스키마가 없는 XML 문서에서의 재사용 가능한 XML Schema 추출 기법

조 정 길[†] · 구 연 설[††]

요 약

웹의 발전으로 인터넷상에 XML 문서의 양이 증가하면서 클라이언트에서 수신된 XML 데이터를 검증하고 데이터베이스에 효율적으로 저장하고 질의하는데 필요한 많은 연구들이 진행되고 있다. 이러한 작업들을 처리하기 위해서는 XML 문서의 DTD나 XML Schema가 필요하다. 그러나 스키마가 없는 XML 문서는 DTD나 XML Schema가 없기 때문에 이러한 작업들을 처리할 수가 없다. 이에 이 논문에서는 well-formed XML 문서이거나 XML Schema가 없는 XML 문서에서 XML 데이터를 검증하고 데이터베이스에 효율적으로 저장하고 질의할 수 있도록 XML Schema를 추출한다. 이 논문에서 제안하는 XML Schema 추출 방법은 반구조적 데이터의 스키마 추출 기법인 데이터가이드와 시뮬레이션을 적용하여 스키마가 없는 XML 문서에서 스키마 그래프를 추출한다. 그리고 스키마 그래프와 재사용성을 고려한 패턴 테이블을 이용하여 XML Schema를 추출하는 기법을 제안한다.

Technique for extracting reusable XML Schema from schema-less XML Documents

Jung-Gil Cho[†] · Yeon-Seol Koo[††]

ABSTRACT

According to development of Web, an amount of XML documents has been increasing. So, many researches are proceeding to verify XML data coming from clients and to store or query efficiently database. In order to verify, store and query, DTD or XML Schema of XML documents is necessary. However, Schemaless XML documents couldn't be operated since they do not have either DTD or XML Schema. In this paper, we extract XML schema in order to verify XML data and store or query efficiently database from either well-formed XML or XML Schemaless documents. XML Schema extracting technique which is proposed in this paper extract Schema graph using simulation and dataguide that is a extracting technique for semistructured characteristics of XML data. Also, we propose extracting technique for XML Schema using pattern tables that are considerated with Schema graph and reusability.

키워드 : XML, well-formed XML 문서, XML Schema 추출, 데이터가이드(Dataguide), 시뮬레이션(Simulation)

1. 서 론

인터넷 문서 표준인 XML을 기반으로 한 전자 문서의 교환과 처리가 다양한 시스템에 응용되고 사용범위가 점차 넓어지고 있으며 각 분야에서 특성에 맞는 XML 문서의 표준이 정의되고 있다[1, 2]. XML 문서에는 well-formed XML 문서와 valid XML 문서가 있다. XML 문법에는 적합하나 적절한 XML Schema가 선언되지 않으면 well-formed XML 문서이고, 적합한 XML Schema가 선언된 후에 XML 문서에서 선언된 엘리먼트들이 사용되면 valid XML 문서이다. XML 문법 규칙에만 어긋나지 않는다면 XML Schema 없이도 XML 문서를 작성할 수가 있으며, well-formed XML 문서에서 적합한 XML Schema가 선언된다면 그 XML 문서는 valid XML 문서가 될 수 있다.

XML 문서에서 XML Schema는 스키마와 같은 역할을 하기 때문에 XML Schema에 정의된 내용대로 문서를 작성하고 전송하며 다른 사이트에서 수신 받은 XML 데이터를 검증하는데 사용한다. 또한 XML 데이터를 데이터베이스에 효율적으로 저장하고 질의하는 데에도 사용한다.

XML 문서는 새롭고 자유로운 태그를 사용하고 고정된 스키마가 없는 점에서 반구조적 데이터(semistructured data)의 성격을 가지고 있다. 반구조적 데이터란 데이터베이스의 스키마와 같이 데이터가 항상 일정한 형태를 가지는 것이 아니라 그 구조를 미리 예측할 수 없는 데이터를 말한다. XML 문서로부터 반구조적 스키마 추출기법을 이용한 스키마 트리의 추출은 상당한 장점이 있다. XML 문서를 관계형 데이터베이스에 저장할 경우에 스키마 트리를 이용하여

† 정 회 원 : 남서울대학교 컴퓨터학과 겸임교수
†† 정 회 원 : 충북대학교 컴퓨터과학과 교수
논문접수 : 2002년 6월 20일, 심사완료 : 2003년 3월 20일

효율적인 관계형 테이블 생성을 가능하게 하고, 또한 데이터베이스로부터 데이터를 XML 형태로 출력할 경우에 스키마 트리를 이용하여 XML 문서의 생성을 용이하게 해준다. 그리고 사용자가 질의를 할 경우에도 사용자에게 편리성을 제공해 준다.

따라서 이 논문에서는 스키마가 없는 XML 문서에서 XML 문서를 검증하고 데이터베이스에 효율적으로 저장하고 질의할 수 있도록 XML Schema를 추출한다. XML Schema가 DTD와는 비교할 수 없을 만큼 복잡한 구조를 가지고 있기 때문에 이 논문에서 제안하는 XML Schema 추출 기법은 반구조적 데이터의 스키마 추출 기법을 이용하며, 절대/상대 스키마 그래프로 스키마 그래프를 추출하고, XML Schema에서 중요한 데이터 형식을 변환하며, 재사용성을 고려하여 Venetian Blind Model을 이용한 정형화된 패턴 리스트로 XML Schema를 생성한다.

2. 관련 연구

2.1 스키마 추출 모델

OEM(Object Exchange Model)은 반구조적 데이터를 표현하기 위한 그래프 구조의 데이터 모델로서 객체라는 개념을 기본으로 반구조적 데이터를 모델링한다. 그러나 OEM은 XML 문서의 순서나 속성을 표현할 수가 없기 때문에 Lore의 XML 기반의 새로운 데이터 모델인 개선된 OEM 데이터 모델로 표현된다[10]. 개선된 OEM 데이터 모델에서 XML 엘리먼트는 <eid, value>쌍으로 표현되는데, eid는 유일한 엘리먼트 ID이고, value는 원자 텍스트 스트링(atomic text string)이거나 컴포넌트를 가지는 복합값이다.

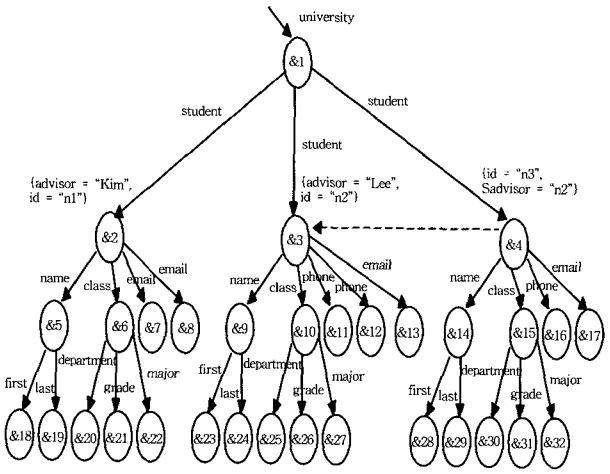
XML 문서는 순서와 방향성과 레이블이 있는 그래프인 개선된 OEM 데이터 모델로 변환될 수가 있다. 태그 엘리먼트 사이의 공백과 코멘트는 제외된다. 태그사이의 텍스트와 CDATA는 원자 텍스트 엘리먼트로 변환한다. 그외에 문서 엘리먼트는 다음과 같이 복합 데이터 엘리먼트로 변환한다. 첫째, 데이터 엘리먼트의 태그는 문서 엘리먼트의 태그이다. 둘째, 데이터 엘리먼트에 있는 속성이름/원자값 쌍의 리스트는 문서 엘리먼트의 속성 리스트로부터 직접 획득한다. 셋째, 문서 엘리먼트에 있는 타입 IDREF의 각 속성값 i 또는 타입 IDREFS의 속성값에있는 컴포넌트 i에 대하여 데이터 엘리먼트에 있는 하나의 교차연결 서브 엘리먼트 <label, eid>이다. label은 속성이름에 대응되고, eid는 ID 속성값에 i로 매치되는 유일한 데이터 엘리먼트로 간주한다. 넷째, 문서 엘리먼트의 서브 엘리먼트는 데이터 엘리먼트의 전형적인 서브 엘리먼트로서 순서적으로 나타난다.

노드는 데이터 엘리먼트를 표현하고, 간선(edge)은 엘리먼트와 자식 엘리먼트의 관계를 표현한다. 복합 데이터 엘리먼트를 표현하는 각각의 노드는 태그와 속성이름/원자값 쌍의 순서 리스트를 포함한다. 원자 데이터 엘리먼트 노드는 스트링 값을 가진다. 그래프에서 간선은 두 가지의 다른 타입이 있는데, 하나는 목적 자식 엘리먼트의 태그로 레이블된 일반적인 자식 엘리먼트 간선이고, 다른 하나는 교차연결을 받아들이는 속성 이름으로 레이블된 교차연결 간선이다.

(그림 1)은 샘플 XML 문서이고 (그림 2)는 개선된 OEM 데이터 모델로 표현한 그래프를 보여준다. eid는 노드안에 나타내고 &1, &2 등으로 쓴다. 속성이름/원자값 쌍은 IDREF 속성으로 ({ }로 묶은)연합된 노드로 보여준다. 자식 엘리먼트 간선은 실선이고, 교차연결 간선은 점선이다. 자식 엘리먼트의 순서는 왼쪽에서 오른쪽으로 나타낸다. 이 모델은 XML 문서의 엘리먼트를 노드에 들어오는 간선 상에 표시해줌으로써 XML 문서를 표현한다. 이렇게 변형된 개선된 OEM 데이터 모델을 통하여 최대/최소 경계 스키마를 추출할 수가 있으며, 이 두 스키마를 가지고 스키마 그래프의 추출이 가능해진다.

```
<university>
 <student advisor = "Kim" id = "n1" >
  <name>
   <first> Byungryul </first>
   <last> Lee </last>
  </name>
  <class>
   <department> Computer Science </department>
   <grade> 4 </grade>
   <major> Database </major>
  </class>
  <email> kim@korea.com </email>
  <email> kim@dreamwiz.com </email>
 </student>
 <student advisor = "Lee" id = "n2" >
  <name>
   <first> Junggil </first>
   <last> Cho </last>
  </name>
  <class>
   <department> Computer Science </department>
   <grade> 3 </grade>
   <major> Software Engineering </major>
  </class>
  <phone> 02-303-5678 </phone>
  <phone> 02-738-4830 </phone>
  <email> Cho@korea.com </email>
 </student>
 <student id = "n3" Sadvisor = "n2">
  <name>
   <first> Younki </first>
   <last> Cho </last>
  </name>
  <class>
   <department> Computer Science </department>
   <grade> 3 </grade>
   <major> Computer Network </major>
  </class>
  <phone> 02-322-4248 </phone>
  <email> Youn@Yahoo.co.kr </email>
 </student>
</university>
```

(그림 1) 샘플 XML 문서

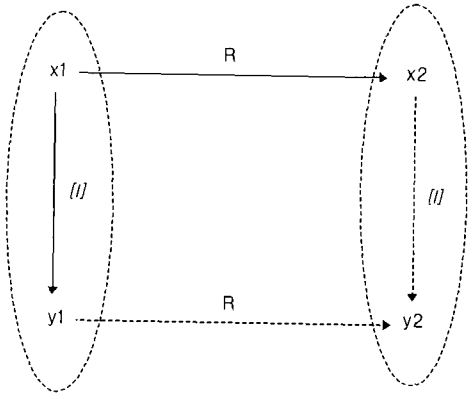


(그림 2) 개선된 OEM 데이터 모델

2.2 데이터가이드와 시뮬레이션

데이터가이드는 데이터베이스의 구조를 간결하고, 정확하고, 편리하게 나타내는 스키마 그래프로 정의된다. 또한 데이터가이드는 데이터 소스의 모든 유일한 레이블 경로를 데이터 소스에 나타나는 빈도에 상관없이 한번만 기술하며, 데이터 소스에 나타나지 않는 경로는 데이터가이드에 나타나지 않는다[5]. 데이터가이드에 대한 이러한 특성은 반구조적 데이터의 최대 경계 스키마 그래프 추출을 가능하게 해준다.

정점의 집합을 V라 하고, $E \subseteq V^2$을 만족하는 간선의 집합을 E, 그리고 정점 v를 label $\ll v \gg$로 매핑하는 함수를 $\ll \cdot \gg : V \rightarrow A$이라 할 때 레이블이 있는 그래프(labeled graph) $G = (V, E, A, \ll \cdot \gg)$로 나타낼 수가 있다. 이때 정점 v를 계승하는 정점(successor)들을 $post(v) = \{u | (v, u) \in E\}$라 하면 정점들의 집합상에서 이진 릴레이션(binary relation) $\leq \subseteq V^2$인 이진 릴레이션에 대해 $u \leq v$은 다음의 두 조건을 만족할 때 정점 v는 정점 u를 시뮬레이트(simulate)한다고 한다 : (1) $\ll u \gg = \ll v \gg$, (2) $u' \in post(u)$인 모든 정점에 대해 $u' \leq v'$이고 $v' \in post(v)$인 정점 v'가 존재한다. 이런 시뮬레이션을 이루기 위한 조건은 (그림 3)과 같다.



(그림 3) 시뮬레이션 다이어그램

주어진 데이터 그래프는 스키마 그래프에 대해 다른 여러 개의 시뮬레이션을 가지며 이것 중에 공통적인 것이 최소 경계 스키마 그래프이다. 이후부터는 용어의 일치를 위하여 정점을 노드라 칭한다.

2.3 재사용 가능한 XML Schema 설계

XML Schema의 구성요소들을 재사용 할 수 있다는 것은 매우 강력한 방법이다. 이러한 재사용은 내부 재사용과 교차 스키마 재사용의 두 가지 방식으로 이용할 수가 있다. 내부 재사용은 현재 작업하고 있는 XML Schema에 이미 정의된 구성요소를 재사용하는 방식이고, 교차 스키마 재사용은 다른 XML Schema에 전역으로 정의되어 있는 구성요소를 이용하는 방식이다[16].

내부 재사용은 모듈의 독립성을 요구한다. 모듈화 설계에서 모듈의 독립성인 결합도(coupling)와 응집도(cohesion)는 중요하다. 결합도는 모듈간의 상호 의존도를 말하며, 이 상호 의존도가 약해야 좋은 설계라고 할 수 있다. 또한 응집도는 한 모듈 내의 응집 정도를 말하며, 응집도가 강한 모듈은 하나의 기능을 수행하도록 설계한다. XML Schema에서 재사용성을 감안하여 엘리먼트나 데이터 형식을 선언하는 설계 방법에는 Salami Slice Design, Russian Doll Design, Venetian Blind Model이 있다.

Salami Slice Design 방법은 모든 엘리먼트나 형식을 전역으로 선언하는 방식이고, Russian Doll Design 방법은 모든 엘리먼트나 형식을 지역으로 선언하는 방식이다. 또한 Venetian BlindModel은 위 두 가지 방식을 적절히 사용하여 지역화시키고 최대의 재사용성을 얻는 설계 방식이다. <표 1>은 세 가지 설계 방법의 응집도, 결합도, 재사용성을 비교한 것이다.

〈표 1〉 설계 방법의 비교

	Salami Slice Design	Russian Doll Design	Venetian Blind Model
응 집 도	약함	강함	중간
결 합 도	강함	약함	약함
재사용성	중간	중간	높음

2.3 반구조적 데이터의 스키마 추출

OEM 모델과 XML 데이터 모델은 유사하여 Lore[9]에서 OEM 모델을 확장하여 XML을 지원하도록 하였으며 이를 XML 기반의 새로운 데이터 모델로 표현하였다. OEM 모델에서는 순서에 대한 정보가 없으나 XML은 각 엘리먼트들 간의 순서가 정해져 있기 때문에 개선된 OEM 데이터 모델에서는 순서에 대한 정보를 저장할 수 있는 리스트를 추가하여 XML을 지원하도록 하였다.

반구조적 데이터의 스키마 추출은 사용자 편의성, 저장 효율성, 질의 최적화 등의 필요성에 의하여 많은 연구가 이

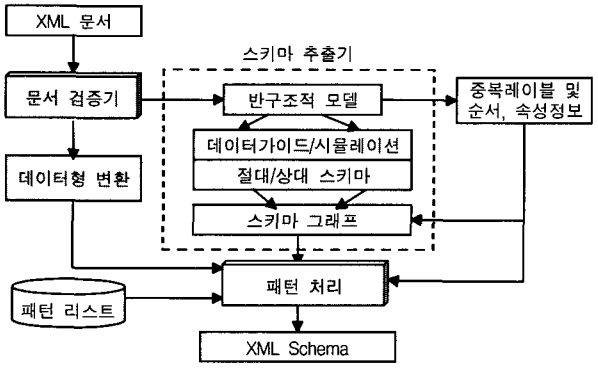
루어져 왔다. 데이터가이드는 Lore 프로젝트에서 소개한 반구조적 데이터 모델인 OEM을 이용하여 최대 경계 스키마 그래프를 추출하고 있다. 데이터로그는 클러스터링 방법을 이용하여 타입의 수를 줄이고 최대 고정점(fixpoint)을 이용하여 최소 경계 스키마 그래프를 추출할 수가 있다[6]. 그러나 데이터로그는 주어진 데이터 그래프의 노드를 방문할 때 모든 객체들이 분류된 초기 타입 릴레이션으로부터 그 릴레이션이 고정점에 도달할때까지 반복하여 검사하기 때문에 $O(n^2)$이라는 시간이 걸리므로 데이터가이드에 비하여 시스템에 많은 부담을 주게된다. 따라서 시뮬레이션은 초기화 단계에서는 $O(n^2)$의 시간이 소요되지만 remove(v)를 이용하여 sim(v)의 원소를 제거하기 때문에 데이터로그보다 성능이 우수하다[7].

[7, 11]에서는 데이터가이드와 시뮬레이션 방법을 이용하여 최대/최소 경계 스키마 그래프를 추출하여 DTD를 생성하였다. 그러나 데이터 중심의 XML 문서위주이기 때문에 문서 중심의 XML 문서에서 중요시되는 엘리먼트간의 순서와 속성은 고려하지 않았다. XTRACTOR[12]에서는 반구조적 데이터에서 추출한 스키마는 임의의 정규식으로 표현될 수가 없고 이것은 정규식으로 표현되는 DTD를 표현하는 것이 불가능하다고 보기 때문에 DTD를 추출하기 위하여 [13, 14]에서 제안한 MDL(Minimum Description Length) 기법을 이용하였다.

그러나 이 논문에서 제안한 방법은 엘리먼트의 순서와 속성이 추가된 개선된 OEM 데이터 모델을 사용하여 XML Schema를 추출하기 때문에 데이터 중심의 XML 문서와 문서 중심의 XML 문서 모두를 만족하여 처리할 수가 있다. 또한 [7, 12]에서 고려하지 않은 데이터 형식의 변환과 재사용성을 추가하여 XML 문서로부터 재사용 가능한 XML Schema의 추출이 가능하다.

3. 시스템 구성 및 최대/최소 경계 스키마 그래프 추출

3.1 XML Schema 추출 시스템



(그림 4) XML Schema 추출 시스템 구성도

문서 검증기(parser)는 스키마가 없는 XML 문서를 입력으로 받는다. 우선 문서 검증기를 이용하여 XML 문서가 문서 형식에 적합한지를 검사한다. 다음 단계에서 스키마 추출기는 XML 문서를 개선된 OEM 데이터 모델로 변환한다. 여기에서 데이터가이드와 시뮬레이션을 이용하여 최소 경계 스키마 그래프를 추출하며, 추출된 최대/최소 경계 스키마 그래프를 기반으로 절대/상대 스키마 그래프를 생성한다. 그리고 이 그래프와 중복 레이블 및 순서, 속성 정보로 스키마 그래프를 추출한다. 마지막으로 스키마 그래프와 개선된 OEM 데이터 모델에서 추출한 데이터 형식, 순서, 속성 정보를 가지고 재사용성을 고려하여 Venetian Blind Model을 이용한 정형화된 패턴 리스트로 XML Schema를 생성한다. 이때에 XML 문서에 있는 텍스트 데이터를 XML Schema의 데이터 형식으로 변환하기 위하여 데이터 형식 변환을 처리한다. (그림 4)는 이와 같은 XML Schema 추출 시스템의 구성을 나타낸 것이다.

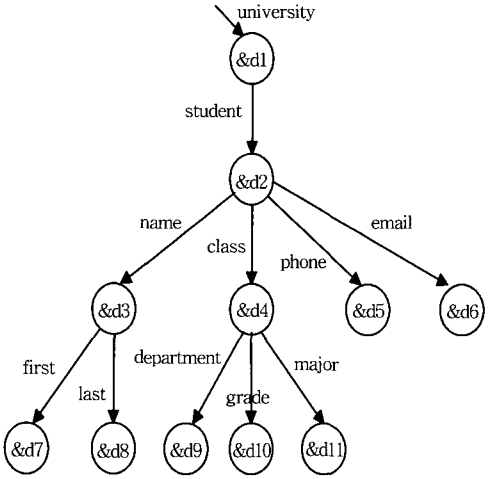
3.2 최대/최소 경계 스키마 그래프 추출

3.2.1 최대 경계 스키마 그래프 추출

(그림 2)의 개선된 OEM 데이터 모델에 대하여 데이터가이드 알고리즘을 적용하면 XML 데이터의 모든 유일한 레이블 경로를 XML 데이터에 나타나는 빈도에 상관없이 한 번만 표시되는 최대 경계 스키마 그래프를 만들 수가 있다. root 노드 {&1}과 &d1로 초기화되어 있는 데이터가이드에서 시작하여 &1의 모든 자식 노드에 대해서 <label, oid> 순서쌍을 원소로 하는 집합 P를 생성하고 P집합의 원소들에 대해서 label이 같은 oid들의 집합들 즉, <label,{oid, oid·}>가 원소가 되는 집합 T를 생성한다. 집합 T의 모든 원소들은 새롭게 데이터가이드에 삽입될 노드(예 : &d1, &d2, &d3 …)를 생성하고 이것을 데이터가이드에 삽입한다. 이때 삽입 노드를 생성하기 전에 T의 원소인 <label,{oid, oid, …}> 순서쌍에서 {oid, oid, …}가 targetHash에 존재하지 않는다면 targetHash에 삽입하고 노드를 생성하여 현재 노드에 연결한다. 그러나 이미 존재한다면 새로운 노드를 생성하지 않고 targetHash에 존재하는 노드 값에 해당하는 데이터가이드의 노드를 현재 노드에 연결한다.

(그림 5)는 (그림 2)에서 보여준 개선된 OEM 데이터 모델을 데이터가이드를 이용하여 추출한 최대 경계 스키마 그래프이다. 데이터가이드의 사용에서 길이가 n인 주어진 레이블 경로가 데이터 소스에 있는지를 검사 할 수가 있다. 예를 들어, (그림 5)에서 경로 university.student.phone이 있는지를 증명하기 위해서는 &d1번과 &d2번 객체의 출력간선을 검사하면 된다. 마찬가지로 데이터가이드에 있는 레이블 경로 l의 단일 인스턴스를 따라서 어떤 객체 o에 도달하면, o의 출력간선 위에 있는 레이블은 데이터 소스에 있는 l을 따라오는 모든 가능한 레이블을 보여준다. (그림 5)에서 객체 &d4번의 서로 다른 레이블인 세 개의 출력간선은 데이터

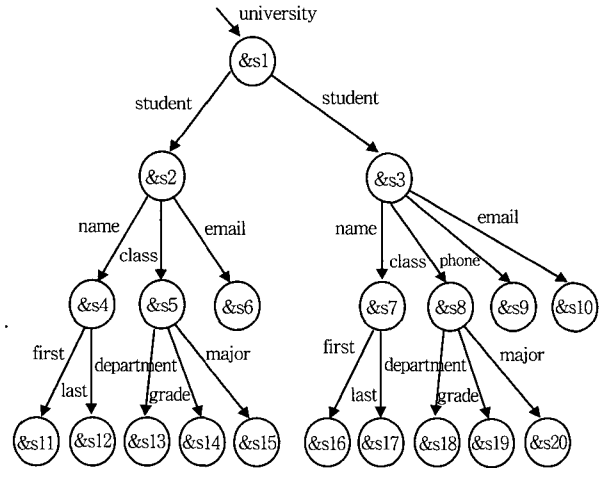
소스에서 class 다음에 올 수 있는 모든 가능한 레이블을 표시한다. (그림 5)에서 데이터가이드는 원자 값을 가지지 않기 때문에 데이터베이스의 구조를 보여주는데 적합하다.



(그림 5) 최대 경계 스키마 그래프

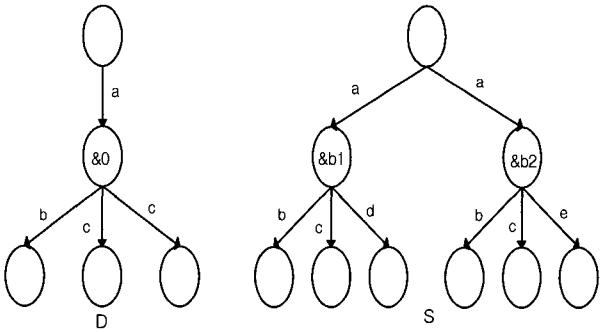
3.2.2 최소 경계 스키마 그래프 추출 및 모호성 제거

(그림 3)에서 직선인 간선들이 존재할 때 점선인 간선들이 존재하고 이에 대응하는 노드 y2가 존재하면 이진 릴레이션 R은 시뮬레이션이 된다. 즉, 점선인 간선 x2[l]y2는 직선인 x1[l]y1을 시뮬레이트한다고 한다. 이와 같은 이론을 기반으로 그래프 시뮬레이션을 이용하여 최소 경계 스키마 그래프를 추출하는 자세한 내용과 알고리즘은 [3, 12]를 참고하면 된다. (그림 2)에 대한 최소 경계 스키마 그래프를 추출한 결과는 (그림 6)과 같다.



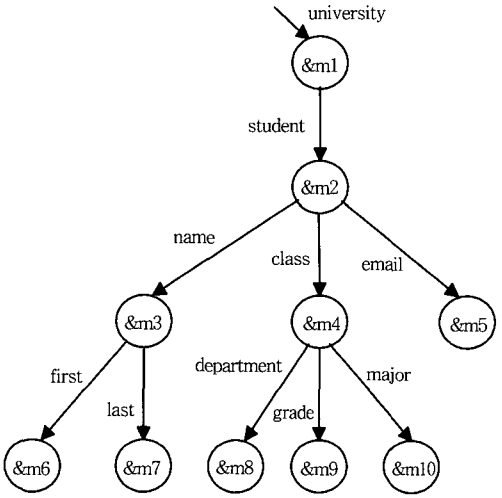
(그림 6) 최소 경계 스키마 그래프

최대 경계 스키마 그래프의 경우에 주어진 데이터 그래프 D에 대하여 타입을 구분할 때 모호성이 발생하지 않는 반면에 최소 경계 스키마 그래프에서는 모호성이 발생하게 된다. 일반적으로 어떤 데이터 그래프 D와 스키마 그래프 S 사이에는 시뮬레이션 R을 하나 이상 가진다. (그림 7)에서 객체 &o는 &b1, &b2 또는 &b1과 &b2 두 노드 모두로 분류될 수 있다. 이것은 주어진 데이터 그래프가 스키마 그래프에 대해 세 가지 서로 다른 시뮬레이션을 가지게 된다는 뜻이다. 예를 들어 (그림 6)에서 &s1을 기준으로 student를 통해 도달할 수 있는 노드는 &s2, &s3으로 간선상의 레이블만을 가지고는 타입을 구성하는데 있어서 모호성이 발생하게 된다.



(그림 7) 분류에 따른 모호성

따라서 같은 레이블을 가진 간선들을 통합하여 이러한 모호성을 제거해 주어야 한다. (그림 6)의 이러한 모호성을 제거한 최소 경계 스키마 그래프를 (그림 8)에서 보여주고 있다.



(그림 8) 모호성을 제거한 최소 경계 스키마 그래프

4. 재사용 가능한 XML Schema 추출

XML 문서에서 반구조적 데이터의 스키마 추출 기법을 이용하여 데이더가이드로 최대 경계 스키마 그래프를 생성하고, 시뮬레이션으로 최소 경계 스키마 그래프를 생성하였다. 그리고 연결자와 발생 지시자를 구분하기 위하여 주어진 데이터 그래프인 개선된 OEM 데이터 모델로부터 결정적, 비결정적 레이블 정보를 얻는다. 이러한 레이블 정보는 최대/최소 경계 스키마 그래프의 노드와 레이블을 비교하여 중복되는 레이블인 절대 스키마 그래프와 중복되지 않

은 레이블인 상대 스키마 그래프를 얻는다. 이 그래프들은 스키마 그래프 생성에 필요한 발생 지시자인 '?', '*', '+'를 추출하고, '|' 연결자를 추출하는데 사용된다. 추출한 발생 지시자와 연결자의 레이블 정보와 절대 스키마 그래프와 상대 스키마 그래프를 가지고 엘리먼트의 반복횟수와 '|' 연결자를 알 수 있는 스키마 그래프를 생성한다. 그리고 스키마 그래프를 기반으로 개선된 OEM 데이터 모델로부터 데이터 형식, 순서, 속성 정보를 가지고 재사용성을 고려한 패턴 형식으로 XML Schema를 생성한다. 이러한 개선된 OEM 데이터 모델의 데이터 구조는 다음의 <표 2>와 같다.

〈표 2〉 개선된 OEM 데이터 모델의 데이터 구조

칼럼 이름	내 용
oid	엘리먼트 고유 ID
ename	엘리먼트 이름
ccontent	원자 객체 여부(True/False)
avalue	원자 객체의 데이터 값
cedge	자식 엘리먼트 리스트
attribute	엘리먼트의 속성 리스트

개선된 OEM 데이터 모델의 데이터 구조에서 oid는 그래프의 객체 식별자에 해당하며, ename은 상위 객체와의 관계를 표현한 레이블에 해당한다. ccontent는 엘리먼트가 원자객체인지를, avalue는 엘리먼트가 원자 객체일 때의 원자값에 해당한다. cedge는 자식 객체에 대한 객체 식별자 리스트를, attribute는 엘리먼트의 속성 정보 리스트를 가진다.

4.1 발생 지시자와 연결자 레이블 정보 추출

한 엘리먼트의 자식 엘리먼트의 정의를 통하여 각 엘리먼트를 결정적인 엘리먼트와 비결정적인 엘리먼트로 구분할 수가 있다. 발생 지시자에는 '?', '*', '+'가 있다. 발생 지시자 '?', '*'는 XML 문서에서 관련된 엘리먼트가 나타나지 않을 수도 있음을 의미하며, '+'는 반드시 한번은 나타남을 의미한다. 전자와 같이 정의된 엘리먼트들을 비결정적이라고 하고, 후자의 경우를 결정적이라고 한다. 또한 자식 엘리먼트들에 대한 연결자로는 ','(sequence connector)와 '|'(choice connector)가 있다. ','는 ','로 연결된 엘리먼트들이 XML 문서내에서 순서대로 나타남을 의미한다. 그러나 '|'로 나열된 엘리먼트들은 이들 중에 하나만 나타남을 의미할 뿐이고, 실제로 XML 문서에서 어떤 엘리먼트가 나타날지는 미리 알 수가 없다. 그러므로 ','로 나열된 엘리먼트들은 결정적이지만 '|'로 나열된 엘리먼트들은 비결정적이다. 따라서 '?', '*', '|'에 관련된 엘리먼트를 비결정적 엘리먼트라하고, 이외의 모든 엘리먼트들을 결정적 엘리먼트라고 한다.

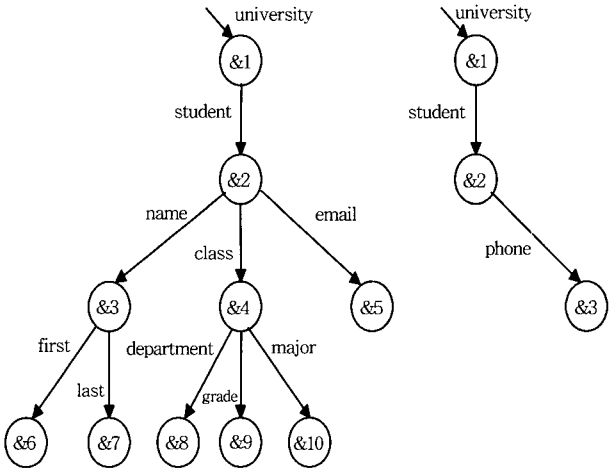
최대 경계 스키마 그래프와 최소 경계 스키마 그래프를 얻게되면 우선 주어진 개선된 OEM 데이터 모델로부터 연결자와 발생 지시자 레이블 정보를 얻어내야 한다. 발생 지시자 정보인 중복이 되는 레이블의 정보는 스키마 그래프를 생성하는데 유용한 정보로 이용된다. 예를 들어 (그림 2)에서 &2인 노드는 레이블이 email인 출력 간선이 2개 존재하고, &3인 노드는 레이블이 phone인 출력 간선이 2개 존재함을 알 수 있다. 이러한 중복되는 레이블 정보를 저장한 테이블은 <표 3>과 같다. 이것은 문서상에 해당 엘리먼트가 여러 번 중복되어 표현됨을 의미하게 때문에 해당 엘리먼트는 XML Schema 상에서 '*' 혹은 '+'인 minCccurs와 maxOccurs로 표현된다. 또한 반구조적 데이터 그래프에서 &2, &3, &4 노드인 엘리먼트의 순서는 왼쪽에서 오른쪽으로 매겨진다. 이러한 순서는 XML Schema 상에서 ',' 연결자인 <sequence>로 표현된다. '|' 연결자인 <choice>는 절대 스키마 그래프와 상대 스키마 그래프에서 같은 이름의 부모 노드에 포함된 서로 다른 레이블이름을 가지는 자식 노드가 단일하게 존재하는 경우이다.

〈표 3〉 레이블 중복 테이블

노 드	출력간선 레이블	출력간선 수
&1	student	3
&2	email	2
&3	phone	2

4.2 절대 스키마 그래프와 상대 스키마 그래프

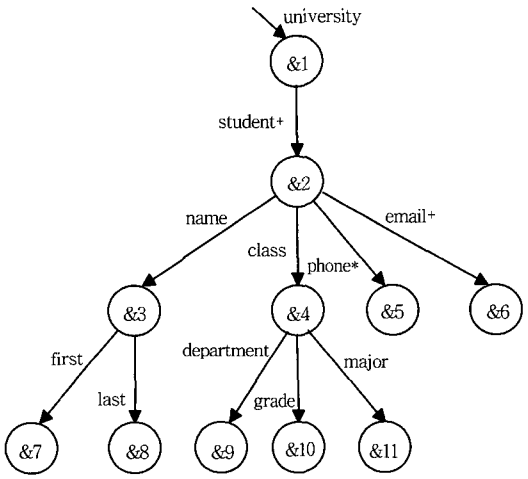
XML 문서상에 반드시 있어야 되는 결정적 엘리먼트인 중복되는 레이블들만으로 구성된 것을 절대 스키마라 칭하며, 중복되는 레이블들을 제외한 비결정적인 엘리먼트인 나머지 레이블들로 구성된 것을 상대 스키마라 지칭한다. 추출한 최대 경계 스키마 그래프와 최소 경계 스키마 그래프를 비교하여 중복되는 레이블로 구성된 절대 스키마 그래프와 중복되지 않은 레이블로 구성된 상대 스키마 그래프를 얻는다. 여기에서 절대 스키마 그래프는 반드시 문서상에 나타나야하는 부분을 나타내며, 상대 스키마 그래프는 문서상에 나타날 수도 나타나지 않을 수도 있는 부분을 나타낸다. (그림 9)은 절대 스키마 그래프와 상대 스키마 그래프를 보여주고 있다.



(그림 9) 절대 스키마 그래프와 상대 스키마 그래프

따라서 (그림 9)의 상대 스키마 그래프에서 레이블이 phone인 간선은 입력 간선의 레이블이 student인 노드로부터 여러 개의 출력 간선으로 표현될 수 있고 또는 표현되지 않을 수도 있기 때문에 스키마 그래프에 phone*로 표시되어야 한다. 또한 절대 스키마 그래프에서 레이블이 email인 경우에는 반드시 문서 내에 존재해야하고 여러 번 중복되어 존재할 수도 있기 때문에(한번 이상은 문서에 존재하기 때문에) 스키마 그래프에 email+로 표시되어야 한다. <표 3>의 레이블 중복 테이블과 (그림 9)의 절대 스키마 그래프와 상대 스키마 그래프로서 스키마 그래프를 구할 수 있다.

(그림 10)은 XML Schema의 발생 지시자에 필요한 정보를 포함한 스키마 그래프를 나타낸 것이다. 이러한 스키마 그래프에서 레이블 뒤에 붙는 발생 지시자(?, *, +)는 엘리먼트 정의에 필요한 minOccurs와 maxOccurs에 이용되고, 연결자 '|'는 엘리먼트의 선택인 <choice>에 이용된다.



(그림 10) (그림 2)에 대한 스키마 그래프

4.3 데이터 형식의 변환

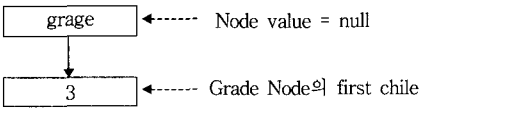
데이터 형식은 데이터의 크기와 형식을 나타낸다. XML Schema에는 45개의 정의된 형식을 가지고 있지만, DTD에는 string이라는 한가지 데이터 형식만을 지원한다. 이러한 차이점으로 인하여 데이터 형식의 변환은 XML 문서에서 XML Schema를 추출할 때에 가장 중요한 요소가 된다.

〈표 4〉 제한된 단순 형식

형 식	종 류
원시 형식	boolean, decimal, float, double, string, time, datetime, date, anyURI, NOTATION
유도된 형 식	int, long, short, byte, NMTOKEN, NMTOKENS, Name, ID, IDREF, IDREFS, ENTITY, ENTITIES, integer

그러나, XML 문서에 있는 태그는 문서 구조에 대한 규칙이 정의되어있지 않기 때문에 태그만으로는 그 값의 데이터 형식을 식별하기란 불가능하다. 이에 이 논문에서는 VBScript에서와 같이 태그에 저장되어 있는 데이터 값의 형식에 따라 적절한 형식의 서브 형식으로 데이터가 변환되는 방식을 제안한다. 또한 XML Schema에서 제공되는 모든 데이터 형식을 변환하기란 불가능하기 때문에 <표 4>에서와 같이 제한된 단순 형식만으로 변환한다.

태그에서의 값의 추출은 DOM을 이용하여 처리한다. 즉, 엘리먼트의 first child가 text 노드인지를 확인하고, text 노드이면 text 노드의 node value를 출력하여 제한된 단순 형식과 변환하여 처리한다. <grade>3</grade> 엘리먼트에서 text node의 추출과정은 (그림 11)과 같다.



	grade node의 값	node value의 값
grade element node	grade	null
3 text node	# text	3

(그림 11) text node의 추출 과정

4.4 Venetian Blind Model을 적용한 패턴처리 방법

모듈의 재사용성을 높게 설계하는 전략은 모듈의 독립성인 응집도를 강하게 하고 결합도를 약하게 하는 것이다. 이 논문에서는 설계 방법인 Venetian Blind Model을 사용하여 엘리먼트와 데이터 형식을 적절히 지역화 시키고 최대의 재사용성을 얻는다.

Venetian Blind Model은 엘리먼트 선언 방식에서 부모 엘리먼트만을 전역으로 선언하고, 자식 엘리먼트들은 지역으로 선언한다. 이렇게 선언함으로 해서 지역 엘리먼트에 대하여 파서가 간접적으로 참조하며, 네임스페이스를 사용할 경우에 네임스페이스를 밝힐 필요가 없기 때문에 재사용성이 높아진다. 또한 데이터 형식의 선언 방식에서는 정규 데이터 형식, <simpleType>, <complexType>의 3가지 데이터 형식을 적절히 사용하면 모듈화 할 수가 있는데, 이러한 모듈화는 재사용 가능하고 유연성이 좋은 스키마를 만들 수가 있다.

<simpleType>을 이용하여 데이터 형식을 정의하는 방식에는 엘리먼트 내부에서 정의하여 사용하는 방식과 외부에 선언하여 참조하는 방식이 있다. <simpleType>을 외부에 선언하는 방식은 독립적으로 선언된 것이기 때문에 다른 외부 엘리먼트에서 참조하여 활용할 수 있으며 높은 재사용성을 가진다. <complexType>은 <sequence>와 <attribute>를 사용하여 정의하는 방식이 다른 곳에서도 이름을 가지고 활용할 수 있기 때문에 재사용 측면에서 권장할 방식이다. 예를 들어, 엘리먼트 name에 대한 자식 엘리먼트들과 속성값을 지정하는 경우 다음과 같이 <complexType>을 정의할 수가 있다.

```
<element name = "name" type = "nameType" minOccurs = "1"
         maxOccurs = "3"/>
<complexType name = "nameType">
  <sequence>
    <element name = "first" type = "string"/>
```

```
    <element name = "last" type = "string"/>
  </sequence>
</complexType>
```

4.5 XML Schema 추출

스키마 그래프인 (그림 10)에서 XML Schema에서 엘리먼트의 반복횟수를 나타내는 발생 지시자와 선택인 <choice>를 나타내는 연결자 '|'에 필요한 정보를 얻게되면, 개선된 OEM 데이터 모델에서 추출한 데이터 형식, 순서, 속성을 가지고 재사용성을 높인 Venetian Blind Model을 적용한 패턴처리 방법으로 XML Schema를 생성한다. XML Schema의 엘리먼트 반복횟수 처리는 <표 5>의 발생 지시자의 패턴처리에 따라 생성한다. 엘리먼트간의 연결자는 <표 6>의 연결자의 패턴처리에 따라 데이터 형식의 선언 방법을 고려하여 생성한다. 또한 엘리먼트의 속성은 <표 2>의 OEM 데이터 모델의 데이터 구조의 속성 정보 리스트와 <표 7>의 속성 디폴트값 설정의 패턴처리에 따라 생성한다. 이때에 데이터 형식의 변환을 위하여 문서 검증기에서 추출한 엘리먼트의 text node와 <표 4>의 제한된 단순 형식 리스트를 비교하여 처리한다.

〈표 5〉 발생 지시자의 패턴처리

기 호	생 성 조 건	XML Schema 생성 패턴
?	같은 엘리먼트명을 가지는 자식노드가 같은 이름의 다른 부모노드에서는 존재하지 않을 경우	<element name = "root" type = "rootType" minOccurs = "0"/> <complexType name = "rootType"> <element name = "count" type = "integer"/> </complexType>
*	같은 엘리먼트명을 가지는 자식노드가 한번 이상 존재하고, 같은 이름의 다른 부모노드에서는 존재하지 않을 경우	<element name = "root" type = "rootType" minOccurs = "0" maxOccurs = "unbounded"/> <complexType cname = "rootType"> <element name = "count" type = "integer"/> </complexType>
+	같은 이름의 부모노드에 포함된 같은 엘리먼트명을 가지는 자식노드가 한번 이상 존재할 경우	<element name = "root" type = "rootType" minOccurs = "1" maxOccurs = "unbounded"/> <complexType name = "rootType"> <element name = "count" type = "integer"/> </complexType>

〈표 6〉 연결자의 패턴처리

기 호	생 성 조 건	XML Schema 생성 패턴
,	같은 부모노드에 포함된 서로 다른 엘리먼트명을 가지는 자식노드가 순서를 고려하여 단일하게 존재할 경우	<element name = "root" type = "rootType"/> <complexType name = "rootType"> <sequence> <element name = "shipTo" type = "string"/> <element name = "billTo' type = "string"/> </sequence> </complexType>
|	같은 이름의 부모노드에 포함된 서로 다른 엘리먼트명을 가지는 자식노드가 단일하게 존재할 경우	<element name = "root" type = "rootType"/> <complexType name = "rootType"> <choice> <element name = "shipTo" type = "string"/> <element name = "billTo' type = "string"/> </choice> </complexType>

예를 들어, (그림 10)의 스키마 그래프에서 레이블 뒤에 붙는 발생 지시자(?, *, +)는 XML Schema 엘리먼트 정의시에 필요한 minOccurs와 maxOccurs를 반영한 패턴처리에 따라 엘리먼트를 정의한다. student+는 <표 5>의 발생 지시자 패턴처리 방식의 '+' 발생 지시자와 자식 엘리먼트가 순서적으로 나열됨에 따라 <표 6>의 연결자의 패턴처리 방식의 ',' 연결자에 따라 다음과 같이 student 엘리먼트를 정의할 수 있다. 또한 id 속성은 데이터 형식 변환에 의하여 integer 형식으로 처리된다.

```
<element name = "student" type = "studentType"
        minOccurs = "1" maxOccurs = "unbounded"/>
<complexType name = "studentType">
  <sequence>
    <element ref = "name"/>
    <element ref = "class"/>
    <element name = "email" type = "string"
             minOccurs = "1" maxOccurs = "unbounded"/>
    <element name = "phone" type = "string"
             minOccurs = "0" maxOccurs = "unbounded"/>
  </sequence>
  <attribute name = "advisor" type = "string" use = "optional"/>
  <attribute name = "id" type = "integer" use = "required"/>
  <attribute name = "Sadvisor" type = "string" use = "optional"/>
</complexType>
```

이러한 방식으로 (그림 1) 샘플 XML 문서의 XML Schema

〈표 7〉 속성 디폴트값 설정의 패턴처리

디폴트값	생 성 조 건	XML Schema 생성 패턴
#IMPLIED	엘리먼트에 한 개 이상의 속성이 있고, 각 속성에 두 개 이상의 속성값이 있는 경우	<attribute name = "advisor" type = "string" use = "optional"/>
#REQUIRED	엘리먼트에 한 개 이상의 속성이 있고, 각 속성에 하나의 속성값만이 있는 경우	<attribute name = "id" type = "string" use = "required"/>
#FIXED	엘리먼트에 하나의 속성과 속성값만을 가지는 경우	<attribute name = "class" type = "string" use = "fixed" value = "A"/>

의 추출 결과는 (그림 12)와 같다. 그리고 grade 엘리먼트는 데이터 형식 변환에 의하여 integer 형식으로 변환된다.

```
<schema>
<element name = "university" type = "universityType"/>
<complexType name = "universityType">
  <element ref = "student" />
</complexType>
<element name = "student" type = "studentType"
         minOccurs = "1" maxOccurs = "unbounded"/>
<complexType name = "studentType">
  <sequence>
    <element ref = "name"/>
    <element ref = "class"/>
    <element name = "email" type = "string" minOccurs = "1"
             maxOccurs = "unbounded"/>
    <element name = "phone" type = "string" minOccurs = "0"
             maxOccurs = "unbounded"/>
  </sequence>
  <attribute name = "advisor" type = "string" use="optional"/>
  <attribute name = "id" type = "integer" use = "required"/>
  <attribute name = "Sadvisor" type = "string"
           use = "optional"/>
</complexType>
<element name = "name" type = "nameType" minOccurs = "1"
         maxOccurs = "1"/>
<complexType name = "nameType">
  <sequence>
    <element name = "first" type = "string"/>
    <element name = "last" type = "string"/>
  </sequence>
</complexType>
<element name = "class" type = "classType" minOccurs = "1"
         maxOccurs = "1"/>
<complexType name = "classType">
  <sequence>
    <element name = "department" type = "string"/>
    <element name = "grade" type = "integer"/>
    <element name = "major" type = "string"/>
  </sequence>
</complexType>
</schema>
```

(그림 12) (그림 2)의 XML Schema

6. 평 가

이 논문은 well-formed XML 문서이거나 XML Schema가 없는 XML 문서에서 문서의 구조적 정보를 정의하는 XML Schema를 추출하는 기법에 관한 연구이다. XML 문서에서 DTD를 생성하는 상용화된 제품들은 많이 있다. 그러나 이 논문의 XML Schema 추출에 대한 연구는 국내외에서 개발되어 상용화된 시스템은 아직 없으며, 단지 연구를 진행하여 프로토타입의 시스템을 보여주고 있다[17, 18]. 스키마가 없는 XML 문서에서 XML Schema 추출을 위해 제안한 추출기법의 비교 평가를 위해 기존의 [7, 17]을 비교대상으로 하였다. 비교대상인 [17]은 XML Schema를 웹 상에서 추출하는 시스템이다. 그러나 [7]은 DTD를 추출하는 시스템으로 XML Schema를 추출하지는 않지만 이 논문과 같은 데이터가이드와 시뮬레이션을 사용한 추출 기법이기 때문에 비교대상으로 삼았다.

실험 평가를 위하여 MDS Schema[15]의 XML 문서인 descriptionExample002를 각각의 추출기를 통해 XML Schema를 추출하였다. 구현한 추출기를 기존의 추출기와 비교하여 시스템의 성능 분석을 수행하였다. <표 8>과 같이 기존의 추출기와 비교해 볼 때 이 논문의 추출기는 엘리먼트와 데이터 형식의 전역과 지역을 적절히 설계하여 응집도를 중간으로 하고 결합도를 약하게 하여 재사용성을 높였다.

이 논문에서 제안한 추출 기법의 장점은 첫째, 객체지향적 요소인 재사용성을 고려하여 XML Schema를 추출하였다. 둘째, 새로운 OEM 데이터 모델로서 엘리먼트의 순서와 속성을 반영하였다. 셋째, 데이터 중심과 문서 중심의 XML 문서 처리가 가능하도록 하였다. 미흡한 점으로는 네임스페이스를 고려하지 않았으며, 부모와 자식 엘리먼트의 구분을 고려한 문서 중심의 XML 문서에 적합한 처리를 하지 못하였다.

7. 결 론

이 논문에서는 스키마가 없는 XML 문서에서 XML Schema를 추출하기 위하여 개선된 OEM 데이터 모델로서 스키마 그래프를 추출하고, 데이터 형식의 변환, 재사용을 고려하여 Venetian Blind Model을 적용한 패턴처리를 이용한 추출기법을 제안하였다. 이 논문에서는 엘리먼트의 순서와 속성이 추가된 개선된 OEM 데이터 모델을 사용하여 XML Schema를 추출하기 때문에 데이터 중심의 XML 문서와 문서 중심의 XML 문서 모두를 만족하여 처리할 수가 있다. 또한 [7, 17]에서 고려하지 않은 재사용성과 데이터

〈표 8〉 이 논문의 추출 기법과 기존 기법의 비교

평가항목	이 논문의 추출 기법	XSD Schema Generator[17]	DTD 추출 기법[7]
응집도	중 간	강 함	약 함
결합도	약 함	강 함	강 함
재사용성	높 음	낮 음	낮 음
특징 및 장점	재사용성 반영 데이터 형식 변환 엘리먼트의 순서, 선택, 속성 처리 반영 데이터중심과 문서중심의 XML 문서 처리	엘리먼트의 순서, 속성 처리 반영 웹 상에서 실행	재사용성 개념 없음 데이터중심의 XML 문서에 적합함 처리가 간단함
단 점	네임스페이스 고려 안함 부모와 자식 엘리먼트 처리 및 구분 없음	네임스페이스 고려 안함 부모와 자식 엘리먼트 처리 및 구분 없음 재사용성 반영 못함 데이터 형식 변환 못함	문서중심의 XML 문서 처리 못함 부모와 자식 엘리먼트 처리 및 구분 없음

형식 변환을 추가하였고 보다 효율적으로 XML 문서로부터 XML Schema의 추출하였다.

XML 문서로부터 반구조적 스키마 추출기법을 이용한 스키마 트리의 추출은 상당한 장점이 있다. XML 데이터를 관계형 데이터베이스에 저장할 경우에 스키마 트리를 이용하여 효율적인 관계형 테이블 생성을 가능하게 하고, 또한 데이터베이스로부터 데이터를 XML 형태로 출력할 경우에 스키마 트리를 이용하여 XML 문서의 생성을 용이하게 해준다. 그리고 사용자가 질의를 할 경우에도 사용자에게 편리성을 제공해 준다.

이 논문에서 제시한 XML Schema 추출 기법에는 다음과 같은 제한 사항을 두었다. 네임스페이스와 속성선언시의 지정타입인 엔티티(ENTITY), NMTOKEN, NOTATION, 열거형(enumeration) 타입을 고려하지 않았다. 또한 태그 엘리먼트 사이의 공백과 코멘트는 제외하였다. 그리고 XML Schema 생성시 내용모델 그룹이 셋 이상 중첩될 경우에는 처리할 수가 없으며, 비교적 단순한 OR('|'연결자) 구조를 처리대상으로 하였다.

향후 연구로는 제안한 방법을 기반으로 한 XML Schema의 추출시 이 논문에서 고려하지 않은 네임스페이스와 문서 중심의 XML 문서에서 중요시되는 속성의 엔티티, NMTOKEN, 열거형 타입과 부모와 자식 엘리먼트의 구분 등을 고려해 데이터 중심과 문서 중심의 XML 문서에 적합한 XML Schema에 대한 연구가 필요하다.

참 고 문 헌

[1] Christof Bornhovd, "Semantic Metadata for the Integration of Web-based Data for Electronic Commerce," IEEE, Nov., 1999.

[2] 조정길, 조윤기, 구연설, "구조적 상이성 분석에 기반한 XML 문서 변환 시스템의 설계 및 구현", 정보처리학회논문지D, 제9-D권 제2호., pp.297-306, 2002.

[3] S. Abiteboul, P. Bunneman, D. Suciu, "Data on the Web : From Relations to Semistructured Data and XML," Morgan Kaufmann, 1999.

[4] P. Buneman, S. Davidson, G. Hillebrand and D. Suciu, "A Query language and optimization techniques for unstructured data," In SIGMOD, Montreal, 1996.

[5] R. Goldman, J. Widom, "DataGuide : Enabling Query Formulation and Optimization In Semistructured Databases,". In Proceedings of the Conference on VLDB, 1998.

[6] S. Nestorov, S. Abiteboul, R. Motwani, "Extracting Schema from Semistructured Data," In SIGMOD, pp.295-306, 1998.

[7] 박경현, 최은선, 이종연, 박정석, 류근호, "최대/최소 경계 스키마 추출 기법을 이용한 XML 문서의 DTD 추출", 컴퓨터정보통신연구논문지, 2000.

[8] H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman and J. Widom, "Integration and Accessing Heterogeneous Information Sources in TSIMMIS," In Proceedings of the AAAI Symposium on Information Gathering, pp.61-64, 1995.

[9] J. McHugh, S. Abiteboul, R. Goldman, D. Quass and J. Widom, "Lore : A Database Management System for Semistructured Data," SIGMOD Record, 26(3), September, 1997.

[10] Roy Goldman, Jason McHugh, Jennifer Widom, "From Semistructured Data to XML : Migrating the Lore Data Model and Query Language," WebDB (Informal Proceedings), 1999.

[11] 박경현, 이경휴, 류근호, "DTD가 없는 XML 데이터의 효율적인 저장 기법", 정보처리학회논문지D, 제8-D권 제5호, pp. 495-506, 2001.

[12] M. Garofalakis, A. Gionis, R. Rastogi, S. Seshadri, K.Shim, "XTRACT : A System for Extracting Document Type Descriptors from XML Documents," In Proc. of the ACM SIGMOD international Conf. on Management of Data, Dallas, Texas, 2000.

[13] A. Brazma, "Efficient identification of regular expressions from representative examples," In Proc. of the Ann. Conf. on Computational Learing Theroy(COLT), 1993.

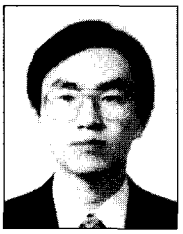
[14] P. Kilpelainen, H. Mannila, and E. Ukkonen, "MDL learning of unions od simple pattern languages from positive examples," In Proc. of the European Conf. on Computational Learing Theroy(Eurocolt), 1995.

[15] IBM, "MPEG-7 Schema Page," http://pmedia.i2.ibm.com : 8000/mpeg7/schema, April, 2002.

[16] Jon Duckett, et. al, "Professional XML Schema," Wrox, 2002.

[17] XML for ASP.NET Developers, "XSD Schema Generator," http://www.xmlforasp.net/codeSection.aspx?csID=16, May, 2001.

[18] RJT Netproductions, "Simple Sample DTD/XML Generator," http://rtiess.tripod.com/dtdxml.htm, Apr., 2002.



조 정 길

e-mail : innocom@dreamwiz.com

1987년 숭실대학교 전자계산학과(공학사)

1993년 숭실대학교 정보과학대학원(이학석사)

2003년 충북대학교 대학원 전자계산학과(이학박사)

1996년~현재 남서울대학교 컴퓨터학과 겸임교수

관심분야 : 객체지향 자료 모델링, XML 문서관리, 질의처리, 정보검색

구 연 설

e-mail : yskoo@cbucc.chungbuk.ac.kr

1964년 청주대학교 졸업(학사)

1975년 성균관대학교 경영행정대학원 전자자료처리학과(경영학석사)

1981년 동국대학교 대학원 통계학과(이학석사)

1988년 광운대학교 대학원 전자계산학과(이학박사)

1994년~1995년 한국정보과학회 부회장

1979년~현재 충북대학교 컴퓨터과학과 교수

관심분야 : 객체지향 테스팅, 품질관리, 정보 검색, 전자상거래