XML 문서간의 스키마 통합 방법론*

강 석 호**·오 제 연**·허 원 창**

《目 次》

Abstract

본 연구에서는 서로 다른 데이터 구조를 갖는 XML 문서간의 통합적 관리 및 일관된 처리 환경을 제공하기 위한 XML 데이터 스키마 통합 방법론을 제안하였다. 본 연구에서 제안한 방법론은 객체지향 개념을 기본으로 하여 객체와 관계를 분리하는 데이터 모델을 정의하고 이 데이터 모델을 통하여 XML 문서들의 서로 다른 스키마들을 변환, 통합하는 체계적 접근법이다. 본 연구를 통해 독자적 데이터 구조를 지닌 XML 문서들에 일관된 통합 환경을 제공함으로써 기업 애플리케이션 통합(EAI: Enterprise Application Integration), 데이터 웨어하우스(DW: Data Warehouse), 기업간 전자상거래(B2B e-commerce) 분야에의 활용을 기대할 수 있다.

Ⅰ. 서 론

XML은 월드 와이드 웹(World Wide Web)의 새로운 문서 표준으로 제안된 이후 자유로운 확장성을 이유로 기업 정보 시스템, 전자상거래 등의 분야들에서 입출력 데이터 모델로 그 사용이 증가하고 있다. 따라서 기업은 애플리케이션 통합 또는 데이터 웨어하우스 등의 통합 시스템을 구축하기 위해서는 그 통합 대상으로 XML을 포함해야만 하게 되었다.

* 본 연구는 한국과학재단 목적기초연구(400-20020112)지원으로 수행되었음.

** 서울대학교 산업공학과

그러나 XML의 자유로운 확장성은 XML로 하여금 많은 분야에서 사용되게 한 반면 동시에 이기종 스키마간의 통합적 관리를 어렵게 한다는 문제를 발생시킨다. 때문에 XML을 통합 대상으로 포함하는 것은 쉽지 않다.

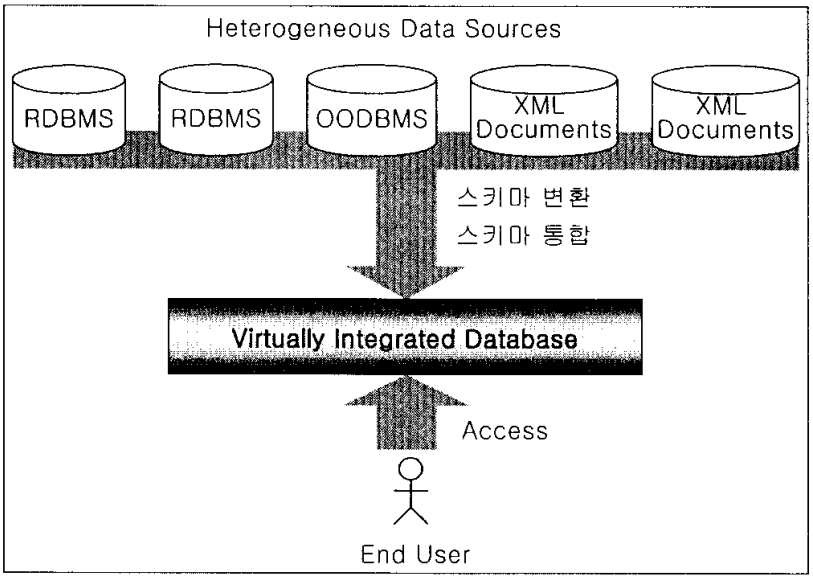
서로 다른 XML 문서들을 통합하는 과정에서 발생하는 세부적 문제들은 다음과 같다. 첫 번째는 XML 문서 스키마의 통합을 위한 통합 데이터 모델 제안이 미비하다는 것이다. XML 문서의 분석을 위한 데이터 모델[1] 또는 XML 문서의 효율적인 저장[2]에 관한 연구들이 활발하게 이루어지고 있으나, 서로 다른 XML 문서 스키마들의 통합을 위한 연구는 그리 많지 않다. 두 번째는 다수의 스키마들을 쉽게 통합할 수 있는 방법론의 연구가 미비하다는 것이다. 현재 널리 연구되고 있는 전자상거래 등의 환경에서는 수많은 데이터들의 통합 검색이 필요하지만, 이전의 데이터베이스 통합 방법론에서는 통합 대상의 수가 증가할수록 통합이 어려워진다[3]. 마지막으로 분산 환경에서의 통합적 사용을 지원하기 위해서는 XML 뿐만 아니라 현재 널리 사용되고 있는 관계형 데이터베이스의 스키마 또한 통합할 수 있는 통합 데이터 모델이 필요하다는 문제가 있다.

본 연구에서는 이러한 문제들을 해결하기 위한 XML 문서 스키마의 통합 방법론을 제시하였다. 이 방법론은 XML DTD의 엘리먼트, 애트리뷰트, 그리고 이들간의 포함 관계를 외형 객체, 속성 객체, 링크로 재구성한다. 그리고 재구성된 스키마들에 통합 규칙을 적용하여 하나의 스키마로 통합한다. 본 연구는 이기종 스키마들에 대한 통합 환경을 제공함으로써 기업 애플리케이션 통합, 데이터 웨어하우스, 기업간 전자상거래 등의 분야에 활용될 수 있다.

Ⅱ. 데이터베이스 통합과 XML

본 연구는 XML 문서의 스키마 통합을 다루고 있다. 서로 다른 스키마의 통합에 대한 연구는 데이터베이스 통합 분야에서 활발히 이루어졌다. 〈그림 1〉은 데이터베이스 통합의 일반적인 단계를 나타내고 있다[4]. 서로 다른 데이터 모델로 이루어진 분산 데이터베이스들을 통합하기 위해, 먼저 서로 다른 데이터 모델로 표현된 스키마들을 공통된 데이터 모델로 각각 변환한다. 그 다음, 공통된 데이터 모델로 변환된 스키마들을 통합 규칙에 따라 하나의 스키마로 통합한다.

〈그림 1〉 데이터베이스 통합



데이터베이스 스키마의 변환과 통합 방법론 외에도 통합 과정에서 발생하는 충돌을 해결하기 위한 연구들이 있다. [5]에서는 데이터베이스 통합 과정에서 발생하는 충돌을 6가지로 정의했으며, [6]에서는 이 충돌들을 다시 의미적 충돌, 구조적 충돌로 재분류하였다.

본 연구에서는 이러한 데이터베이스 통합 분야의 선행 연구들을 기초로 하여 XML 문서 스키마를 통합하기 위한 스키마 통합 방법론을 제안한다.

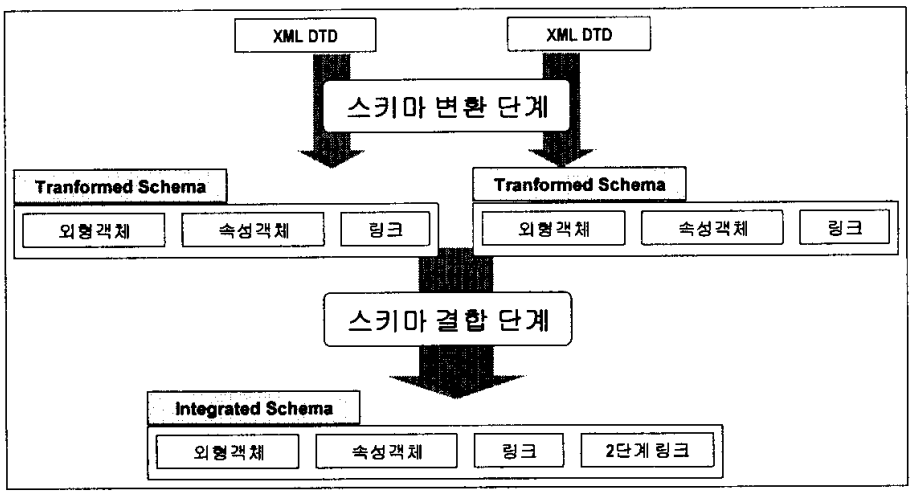
XML(eXtensible Markup Language)[7]은 W3C(World Wide Web Consortium)에 의해 제안된 웹 표준 문서이다. HTML(Hypertext Markup Language)은 데이터를 속성 관점으로 나타내기 힘들며, SGML(Standard Generalized Markup Language)은 사용하기 힘든 반면에, XML은 문서 스키마를 사용자가 정의할 수 있기 때문에 높은 확장성을 가지고 있다. 그래서 XML은 기업 정보 시스템, 전자상거래 등의 분야에서 응용 프로그램의 입출력 표준으로 자리잡고 있으며, 의료나 제조 등 특정 산업을 위한 XML 문서 스키마 표준화 작업들도 진행되고 있다[8].

XML의 스키마를 표현하는 방법으로 XML DTD와 XML Schema가 널리 사용되고 있다. XML Schema는 높은 표현력으로 인해 주목받고 있지만, 본 연구에서는 많은 응용 프로그램들이 사용하고 있는 XML DTD를 통합 대상으로 사용하였다.

Ⅲ. XML 문서의 스키마 통합

이 장에서는 본 연구에서 제안하는 XML 문서의 스키마 통합 방법론을 설명한다. 먼저 통합 데이터 모델(CDM: Common Data Model)을 정의하고, 이 모델을 사용하는 XML 문서의 스키마 통합 과정을 설명한다. 〈그림 2〉는 스키마 통합 과정을 보여주고 있다. 이 과정은 XML DTD로 주어진 스키마들을 이 통합 데이터 모델로 바꾸는 스키마 변환 단계, 변환된 스키마를 단일한 스키마로 결합하는 스키마 결합 단계로 구성된다.

〈그림 2〉 본 연구의 XML 문서 스키마 통합 과정



3.1 통합 데이터 모델(CDM: Common Data Model)

이 절에서는 XML 문서의 스키마 통합을 위한 통합 데이터 모델을 설명한다. 이 통합 데이터 모델은 첫 번째로 확장이 자유로운 XML 문서 스키마의 복잡한 구조를 표현할 수 있어야 하고, 두 번째로 통합 대상의 수가 증가해도 구조의 복잡성은 증가하지 않아야 하며, 세 번째로 현재 데이터 저장 구조로 가장 많이 사용되는 관계형 모델도 통합 대상으로 포함할 수 있어야 한다.

XML 문서 스키마의 구조를 표현할 수 있게 하기 위해 본 연구의 통합 데이터 모델은 개체의 복잡한 구조를 가장 잘 나타낼 수 있는 객체지향 개념을 근간으로 한다. XML DTD의 엘리먼트들이 각각 하나의 객체에 대응되는 방식을 따른다.

통합 대상의 수가 증가해도 통합의 복잡성은 증가하지 않게 하기 위해 본 연구에서는 객체와 관계를 분리한다. 객체와 관계가 분리되어 관리됨으로써 객체와 관계의 수가 각각 증가해도 서로에게는 영향을 미치지 않도록 한다.

관계형 모델의 통합도 지원할 수 있도록 하기 위해 본 연구에서는 통합모델의 객체 개념의 종류를 두 가지로 제한한다. 외형 객체와 속성 객체가 그 두 가지이며, 두 객체 종류와 관계의 개념을 통해 관계형 데이터 모델의 개념을 모두 지원하며 관계형 데이터 모델을 통해 표현될 수 없는 개념 또한 없도록 한다. 〈표 1〉은 통합 데이터 모델의 구성 요소들을 정리하고 있다.

〈표 1〉 통합 데이터 모델의 구성 요소

외형 객체 (Exterior)	정의	Exterior(String namd, Interior[] interiorSet)
	속성	name - 해당 외형 객체의 이름 interiorSet - 해당 외형 객체에 속하는 속성 객체들의 집합
속성 객체 (Interior)	정의	Interior(String name, Cardinality card, Data value)
	속성	name - 해당 속성 객체의 이름 card - 해당 속성 객체가 외형 객체에 속하는 카디널리티 value - 해당 속성 객체의 인스턴스의 값
링크 (Link)	정의	Link(Exterior[] exteriorSet, Cardinality card)
	속성	exteriorSet - 링크가 잇고 있는 외형 객체의 쌍 card - 외형 객체 포함 관계의 카디널리티

• 외형 객체(Exterior)

외형 객체는 하나의 개체를 나타내지만 속성을 직접 가지지 않는 객체이다. 개체의 속성과 그 값은 속성 객체가 갖도록 하고 이 속성 객체들을 포함하는 기능을 지닌다. 〈표 1〉에서와 같이 외형 객체는 자신의 이름과 자신이 갖는 속성을 나타내는 속성 객체들의 집합으로 이루어진다.

하나의 외형 객체는 일반적으로 XML DTD의 엘리먼트에 대응하며 관계형 데이터 모델에서는 테이블에 대응함과 동시에 하나의 레코드를 나타내기도 한다.

• 속성 객체(Interior)

속성 객체는 개체에 속하는 속성들 중 하나의 값을 가지는 객체이다. 〈표 1〉에서와 같이 자신의 이름과 자신이 속하는 외형 객체와의 카디널리티, 그리고 문서 인스턴스의 값으로 구

성된다.

속성 객체는 XML DTD의 엘리먼트 또는 애트리뷰트에 대응하며 관계형 데이터 모델에서는 필드에 대응한다.

• 링크(Link)

링크는 외형 개체 사이의 관계이다. 외형 객체와 속성 객체의 관계는 외형 객체가 속성 객체들의 집합을 포함함으로써 나타내고 있다. 〈표 1〉에서와 같이 링크는 두 가지 요소로 구성되는데, 첫 번째는 자신이 잇는 외형 객체의 쌍이고, 두 번째는 대응 관계의 카디널리티이다.

링크는 XML DTD에서 엘리먼트의 포함 관계에 대응하며, 관계형 데이터 모델에서는 테이블간의 참조 관계에 대응한다.

3.2 XML 스키마 통합 과정

XML 문서 스키마의 통합 방법론은 데이터베이스 통합의 일반적인 과정을 따른다. 먼저 XML DTD들을 3.1에서 제시된 통합 데이터 모델로 변환하고, 통합 데이터 모델로 변환된 두 개의 스키마를 하나로 결합한다. 스키마의 통합은 먼저 객체를 나타내는 외형 객체들을 통합하고, 관계를 나타내는 링크들을 통합한다. 여기에서는 XML DTD의 통합 데이터 모델로의 변환, 외형 객체들의 통합, 링크들의 통합을 순서대로 설명한다.

• 스키마 변환

스키마 변환 과정은 주어진 XML DTD를 3장에서 언급한 통합 데이터 모델로 변환된 스키마를 출력한다.

먼저 XML DTD의 엘리먼트와 애트리뷰트들을 외형 객체와 속성 객체로 변환한다. 자신의 하위 엘리먼트 또는 애트리뷰트를 가지는 엘리먼트들에 대해 외형 객체를 정의한다. 그렇지 않은 엘리먼트들과 모든 애트리뷰트들에 대해 속성 객체를 정의한다.

다음으로 모든 외형 객체들이 최상위 객체가 되도록 외형 객체들 간의 관계를 분리한다. 이 관계를 링크로 정의한다.

변환 과정에서 자신이 최상위 객체인 속성 객체가 발생할 수 있다. 이 경우 하나의 외형 객체를 정의하여 속성 객체가 포함되도록 한다.

스키마 변환 과정의 알고리즘은 다음과 같다.

(1) 외형 객체의 생성

For *Each Element e*

If *e has sub element or e has attribute* **Then**

e → Exterior(e.name,)

Else

e → Interior(e.name,)

End If

Next

For *Each Attribute a*

a → Interior(a.name,)

Next

(2) 링크의 생성

For *Each Exterior e*

If *a Exterior e′ is parent of e* **Then**

l = Link(e′, e,)

End If

Next

(3) 속성 객체의 생성과 처리

For *Each Interior i*

If *an Exterior e is parent of i* **Then**

add(e.interiorSet, i)

End If

If *i has no parent* **Then**

i → Exterior(i.name,)

i′ = Interior(i.name′,)

add(i.interiorSet, i′)

End If

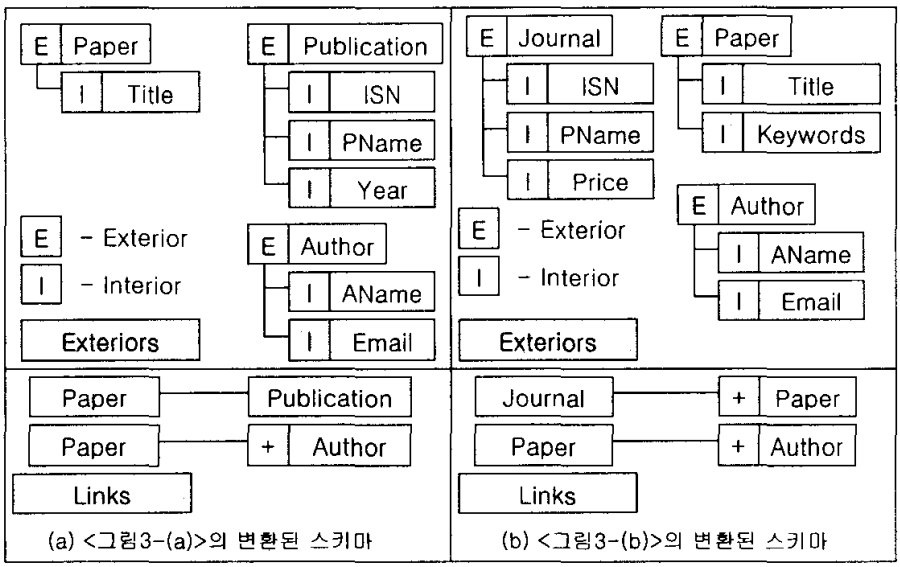
Next

〈그림 3〉, 〈그림 4〉에서 스키마 변환 과정의 예제를 보여준다. 〈그림 3〉은 통합 대상이 되는 두 개의 XML DTD 문서들을 보여준다. 〈그림 4〉는 이 XML DTD들이 각각 스키마 변환 과정을 거쳐 통합 데이터 모델로 변환된 결과를 보여준다.

〈그림 3〉 통합될 XML DTD 문서들

<!ELEMENT Paper(Title, Publication, Author+)> <!ELEMENT Title(#PCDATA)> <!ELEMENT Publication(PName, Year)> <!ATTLIST Publication ISN CDATA #REQUIRED> <!ELEMENT PName(#PCDATA)> <!ELEMENT Year(#PCDATA)> <!ELEMENT Author(AName, Email?)> <!ELEMENT Aname(#PCDATA)> <!ELEMENT Email(#PCDATA)> (a) Paper를 나타내는 XML DTD	<!ELEMENT Journal(ISN, PName, Price, Paper+)> <!ELEMENT ISN(#PCDATA)> <!ELEMENT PName(#PCDATA)> <!ELEMENT Price(#PCDATA)> <!ELEMENT Paper(Title, Keywords*, Author+)> <!ELEMENT Title(#PCDATA)> <!ELEMENT Keywords(#PCDATA)> <!ELEMENT Author(AName, Email?)> <!ELEMENT Aname(#PCDATA)> <!ELEMENT Email(#PCDATA)> (b) Journal을 나타내는 XML DTD

〈그림 4〉 통합 데이터 모델로 변환된 스키마들



• 외형 객체들의 통합

외형 객체의 통합은 외형 객체간의 조화 관계를 정의한 후 조화 관계에 따라 외형 객체들을 통합하는 과정으로 이루어진다. 외형 객체들의 조화 관계 정의는 [3]에서 정리된 방법론을 따른다. 외형 객체들이 가질 수 있는 조화 관계는 '동일', '포함', '교집합 존재', '관계없음'이 있다. 이러한 관계들에 대해 외형 객체들의 대응 속성이 되는 속성 객체가 존재하고, 이를 WCI(with corresponding identifiers)라 한다. 또 외형 객체들의 속성들 중 중복되는 속성 객체들이 존재하고, 이를 WCA(with corresponding attributes)라 한다.

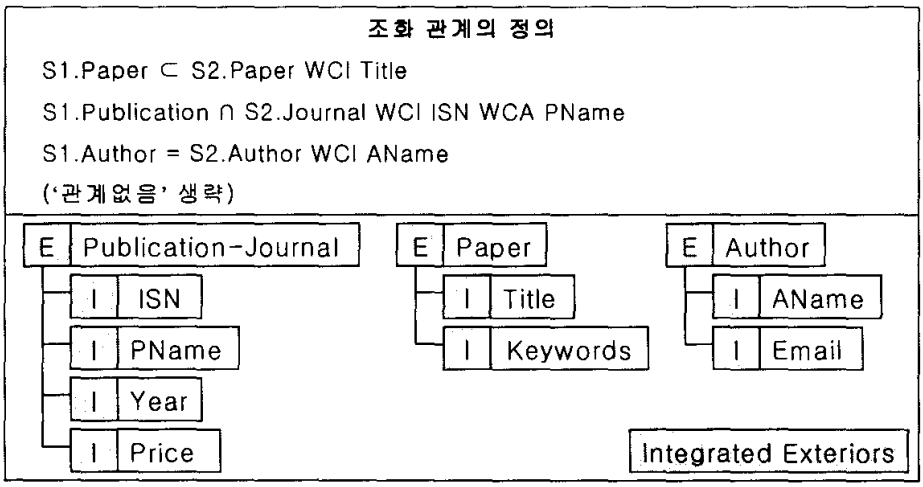
조화 관계에 따라 외형 객체들을 통합하는 방법은 다음과 같다. '동일', '교집합 존재', '포함'의 경우에는 두 외형 객체의 합집합을 통합 외형 객체로 정의한다. '관계없음'의 경우 통합할 필요가 없다. 이 과정에서 WCI인 속성 객체, WCA인 속성 객체, WCI도 WCA도 아닌 속성 객체에 대해서는 질의 처리를 위한 매핑 테이블이 별도로 작성되어야 한다.

외형 객체의 통합 알고리즘은 다음과 같다.

For *Each couple of Exterior e1, e2*
 If $e1 \neq e2$ *Then*
  Continue
 Else If $e1 \cap e2$ *WCI i1, WCA interiorSet1* **Then**
  $e = Exterior(e.name, (i1 \cup interiorSet1 \cup e1.interiorSet \cup e2.interiorSet))$
 Else If $e1 \supseteq e2$ **Then**
  $c = Exterior(c1)$
 End If
Next

〈그림 5〉는 외형 객체들의 통합 과정과 결과를 나타낸다. 〈그림 4〉의 두 스키마에 외형 객체 통합 과정을 적용한 결과이다.

〈그림 5〉 외형 객체들의 통합



• 링크들의 통합

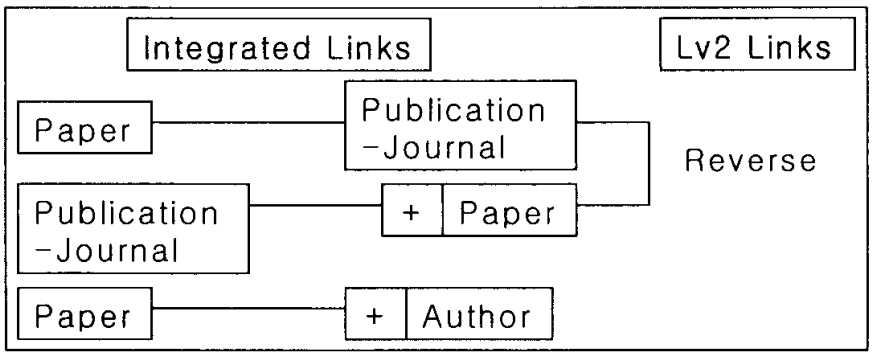
외형 객체 통합 과정은 객체를 변화시킴으로써 객체를 잇는 링크도 변화시킨다. 다시 말하면, 이 과정에서 링크들은 연결 객체 또는 카디널리티가 바뀌거나, 링크 자체가 삭제될 수도 있다. 링크 통합 과정에서는 이 변화 과정을 거친 링크들을 통합한다. 링크들의 통합은 중복되는 링크를 제거함으로써 간단히 이루어진다.

이 과정에서 외형 객체, 속성 객체, 링크만으로는 해결할 수 없는 문제들이 몇 가지 발생한다. 첫 번째는 반대의 포함관계를 가지는 링크의 쌍이 존재할 수 있으나 보다 유연한 통합을 위해 둘 중 하나를 제거할 수 없고 통합 스키마에서 이 문제를 표현해 주어야 한다는 것이다. 두 번째는 링크가 외형 객체의 포함관계 중 DTD에서 나타날 수 있는 OR 관계나 집합 관계를 표현할 수 없다는 것이다. 그러나 이러한 문제들을 해결하기 위하여 객체나 링크의 정의를 복잡하게 만드는 것은 본 연구의 의도인 구조적 통합의 단순화에 어긋난다.

본 연구에서는 이를 해결하기 위해 링크의 관계를 나타내는 2단계 링크를 정의하였다. 2단계 링크는 대상이 되는 링크들과 관계 속성으로 정의된다. 관계 속성은 링크들 간의 역관계, OR 관계, 집합 관계를 표현하며, 객체와 관계로 변환할 수 없는 모든 정보들을 포함한다.

〈그림 6〉은 〈그림 4〉로부터 링크 통합 과정에 의해 통합된 링크들을 나타낸다.

〈그림 6〉 링크들의 통합



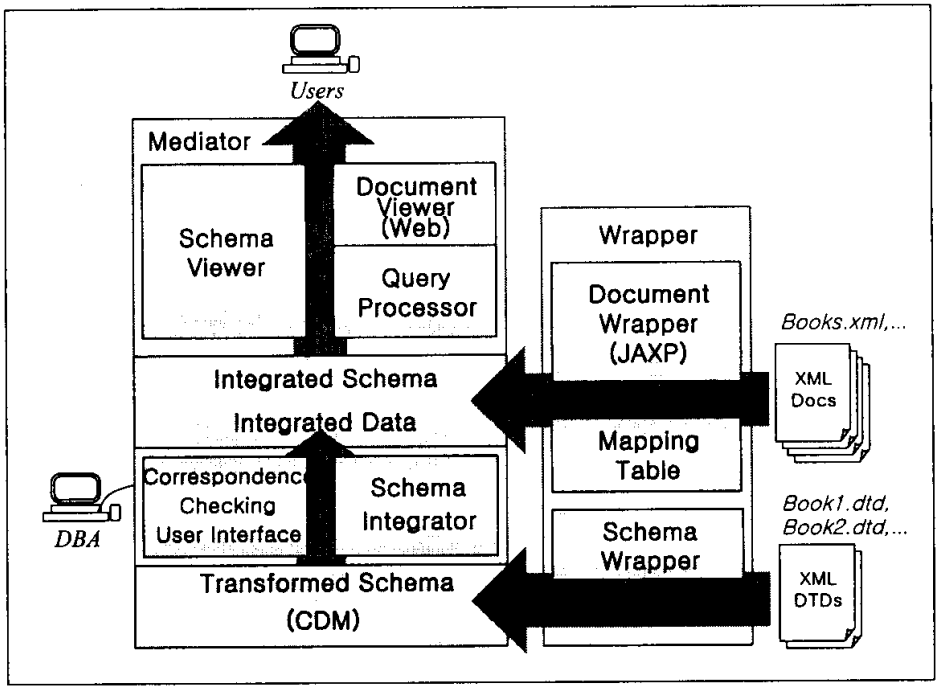
Ⅳ. XML 문서 통합 시스템의 설계

본 연구가 적용된 XML 문서 통합 시스템은 데이터베이스 통합 시스템의 일반적 형태인 Mediator-Wrapper Approach를 따른다.

〈그림 7〉은 시스템의 구조를 나타낸다. 그림의 오른쪽과 같이 통합 대상이 되는 XML DTD와 XML 문서가 주어졌다고 하자. 그 중 XML DTD는 스키마 통합 과정에서 스키마 래퍼(Schema Wrapper)에 의해 통합 데이터 모델(CDM)로 변환된다. 이 CDM은 데이터베이스 관리자(DBA)에 의해 수동으로 조화 관계(correspondence)가 정의되고, 미디에이터(Mediator) 내의 스키마 통합기(Schema Integrator)를 통합 과정을 거쳐 스키마 뷰어(Schema Viewer)를 통해 사용자에게 제공된다.

XML 문서는 도큐먼트 래퍼(Document Wrapper)와 스키마 변환/통합 과정에서 생성된 대응 관계 테이블을 통해 통합 스키마의 인스턴스로 파싱되어 도큐먼트 뷰어(Document Viewer)를 통해 사용자에게 제공된다.

〈그림 7〉 XML 문서 통합 시스템 구조



V. 결 론

본 연구에서는 서로 다른 데이터 구조를 갖는 XML 문서간의 스키마 통합 방법론을 제시하였다. 본 방법론의 장점은 통합 대상의 수에 관계없이 단순함을 유지하는 통합 절차의 명료성과 XML 이외의 다양한 데이터 모델들도 통합 대상으로 지원할 수 있는 확장성이라 할 수 있다. 이 장점들을 지원하기 위해 객체와 관계를 분리하는 통합 데이터 모델을 제시하고, 이를 바탕으로 하는 스키마 변환과 통합 과정을 제시하였다.

이러한 연구 결과를 통하여 기업 애플리케이션 통합, 데이터 웨어하우스, 기업간 전자상거래에서 요구되는 XML 문서 통합 환경을 제공할 수 있을 것으로 기대된다.

본 연구는 XML 문서 스키마의 구조적 통합 방법론의 구축에만 초점을 두었다. 추후 연구 과제로는 크게 스키마의 의미적 통합 방법론 구축과 질의 처리 프로세스 구축의 두 가지를 들 수 있다. 스키마의 의미적 통합 방법론을 구축함으로써 본 연구에서 제시된 통합 과정 중 조화 관계 정의 과정을 자동화 시킬 수 있다. 질의 처리 프로세스에 대한 연구는 본 연구에서 제안한 통합 모델의 단순성과 결합되어 질의의 복잡성을 높일 수 있을 것으로 기대된다.

참 고 문 헌

1. W3C, 'Document Object Model(DOM) Technical Reports, http://www.w3c.org/DOM/DOMTR/
2. Masatoshi Yoshikawa, Toshiyuki Amagasa, Takeyuki Shimura, Shunsuke Uemura, 'XRel: a path-based approach to storage and retrieval of XML documents using relational databases,' ACM Transactions on Internet Technology 1(1), 2001
3. Samuel Robert Collins, Shamkant Navathe, 'XML schema mappings for heterogeneous database access,' Information and Software Technology 44, 2002
4. Christine Parent, Stefano Spaccapietra, 'Issues and approaches of database integration,' Communications of the ACM, 1998
5. Chiang Lee, Chia-Jung Chen, Hongjun Lu, 'An aspect of query optimization in multidatabase systems,' ACM SIGMOD RECORD 24(3), 1995
6. Frank S.C. Tseng, Jeng-Jye Chiang, Wei-Pang Yang, 'Integration of relations with conflicting schema structures in heterogeneous database systems,' Data & Knowledge Engineering 27, 1998
7. W3C, 'Extensible Markup Language(XML),' http://www.w3c.org/XML/
8. Cover Pages, 'XML Applications and Initiatives,' http://xml.coverpages.org/xmlApplications.html