 데이터의 계층구조를 표현한다. [그림 3]은 요소 책에 대하여 모델링한 구조의 DTD표현이다.



[그림 2] HTML 문서 구조와 XML 문서의 구조  
 [Fig. 2] HTML and XML Document Structure

```

<!ELEMENT book (title, author, publisher)>
<!ATTLIST book size CDATA REQUIRED>
<!ATTLIST book price CDATA REQUIRED>
<ELEMENT title (#PCDATA)>
<!ATTLIST title subtitle CDATA REQUIRED>
<ELEMENT author (#PCDATA)>
<!ATTLIST author email CDATA IMPLIED>
<ELEMENT publisher (#PCDATA)>
<!ATTLIST publisher phone CDATA REQUIRED>
    
```

[그림 3] DTD 문서의 예  
 [Fig. 3] An Example of DTD Document

DTD는 크게 요소 선언, 속성 선언, 엔티티 선언, 노테이션 선언의 네가지 형식으로 나눌 수 있으나, 여기에서는 간단한 개념설명을 위하여 요소 선언과 속성 선언만을 사용하여 "book"과 그의 하위 요소인 "title", "author", "publisher"에 대한 정의를 하였다.

### 3.1 XML 문서의 선형 구조와 트리 구조

먼저 일반문서의 예를 보자.

“오늘 12시에 원광대학교 도서관에서 신성철씨가 XML에 대하여 특강을 실시합니다. 컴퓨터관련 전공자들은 모두 참석해 주시기 바랍니다.”

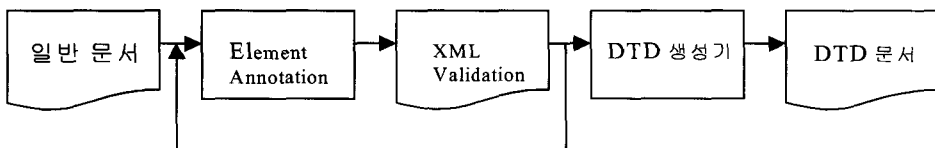
이 문장은 누구나 흔히 사용하는 보편적인 문장이다.

### 3. XML 문서로 변환

사용자에게 정보 검색의 용이성이나 정보 표현성을 볼 때, 기존의 일반 문서를 의미 기반의 XML 문서로 변환할 필요가 있다. 이 때 XML로 표현하기 위해 문서를 다시 작성한다면 시간과 노력, 비용이 많이 소요될 것이다. 기존의 일반 문서를 XML 문서로 변환할 수 있는 방안을 제시하고 T2XG(Text to XML Generator)시스템을 구현하였다. [그림 4]에 시스템의 흐름도를 나타내었다.

#### 3.1.1 선형 구조

위 문장을 XML로 표현하면 다음과 같다.



[그림 4] T2XG 시스템 흐름도  
 [Fig. 4] T2XG System Flow Diagram

```
<?XML version="1.0" encoding="euc-kr" ?>
<문장>
  <날짜>오늘</날짜>
  <시간>12시에</시간>
  <위치>원광대학교</위치>
  <장소>도서관에서</장소>
  <연사>신성철씨가</연사>
  <주제>XML에 대하여</주제>
  <행사>특강을</행사> 실시합니다.
  <대상>컴퓨터관련 전공자들은</대상> 모두 참석하여
주시기 바랍니다.
</문장>
```

위 XML 문서는 <문장>이라는 상위 요소를 가졌고, 나머지 요소들은 모두 같은 단계의 하위 요소로 구성되어 있어 선형 구조를 이루고 있다. 이 문서를 토대로 DTD를 만들면 아래와 같다.

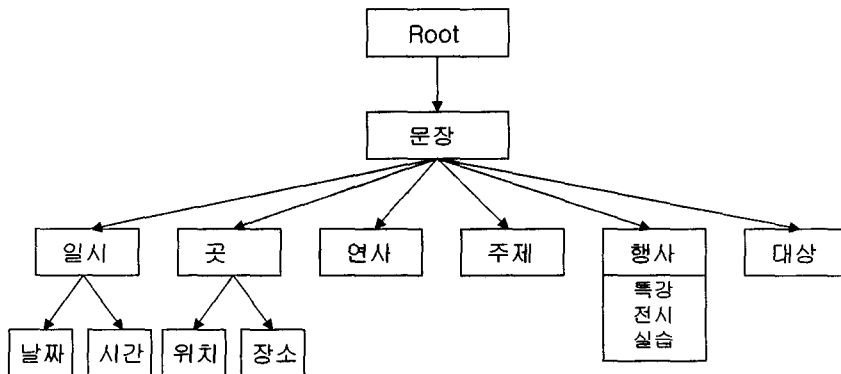
```
<!ELEMENT 문장 (날짜, 시간, 위치, 장소, 연사, 주제,
행사, 대상)>
<!ELEMENT 날짜(#PCDATA)>
<!ELEMENT 시간(#PCDATA)>
<!ELEMENT 위치(#PCDATA)>
<!ELEMENT 장소(#PCDATA)>
<!ELEMENT 연사(#PCDATA)>
<!ELEMENT 주제(#PCDATA)>
<!ELEMENT 행사(#PCDATA)>
<!ELEMENT 대상(#PCDATA)>
```

여기에서 보는바와 같이 XML문서와 DTD문서는 공통적으로 하위 요소나 속성을 가지지 않은 전형적인 선형 구조를 보이고 있다. 그러나 위 문장에서 날짜와 시간의 상위 요소로 <일시>라는 태그를 형성할 수 있고, 위치와 장소가 합해서 <곳>이라는 상위 요소로 표현될 수 있다. 따라서 위의 문서들은 좀 더 복잡한 구조인 트리 구조로 구성할 수 있다.

### 3.1.2 트리 구조

XML 문서가 갖는 구조적인 특징은 트리 구조이다. 위의 예문도 계층적인 트리 구조로 표현할 수 있다. [그림 5]는 예문을 트리 구조로 그린 것이다. 태그 <일시>와 <곳>을 첨가하여 상위 요소를 만들고, 행사요소에 속성을 부여하여 XML 문서를 만들었다.

```
<?XML version="1.0" encoding="EUC-KR" ?>
<문장>
  <일시>
    <날짜>오늘</날짜>
    <시간>12시에</시간>
  </일시>
  <곳>
    <위치>원광대학교</위치>
    <장소>도서관에서</장소>
  </곳>
  <연사>신성철씨가</연사>
  <주제>XML에 대하여</주제>
  <행사 속성="특강" 전시="실습" />
  <대상>컴퓨터관련 전공자들은</대상>
  모두 참석하여주시기 바랍니다.
</문장>
```



[그림 5] 예문의 트리 구조

[Fig. 5] Tree Structure of Sample Statement

```

<연사>신성철씨가</연사>
<주제>XML에 대하여</주제>
<행사>특강을</행사> 실시합니다.
<대상>컴퓨터관련 전공자들은</대상> 모두 참석하여
주시기 바랍니다.
</문장>
    
```

트리 구조를 갖춘 XML 문서를 DTD로 나타내면 다음과 같다. DTD에는 요소가 계층을 이루도록 상위 요소와 하위 요소를 구분하였고, <행사> 요소에는 속성을 부여하였다.

```

<!ELEMENT 문장 (일사, 날짜, 시간, 곳, 위치, 장소,
연사, 주제, 행사, 대상)>
<!ELEMENT 일사 (날짜, 시간)>
<!ELEMENT 날짜(#PCDATA)>
<!ELEMENT 시간(#PCDATA)>
<!ELEMENT 곳(위치, 장소)>
<!ELEMENT 위치(#PCDATA)>
<!ELEMENT 장소(#PCDATA)>
<!ELEMENT 연사(#PCDATA)>
<!ELEMENT 주제(#PCDATA)>
<!ELEMENT 행사(#PCDATA)>
<!ATTLIST 행사 PREFERRED (특강 | 전시 | 실습)>
<!ELEMENT 대상(#PCDATA)>
    
```

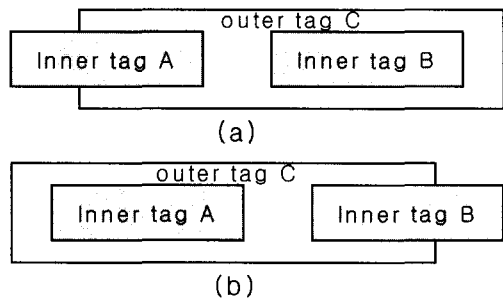
선형 구조에서 만들었던 XML 문서와 DTD 문서에 상위 요소를 추가할 경우 중첩된 요소에 대하여 제한을 두어 잘 정의된 문서로 구성하였다. 새로 만드는 상위 요소는 기존의 하위 요소를 완전히 포함해야 한다.

### 3.2 변환 알고리즘

XML 문서로 변환하기 위하여 본 논문에서는 다음과 같은 알고리즘을 제안한다. XML 생성기는 텍스트 형태의 문서를 입력하거나 읽어올 수 있고, 그러한 문서는 단어 또는 구문을 블록으로 설정하므로서 XML 태그 이름을 입력할 수 있다. XML로 표현된 태그는 열린 태그와 닫힌 태그가 자동으로 삽입되어 XML 폼에 맞는 문서를 구성하게 된다.

XML 태그를 구성할 때 주의할 점은 중첩된 태그의 경우 서로 충돌이 일어나지 않아야 한다. 내부태그는 온전히 외부태그의 일부이어야 하고, 외부태

그는 내부태그를 완전히 포함하여야 한다. [그림 6]은 중첩된 태그를 구성할 때 서로 충돌이 일어나는 경우이다. (a)는 내부태그 A가 외부태그 C의 일부만 포함되어 정상적으로 처리되지 못하는 예이고, (b)는 내부태그 B가 외부태그 C에 일부만 포함되어 있는 예를 보였다. 이런 경우, 외부태그를 정의할 때 내부태그가 존재하는지를 확인하고, 존재한다면 내부태그를 모아 TagList를 만든다. TagList는 새로 정의할 외부태그와 교차하지 않는지를 확인할 수 있도록 해준다.



[그림 6] 중첩된 태그에서의 충돌  
[Fig. 6] Conflict on Nested Tag

XML 문서를 생성하기 위한 변환 알고리즘을 표현하면 다음과 같다.

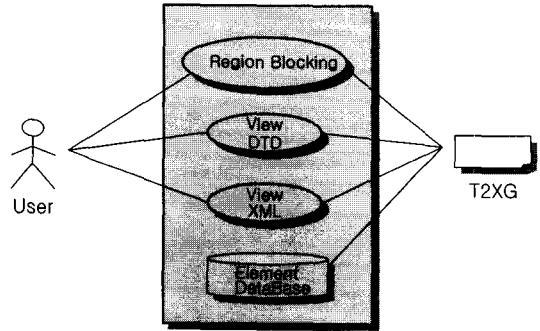
- Algorithm : Document Model Tree Generation  
 Input : marked text region, element and attributes  
 Output : DTD tree Structure  
 Variables : root, mtr, tag, inner tag list TagList
- ① Create document node called root.
  - ② Set the marked text region into mtr.
  - ③ If mtr contains any tags of elements  
 then make an inner text region list called TagList  
 else return mtr attached to parent node
  - ④ Create a new element node mtr with attributes.
  - ⑤ for tag = First(TagList) to End(TagList)  
 if mtr and tag are crossing in their regions  
 then return Error("Crossing Regions")  
 else modify the attachment of tag to mtr
  - ⑥ Attach mtr to parent node.
  - ⑦ Repeat step2 until no more marked text region.

## 4. 구현

### 4.1 모델링

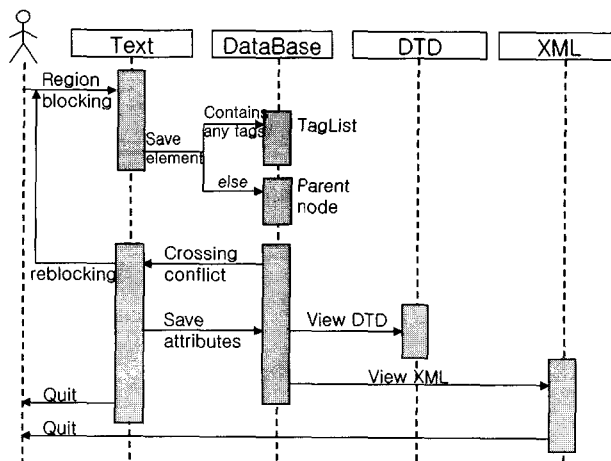
본 논문에서 설계하고, 구현한 T2XG 시스템에 적용할 문서는 일반문서이다. 본 논문에서는 객체지향 설계 및 구현 방법을 적용하여 시스템을 구현하였다.

시스템을 설계하기 위한 유즈케이스 다이어그램을 [그림 7]에 나타내었다.<sup>12)</sup> 액터인 사용자가 T2XG에서 텍스트 파일의 블록을 설정하여 태그와 속성을 정의하면, 다른 액터인 T2XG는 정의된 태그와 속성을 요소 저장소에 저장한다. 이 때, 요소 저장소에서 다른 요소들을 비교하여 이전에 중복된 블록에 대한 요소가 있는지를 검색한다. 중복된 요소가 없으면 TagList에 부모 요소로 저장하게 된다. 만일 중복된 요소 있다면, 기존의 요소를 완전히 포함하는지, 아니면 일부만 포함하는지를 비교하여 중복으로 인한 충돌여부를 파악한다. 충돌현상이 생기는 경우는 완전한 요소를 정의할 수 없으므로 충돌현상은 피하여야 한다. 완전히 내포하고 있는 관계일 경우에만 TagList에 등록한다. 요소에 대한 이름을 정의한 뒤, 필요하면 속성을 부여해 준다. 속성은 요소에서 취할 수 있는 값의 범위를 설정한다.



[그림 7] 유즈케이스 다이어그램  
[Fig. 7] Usecase Diagram

[그림 8]은 순차 다이어그램으로서 T2XG시스템의 동적 구조를 기술하고 있다. 여기에서는 시스템의 사용자와 텍스트 문서, 요소 저장소, DTD, XML 사이의 동적인 행위를 기술하고 있다. 순차 다이어그램은 종 좌표축으로 시간개념을 도입하고 횡 좌표축으로 객체들을 나열하여 그 사이의 상호작용을 표시하였다.



[그림 8] 순차 다이어그램  
[Fig. 8] Sequence Diagram



### 4.2 구현

본 시스템은 Visual Basic6.0으로 구축되었으며, 각 이벤트에 대한 메소드 내용은 다음과 같다. Open\_Click은 파일명을 주고, 해당 텍스트 파일을 불러온다. 텍스트 파일이 열려있는 상태에서 어느 영역을 블록으로 설정하기 위해서 끌기 동작을 취하게 되면, 태그와 속성을 입력할 수 있는 RichTextBox1\_MouseUp 메소드를 실행한다.

```
Private Sub Open_Click()
    CommonDialog1.ShowOpen
    Text1.Text = CommonDialog1.FileName
    Open CommonDialog1.FileName For Input As #1
    .....
End Sub
```

```
Private Sub RichTextBox1_MouseUp(Button As Integer,
Shift As Integer, x As Single, y As Single)
```

```
.....
If Button = 1 Then
    Clipboard.SetText (RichTextBox1.seltext)
    frmTag.Show
    frmTag.seltext.Caption = "[" + Clipboard.GetText +
"] 태그와 속성이름을 입력하십시오"
End If
.....
frmT2XG.File1.Text = File1.Text + ".dtd"
frmT2XG.File1.Text = File1.Text + ".xml"
.....
End Sub
```

```
Private Sub EDIT_Click()
    frmT2XG.RichTextBox1.Text = frmT2XG.RichText
Box1.Text + str + vbCrLf
.....
End Sub
```

```
Private Sub DTD_Click()
    Open File1.Text + ".dtd" For Input As #1
    frmT2XG.Show
    .....
End Sub
```

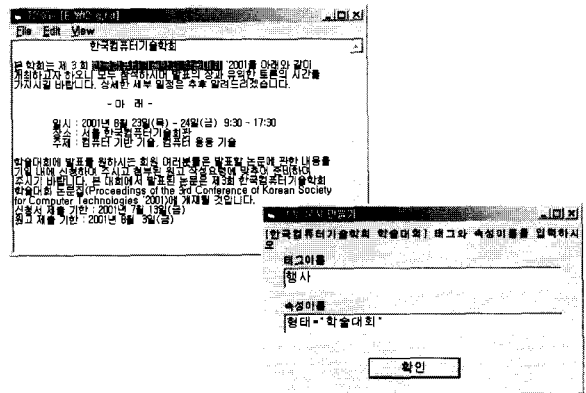
```
Private Sub XML_Click()
```

```
frmT2XG.Show
Open File1.Text + ".xml" For Input As #1
.....
End Sub

Private Sub SAVE_Click()
    Print #1, RichTextBox1.Text
    .....
End Sub
```

본 시스템의 실행 결과를 보기 위해 학회에서 발 행하는 정기적인 학술대회에 대한 내용을 안내하는 공문을 일반 문서로 채택하였다.

[그림 9]에는 문서에서 ‘한국컴퓨터기술학회 학술 대회’를 블록으로 설정하여 ‘XML 문서 만들기’창이 뜨면 태그이름과 속성을 입력하도록 한다. 그 결과 로 생성된 DTD문서와 XML문서가 각각 [그림 10] 과 [그림 11]에 나타나 있다.



[그림 9] T2XG의 실행 예  
[Fig. 9] An Example of T2XG Execution

### 5. 결론

본 연구에서는 정보의 지식처리를 위해서 일반 텍스트 문서를 문서의 의미처리가 가능한 XML문서로 변환하는 시스템을 구현하였다. 방법은 객체지향 분석 및 설계 방법을 활용하였고 변환 알고리즘을 개발하여, XML 문서를 추출하였다.

구현된 시스템은 일반문서에 활용은 물론 도서관에서 학위논문이나 연구보고서의 원문구축 또는 가상문서 브라우징 등 다양하게 응용할 수 있는 개념적인 기초모델을 제시하였다.

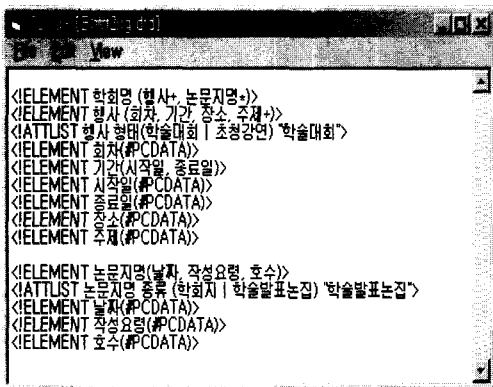
이 시스템에 효율을 높이기 위하여 향후에 추가 설계하여야 할 사항은 다음과 같다.

첫째, 사용자 인터페이스를 강화하여 개발한다.

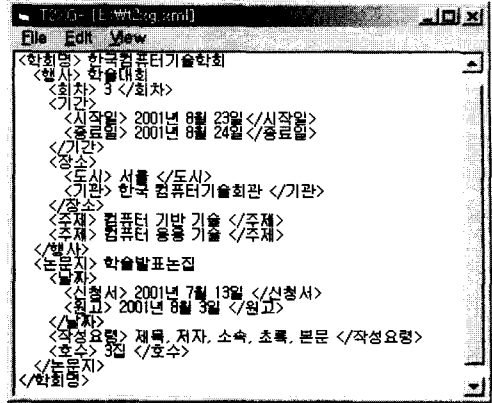
둘째, 자동 요소영역을 추출하는 자동생성 능력을 향상시킨다.

셋째 HWP, PDF, DOC 등의 다양한 파일 형식을 지원할 수 있는 호환성을 높인다.

일반 문서의 XML 문서로의 변환은 의미, 지식 처리에 필수이다. 응용 분야로는 도서관 자료 정리, 웹에서의 효율적인 정보 검색 제공 등 다양하다. 본 논문에서는 정확하고 효과적인 DTD 문서와 XML 문서 생성 시스템을 구현하였다.



[그림 10] DTD 문서  
[Fig. 10] DTD Document



[그림 11] XML 문서  
[Fig. 11] XML Document

#### \* 참고문헌

- [1] W3C, Extensible Markup Language(XML) 1.0, "http://www.w3.org/TR/1998/REC-xml-19980210.html", Feb.1998
- [2] E. Herwijnen, "Practical SGML", 2nd Edition, Kluwer Academic Publishers, 1994.
- [3] Michael Eldmann, Rudi Studer, "Ontologies as Conceptual models for XML Documents", Institut fur Angewandte Informatik und Formale Beschreibungsverfahren(AIFB), University of Karlsruhe(TH), D-76128 Karlsruhe(Germany)
- [4] Mylopoulos, John. (1998). "Information Modeling in the time of the revolution". Information Systems 23(3/4). pp.127-155.
- [5] 김채미, 김심석, 최학열. 2000. "XML DTD와 스키마로 객체지향 모델링하기", Microsoftware 207. pp. 344-353
- [6] 김현희. (1999). "XML을 이용한 가상대학 교육 문헌 구조 설계 및 데이터베이스 구축에 관한 연구". 명지대학교 인문과학연구논총 19. pp. 307-336.

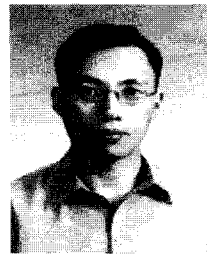
- [7] ArborText Inc., "Data modeling Report Prepared for: W3C XML Specification DTD('XML spec')", September 1998, <http://www.oasis-open.org/cover/xml-report-19980910.htm>.
- [8] 신동규, 신동일, "XML/EDI 시스템의 설계 및 구현", 한국정보처리학회논문지 제8-D권 제2호, 2001.4.
- [9] Yair Wand, "Ontology as a function for meta-modeling and method engineering", ELSEVIER, Information and Software Technology 38(1996) 281-287
- [10] Dick R. Miller, "<http://www.ljdigital.com/netconnect.asp>", 2000.
- [11] <http://www.xml.org/>
- [12] G. Booch, "UML for XML Schema Mapping Specification", Dec 8, 1999.

최 유 순



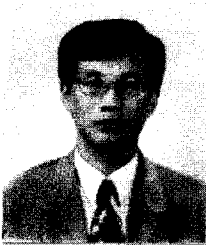
1986년 원광대학교 컴퓨터공학과 학사  
 1990년 원광대학교 컴퓨터공학과 석사  
 2000년 원광대학교 컴퓨터공학과 박사과정 수료  
 2001년~현재 원광대학교 BK21 겸임교수  
 ※ 관심분야 : 소프트웨어공학, 컴포넌트 소프트웨어, 웹프로그래밍, 웹서비스

김 변 곤



1990년 한국항공대학교 항공전자공학과 졸업  
 1997년 전북대학교 전자공학과 석사  
 2001년 전북대학교 전자공학과 박사  
 2001년~현재 : 한국표준과학연구원 인간정보그룹 박사후 연수  
 ※ 관심분야 : Web Server & Network 프로그램, 웹 DB System, 데이터 마이닝, IPv6

김 정 옥



1998년 전북대학교 컴퓨터과  
학과 석사  
2002년 전북대학교 컴퓨터과  
학과 박사 수료  
1985년~1990년 포항제철 시  
스템 개발실  
1990년~1996년 한신생명 정  
보시스템부  
1997년~현재 정인대학 겸임  
교수  
2000년 정보처리기술사  
※ 관심분야 : 컴포넌트 소프  
트웨어, 웹 프로그래밍,  
EJB

박 중 구



1969년 동국대학교 농업경제  
학과 학사  
1975년 동국대학교 전자정보  
처리학과 석사  
1999년 동국대학교 통계학과  
박사  
1981년~현재 원광대학교  
컴퓨터공학과 교수  
※ 관심분야 : 전문가시스템,  
소프트웨어공학, 소프트웨  
어 신뢰성 공학, 모바일  
프로그래밍

한 성 국



1979년 인하대학교 전자공학과  
학사  
1981년 인하대학교 전자공학과  
석사  
1988년 인하대학교 전자공학과  
박사  
1984년~현재 원광대학교 컴퓨  
터공학과 교수  
1979년~1984 대성전자통신연  
구소 비상임연구원  
1988년~현재 미르칸전자 기술  
고문  
1989년~현재 (주)비트컴퓨터  
기술고문  
1990년12월~1992년2월 미국  
펜실버니아대학교 Post-Doc.  
관심분야 : 인공지능(자연언어  
처리), 정보공학, 인지과학,  
객체지향시스템, 컴파일러,  
컴퓨터교육, WBI