

Legacy 데이터베이스를 위한 DTD의 IDREF-ID 속성 관계 모델링

IDREF-ID Attribute Reference Modeling of DTD for Legacy Database

김 정 회* 꺾 호 영**
Jeong-Hee Kim Ho-Young Kwak

요 약

본 논문에서는 Legacy 데이터베이스에서 추출된 정보들에 대해 XML 기술을 적용하고자 했을 때, 참조할 DTD(Document Type Definition) 생성과정 방법을 제안한다. 이를 위해 테이블 간의 복잡한 상호 연결 관계를 표현하고 사전에 ID의 일방적인 추가 단계를 배제할 수 있는 IDREF-ID 속성 관계 모델링을 적용한다. 이 방식에서는 테이블 사이의 관계 및 테이블 사이의 데이터들이 빈번하게 탐색되는 방향을 조사하면서 동시에 ID 추가 작업을 병행 처리하게 된다. 그 결과 IDREF-ID 속성 관계를 파악하면서 동시에 ID 추가 작업이 가능하며, 참조할 DTD 생성도 가능하다.

Abstract

A method of DTD generating step is suggested for applying the XML technology to the information data extracted from the Legacy databases. The IDREF-ID attribute reference modeling is used for representing the complex relationship between tables and excluding the prearranged step of ID insertion. ID insertion procedure is performed in parallel with investigating the relationship between the tables and the frequent search direction between the table data. As a result, ID insertion procedure can be performed simultaneously with understanding of the IDREF-ID relationship between tables, and DTD are also generated.

1. 서 론

인터넷의 발전이 진전되면서 이 기종 간의 시스템에서 작성된 문서에 대한 데이터베이스의 구축과 검색 그리고 상호 교환의 중요성이 높아지고 있다. 이에 따라, 다양한 형식으로부터 원하는 정보를 효율적으로 관리, 공유하기 위해서는 문서를 일관성 있게 구조화하는 기술의 필요성이 대두되었고, 1986년 ISO(International Organization for Standardization)에서는 SGML(Standard Generalized Markup Language)이라는 문서의 논리구조를 표현하는 국제적인 표준안을 마련했다[1].

하지만 SGML은 다양한 기능에도 불구하고, 그

구성이 너무 복잡하다는 단점을 가지고 있고, 이를 해결하고자 HTML(Hypertext Markup Language)이 제기되었지만 이는 제한된 태그로 인해 한계를 가지고 있어서 사용이 부적당하였다. 이에 따라 W3C에서는 일반화된 마크업(Generalized Markup), 복합구조(Complex Structure), 검증(Validation)의 특성을 그대로 지원하는 한편 사용자에 의한 확장성(Extensibility)을 가지고 있는 XML(eXtensible Markup Language)을 제안하였다[2,3,4]. 그래서, 최근의 웹(Web) 또는 디지털 전자 도서관 시스템, CALS(Commerce At the Light Speed), 수학 분야, 채널 기술의 CDF(Channel Definition Format), 이동 통신에서의 HDML(Handheld Device Markup Language)들과 같은 환경에서 많은 문서들이 XML 마크업 언어를 적극 활용하고 있으며 이러한 언어로 문서들을 표현함으로써 문서의 논리적인 구조를 표현할 수 있고, 그럼으로써 문서들을 데이터베이스

* 제주산업정보대학 컴퓨터정보계열 겸임교수
carina@cheju.ac.kr

** 제주대학교 통신컴퓨터공학부 교수
kwak@cheju.ac.kr

에 저장하고 검색 할 수 있는 필요성들이 대두되고 있다[5,6,7,8].

현재 관계형 데이터베이스상에 XML 문서를 저장하거나 추출하는 연구들이 진행중이며 이미 ADO(Active Data Object) 2.5와 SQL Server 2000에서는 각각 일차원적인 구조를 가진 레코드셋을 XML문서로 반환하거나 조인(Join)된 구조를 XML로 직접 추출해 내고 있다. 하지만 이러한 기술들은 계속 성숙되고 있는 단계이며, 관계형 데이터베이스의 테이블 간의 다대 다 관계와 같이 IDREF-ID로 표현되어야 하는 복잡한 관계를 처리하지는 못하고 있다[9].

이에 본 논문에서는 사전에 XML과 관련 없이 저장된 데이터베이스에서 추출된 내용들에 대해 XML 문서 형식들을 적용하고자 했을 때 참조할 DTD(Document Type Definition)의 생성과정을 모델링한다. 모델링 과정에서는 테이블간의 탐색방향과 관계를 조사하고 동시에 ID를 추가함으로써 사전에 ID가 추가되던 작업을 배제하면서 복잡한 IDREF-ID 관계를 표현하도록 하고 있다. 이를 위해 관계형 데이터베이스는 현재 가장 많이 사용하고 있는 오라클(Oracle)을 저장 시스템으로 활용한다.

본 논문의 구성은 다음과 같다. 2장에서는 XML을 중심으로 기존 관련연구를 살펴보고 3장에서는 DTD IDREF-ID 관계 모델링 방식을 설명하고, 타당성을 검증하며 마지막 4장에서는 결론 및 향후 연구 방향을 제시한다.

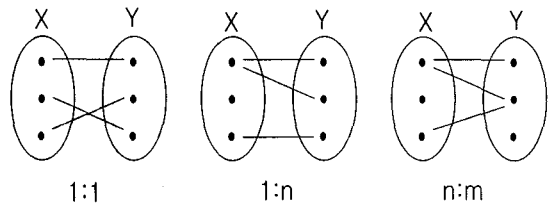
2. 관련 연구

IDREF-ID 관계를 모델링하기 위해서 가장 핵심적인 내용은 테이블간의 관계를 명시하는 작업이다. 처리하고자 하는 작업에 따라 작업 대상이 될 테이블의 리스트가 달라질 것이며, 그에 따라 테이블간의 탐색 방향도 상이할 것이다. 또한 테이블간에는 관계가 부여되어 있어서 1:1(일대 일), 1:n(일대 다), n:m(다대 다)의 특성을 갖는다. 따라

서 테이블간의 관계와 기존에 IDREF-ID 관계를 모델링 한 예를 중심으로 살펴보자.

2.1 테이블간의 관계

두 집합 X,Y가 있을 때 관계를 표현하기 위하여 두 개의 사상 $f: X \rightarrow Y$, $g: Y \rightarrow X$ 로서 생각한다면 이 관계의 사상 특성을 함수성(Functionality)으로 정의할 수 있다. 즉, 사상 f와 g가 모두 함수적이면 X와 Y의 관계는 1:1이며, 어느 하나만 함수적이면 1:n, 그리고 모두 함수적이지 않으면 n:m으로 테이블의 관계가 정의된다[9].



(그림 1) 관계의 함수성 ($f: X \rightarrow Y$, $g: Y \rightarrow X$)

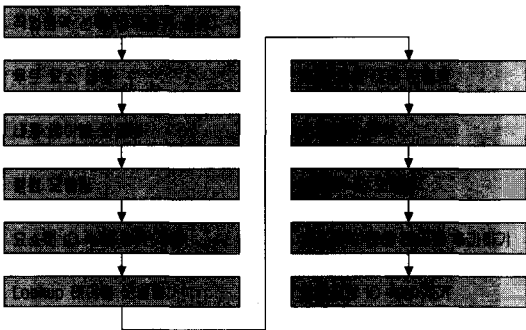
2.2 테이블간의 접근경로

테이블간의 접근경로는 기본 접근 방법 경로와 보조 접근 방법 경로로 구분되며, 기본 접근 경로는 기본키를 이용하여 초기 레코드 적재 또는 레코드 배치와 관련이 있으며, 보조 접근 방법은 보조 인덱스를 통한 레코드 접근을 말한다. 그리고 탐색 방향성이란 테이블간의 데이터를 참조하는 방향이라 볼 수 있다. 그러므로 처리하고자 업무에 따라 테이블간의 접근경로와 탐색방향성은 여러 가지가 나올 수 있으며, 그 중에서 업무 처리에 적절한 접근경로와 탐색 방향성을 결정하여 사용하게 된다[10].

2.3 루트 요소를 제외한 전체 요소에 ID 추가 방식

이 방식은 DTD 추출 방식에서 어떠한 테이블

들이 서로 Join되는지 그 관계를 표현하기 위해 일단 루트 요소를 제외한 전체 요소에 ID를 추가하여 처리하는 방식이 된다. 그리고 최종 완성 단계에서 추가된 ID중에서 참조되지 않는 ID를 삭제함으로써 테이블간의 복잡성을 표현해 내고 있다. 처리해야 하는 테이블의 수가 많아지면 많아질수록 참조되지 않는 ID를 추가하고 삭제해야 하는 단점이 있다[10]. 참고로 이 방식은 다음과 같은 단계로 모델링을 진행하고 있다.



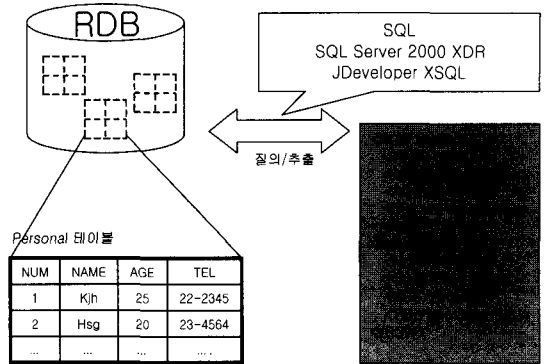
(그림 2) 기존 모델링 알고리즘

2.4 Legacy database에서 XML 문서 추출

관계형 데이터베이스 구조에서 XML 구조로 개발하기 위해 ADO나 SQL Server 2000, 그리고 JDeveloper의 XSQL과 같은 도구들을 이용하면 간단하게 XML 문서를 추출할 수 있다. 하지만 테이블간의 관계에 대한 정보와 참고할 DTD의 생성은 수행되지 않는다. 즉, 그림 3에서 보여주는 것처럼 위에서 언급한 도구들을 이용하여 관계형 데이터베이스에서 데이터를 추출하게 되었을 때 추출된 데이터는 XML문서 형식이지만 테이블들간의 관계 정보나 참조할 DTD 생성은 이루어지지 않고 직접 XML문서로 추출되고 있다[10].

3. DTD의 IDREF-ID 속성 관계 모델링

제안하는 모델링을 위해 온라인 쇼핑몰의 물품 구매와 관련된 데이터베이스 구조를 그림 3과 같이

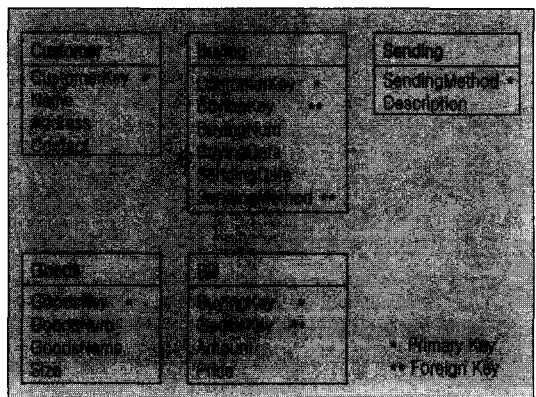


(그림 3) 직접적인 XML 문서 추출

간략화하여 사용한다. 생성될 DTD의 범위는 권고안[10]의 규칙처럼 업무 범위를 선정함으로써 선택될 수 있다. 여기에서는 구조에 있는 모든 정보를 업무범위로 한다. 그리고 모델링 순서는 권고안[10]의 순서로 진행하되, ID 추가 방식을 별도로 처리하지 않고 테이블간의 관계와 탐색 방향성을 처리하는 단계에서 병행으로 처리하도록 한다.

3.1 업무범위와 테이블의 기본키와 외부키 결정

다음 그림은 처리될 데이터베이스의 테이블의 구조를 보여주고 있으며, 업무범위는 5개의 테이블을 사용하는 것으로 가정하며, 기본키와 외부키는 다음 그림에서 보여주고 있다.



(그림 4) 업무범위와 테이블의 기본키, 외부키

3.2 DTD의 루트요소 생성

테이블의 데이터를 XML로 표현하기 위한 DTD의 루트 요소를 생성한다. <ShoppingData>를 루트 요소로 정했다. 그리고 BizTalk와 같은 XML 서버는 envelope라는 메커니즘을 제공하므로, 이와 같은 종류의 내용을 루트 요소에 속성으로 추가할 수 있다.

3.3 테이블 모델링 하기

테이블은 내용 테이블과 Lookup 테이블들로 구분되는데, 내용 테이블은 어떤 회사의 고객 주소와 같은 레코드들을 포함하는 테이블이며, Lookup 테이블은 정보를 좀 더 세밀히 분류하기 위해 내용 테이블의 각 정보에 대해 상세 설명을 저장하는 테이블들을 말한다. 그리고 일반적으로 내용 테이블은 XML문서에서 하나의 요소(Element)에 해당되므로 업무범위의 5개의 테이블 중에서 4개의 테이블(Customer, Buying, Goods, Bill)를 루트 요소의 자식 요소로 삽입하는 과정을 모델링한다. Lookup 테이블(Sending)은 외부키의 존재 여부에 따라 속성으로 처리되기 때문에 테이블이지만 테이블 모델링에서 처리될 수 없으며, 본 논문에서는 컬럼 모델링이 속성으로 모델링되므로 이 과정에서 Lookup 테이블을 병행으로 처리한다.

3.4 루트요소 내용 모델링

이는 DTD의 루트요소에 대한 내용 모델을 정의하는 과정이다. 업무상 서로 통신하고자 하는 정보의 형태에 적절한 자식 요소들을 결정하는 것이다. 예를 들면 루트 요소를 결정하고 난 후 루트 요소의 내용으로 요소들이 구성 형태를 정하는 것이다. 모델링의 결과는 다음과 같다. 구매정보만을 원한다면 다음과 같이 모델링 될 것이다.

```
<!ELEMENT ShoppingData (Buying*)>
```

그리고 이 과정의 모델링을 속성 모델링보다 먼저 수행하는 이유는 속성 모델링 과정에서 Lookup 테이블을 병행해서 처리하기 위함이다.

3.5 속성 모델링

속성 모델링에는 외부키가 아닌 테이블의 컬럼들을 요소의 속성으로 모델링하고, 또한 외부키이면서 다른 테이블에 속한 컬럼의 세부 설명을 담고 있는 테이블들(Lookup 테이블)을 속성으로 모델링하고, 최후에 IDREF-ID 속성 관계 모델링을 처리한다.

3.5.1 테이블내의 컬럼/Lookup 테이블 모델링 하기

여기에서는 테이블의 외부키를 제외한 나머지 컬럼들을 모델링하는데 이는 테이블 모델링 할 때 루트 요소의 자식 요소로 삽입된 해당 요소의 속성으로 정의한다. 컬럼의 데이터의 형식, 또는 속성에 따라 #FIXED, CDATA, #REQUIRED, #IMPLIED를 적절하게 삽입하도록 한다. 그리고 테이블 모델링에서 언급한 것처럼 Lookup 테이블은 속성으로 처리할 수 있으므로 외부키를 가지고 있으면서 Lookup 테이블을 참조하면 외부키가 존재하는 테이블의 요소에 적절하게 속성을 추가해야 한다.

3.5.2 IDREF-ID 속성 관계 모델링 하기

본 논문에서 가장 중요한 부분이다. IDREF-ID 속성 관계는 일반적으로 테이블 내의 외부키가 존재함으로써 발생하는 관계이다. 즉, 테이블들은 DTD를 생성할 때 하나의 요소로 맵핑될 수 있으며, 또한 테이블 간에는 서로 참조되는 관계들을 가질 수 있다. 결국, 테이블 간의 관계는 외부키에 의한 참조 관계로서 표현될 수 있으며, 이러한 테이블 간의 참조관계는 XML문서의 DTD 구조에서 데이터 간의 참조를 가능케 하는 IDREF-ID 속성 관계로서 표현하게 된다. 따라서, 모델링 방법은

작업 영역에 해당하는 테이블 간의 외부키 존재 여부와 테이블 간의 관계와 탐색 방향성을 결정해 내는 것으로 처리한다. 테이블 간의 관계 결정에서 1:1(일대 일), 1:n(일대 다)이면서 탐색방향성에서 목적지가 유일하면 해당하는 테이블 간의 관계는 포함관계로 처리하며, 방향성과는 관계없이 관계가 n:1(다대 일)인 경우에 부모 테이블은 IDREF 속성을 지니며 자식 테이블은 ID 속성을 지니도록 처리한다. 즉, 병행으로 ID를 처리하도록 한다. 즉, 포함관계, 외부키, 탐색방향을 파악하면 IDREF-ID 속성관계를 파악하는 것이 가능하다. 따라서, 루트 요소를 제외한 모든 요소에 ID를 사전에 미리 추가하는 작업은 배제해도 가능하다는 것을 알 수 있다. 아래 그림은 IDREF-ID 속성 관계 모델링을 위해 필요한 데이터를 파악한 상황을 보여주고 있는데, 여기에서 사전에 모델링 된 컬럼들은 해당사항이 없으므로 표현에서 제외했다.

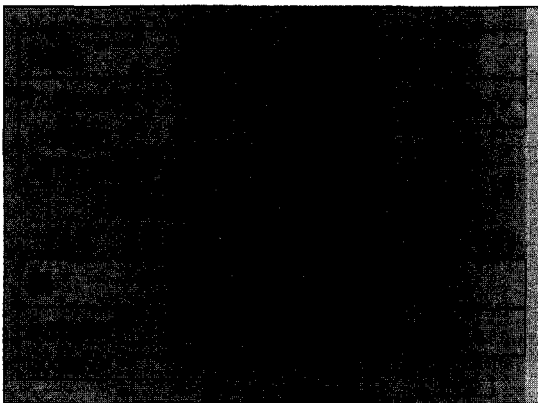
그리고 테이블을 하나의 요소로 처리하며 컬럼

(데이터 포인트)은 요소의 속성으로 처리한다. 속성이 아닌 요소의 내용으로 컬럼을 처리하고자 한다면 컬럼들에 대해 요소로 생성하고 테이블 요소에 추가해 주면 가능하다.

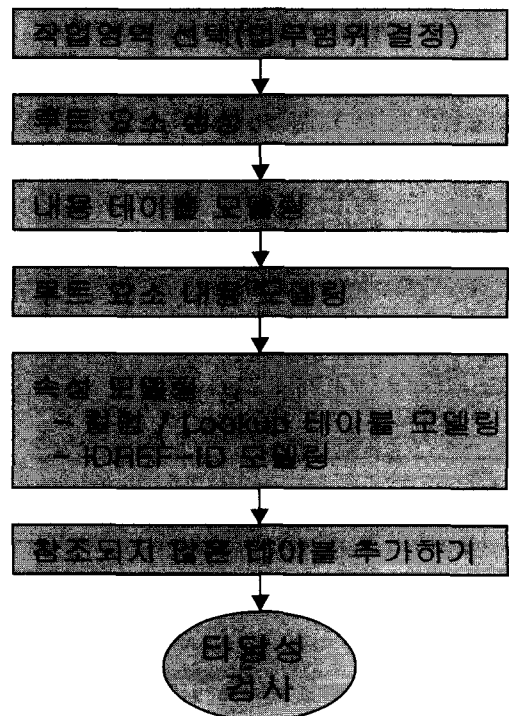
3.6 참조되지 않는 테이블 추가하기

참조되지 않는 테이블이란 구조안의 다른 어떠한 요소의 요소 내용 모델로 나타날 수 있는 것이 아니며 이러한 이유는 참조가 되지 않고 있어서 포함되지 않았기 때문이다. 위 그림 5에서 Buying과 Bill 테이블의 관계가 Buying 요소 안에 Bill 요소가 속하는 형식이 되고 Customer 테이블과 Goods 테이블은 서로 어떤 요소안에 포함될 성격이 아니기 때문에 이 두 개의 요소를 루트 요소에 추가하게 된다.

다음 그림은 DTD의 IDREF-ID 속성 관계 모델링 처리과정을 정리하여 보여준다.



(그림 5) 테이블 간의 관계성, 탐색방향성(테이블 간 관계)



(표 1) 관계성과 탐색방향성에 따른 포함 및 IDREF-ID 속성 관계

탐색 방향성과 관계성에 따른 요소 관계	1:1(일대일)	1:n(일대다)	n:1(다대일)
	포함관계		IDREF-ID 속성관계
관계성(n:1)에 따른 요소 관계	부모테이블	자식테이블	
	IDREF	ID	

(그림 6) DTD의 IDREF-ID 속성 관계 모델링 과정

3.7 제안된 방식에 의해 생성된 DTD의 타당성 검증

제안된 방식의 처리과정을 통해 예로 보인 온라인 쇼핑몰 관련 데이터베이스의 정보에 해당되는 전체 문서의 DTD 구조는 아래와 같이 모델링되며, DTD에서 이탤릭체로 표시된 부분은 제안된 모델링 방법을 이용하여 생성된 IDREF-ID 속성 관계가 되고 있다.

```
<!ELEMENT ShoppingData (Buying*, Customer*, Goods*)>
<!ATTLIST ShoppingData
  info (N_Version | U_Version | Copied) #REQUIRED>
<!ELEMENT Buying (Bill*)>
<!ATTLIST Buying
  BuyingNum CDATA #REQUIRED
  BuyingDate CDATA #REQUIRED
  SendDate CDATA #REQUIRED
  SendingMethod (Usps | Fedex | UPS) #REQUIRED
  CustomerIDREF IDREF #REQUIRED >
```

```
<!ELEMENT Bill EMPTY>
<!ATTLIST Bill
  GoodsIDREF IDREF #REQUIRED
  Amount CDATA #REQUIRED
  Price CDATA #REQUIRED>
```

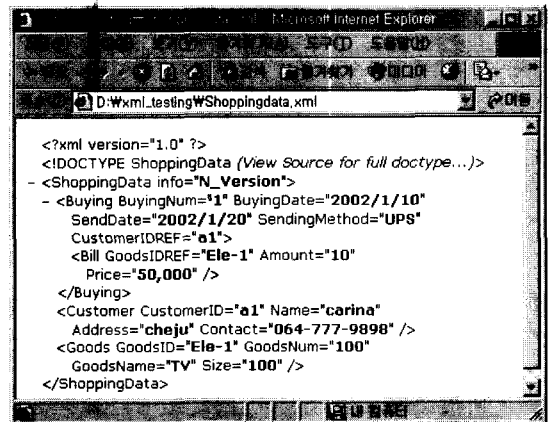
```
<!ELEMENT Customer EMPTY>
<!ATTLIST Customer
  CustomerID ID #REQUIRED
  Name CDATA #REQUIRED
  Address CDATA #REQUIRED
  Contact CDATA #REQUIRED>
```

```
<!ELEMENT Goods EMPTY>
<!ATTLIST Goods
  GoodsID ID #REQUIRED
  GoodsNum CDATA #REQUIRED
  GoodsName CDATA #REQUIRED
  Size CDATA #REQUIRED>
```

그리고 위의 DTD 구조를 따르는 XML 문서는 아래와 같다.

```
<?xml version="1.0"?>
<!DOCTYPE ShoppingData SYSTEM "shoppingdata.dtd">
<ShoppingData info="N_Version">
  <Buying
    BuyingNum="1"
    BuyingDate="2002/1/10"
    SendDate="2002/1/20"
    SendingMethod="UPS"
    CustomerIDREF="a1">
    <Bill GoodsIDREF="Ele-1"
      Amount="10"
      Price="50,000" />
  </Buying>
  <Customer CustomerID="a1"
    Name="carina"
    Address="cheju"
    Contact="064-777-9898" />
  <Goods GoodsID="Ele-1"
    GoodsNum="100"
    GoodsName="TV"
    Size="100" />
</ShoppingData>
```

생성된 DTD에 기반한 XML 문서의 결과 모델링은 아래와 같다.



(그림 7) 모델링 결과

4. 결론

본 논문에서는 Legacy 데이터베이스로부터 얻은

데이터를 XML 문서로 저장할 때 참고할 DTD 생성 과정에서 테이블간의 복잡한 참조관계를 IDREF-ID 속성 관계로 모델링 하였다. ADO나 SQL Server 2000은 직접적인 XML 문서를 추출해 버리는 반면 참고할 DTD는 생성하지 못하고 있으며, 이러한 원인은 테이블간의 복잡한 조인(Join)관계에 기인하고 있다. 그래서 복잡한 테이블간의 관계를 기존 방식보다 생성 단계를 줄이면서 DTD내에 표현할 수 있는 방법을 본 논문에서 모델링 하였으며, 그 결과 다음과 같이 기존 규칙들이 배제됨을 알 수 있었다.

- ▶ XML 구조에서 루트 요소를 제외한 모든 요소에 ID 속성을 추가한다.
- ▶ 참조되지 않는 불필요한 ID 속성을 삭제한다.

한 편으로, Legacy 데이터베이스에서 추출되는 데이터를 위한 DTD는 표현하고자 하는 정보의 종류에 따라 DTD 구조는 달라질 수 있다. 즉, 적용할 업무에 따라 선택되는 테이블의 종류가 상이할 수 있고, 또한 선택될 필드들도 고정적일 수 없기 때문이다. 필요한 데이터들만 최적화 해야 한다면 응용에 따른 DTD는 여러 가지가 생겨날 수 있다. 그래서 제안된 모델링 방식의 기준을 따라서 참조할 DTD를 생성하고 이를 이용함으로써 다음과 같은 기대 효과를 보일 수 있다.

- ▶ 인터넷 또는 인트라넷 상의 다른 시스템과 업무 데이터를 공유할 수 있다.
- ▶ 이질적인 시스템간에 정보를 처리할 수 있다.
- ▶ 레거시(legacy) 데이터를 XML을 이용하는 애플리케이션에 제공할 수 있다.
- ▶ B2B 등의 기업간 거래 시 참고 문서로 활용할 수 있다.
- ▶ 콘텐츠 배급을 위한 기본 문서 구조로 활용할 수 있다.

향후 연구과제로는 제안된 모델링의 구현과,

멀티미디어 타입의 속성을 가지는 관계형 데이터베이스의 필드들을 DTD 내로 표현하고자 한다.

참고 문헌

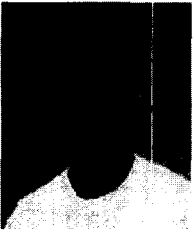
- [1] 손정환, 이희주, 장재우, 심부성, 주종철 “구조화된 문서를 위한 정보검색시스템의 설계 및 구현”, '98 동계 데이터베이스 학술대회 논문집 제14권 1호, pp. 102~106, 1998.
- [2] 연제원, 장동준, 김용훈, 이강찬, 이규철. “효율적인 검색 지원 SGML 저장 관리기의 설계 및 구현”, '99 한국 데이터베이스 학술대회 논문집 15권 1호, pp. 136~143. 1999.
- [3] 유재수의 8명, “전자도서관 표준문서관리를 위한 XML 저장관리기 기술 개발”, 케이오텍 최종보고서, 1999.
- [4] Charles L. A. Clarke, Gordon V. Cormack, Forbes J. Burkowski “An Algebra for Structured Text Search and a Framework for its Implementation. The Computer Journal 38(1)”, pp. 43~56, 1995.
- [5] Dongwook Shin, Hyuncheol Jang, and HongLan Jin “Bus : An Effective Indexing and Retrieval Schema in Structured Documents”, ACM. pp. 235~243, 1998.
- [6] Francois. “Generalized SGML repositories: Requirements and Modeling”, Computer Standards & Interfaces, 1996.
- [7] Tuong Dao, Ron Sacks-Davis, James A.Thom. “An indexing scheme for structured documents and its implementation”, Proceedings of the 4th International Conference on DATABASE Systems for Advanced Applications, Melbourne, Australia. pp. 125~135, 1997.
- [8] 맹성형, 주종철 “문서 구조화와 정보 검색”, 정보과학회지, 제16권, 제8호 1998. 8.
- [9] 석호, “데이터베이스 시스템” 정익사, 2000
- [10] Kevin Williams 외 9인 공저 “Professional XML

Databases” Wrox, 2001.

[11] Brian Lowe, Justin Zobel, Ron Sacks Davis.

“A Formal Model for Databases of Structured Text”, DASFAA 1995, pp. 449~456.

◎ 저자 소개 ◎



김 정 희

1994년 제주대학교 정보공학과 졸업(학사)

1997년 제주대학교 대학원 정보공학과 졸업(석사)

2002년 제주대학교 대학원 정보공학과 박사과정

1998년~현재 : 제주산업정보대학 컴퓨터정보계열 겸임교수

관심분야 : XML, 데이터베이스, Internet Application, 프로그래밍 언어론, GIS

E-mail : carina@cheju.ac.kr



곽 호 영

1983년 홍익대학교 전자계산학과 졸업(학사)

1985년 홍익대학교 대학원 전자계산학과 졸업(석사)

1991년 홍익대학교 대학원 전자계산학과 졸업(박사)

1990년~현재 : 제주대학교 통신컴퓨터공학부 교수

관심분야 : 객체지향 프로그래밍, 프로그래밍 언어론, GIS

E-mail : kwak@cheju.ac.kr