DTD가 없는 XML 데이터의 효율적인 저장 기법

박 경 현† · 이 경 휴†† · 류 근 호†††

요 약

XML이 인터넷상의 데이터 교환의 표준으로 대두되면서 데이터 모델이나 플랫폼에 관계없이 데이터의 전송이 가능하게 되었다. 특히 데이터 중심의 XML문서의 경우 전송시의 부하를 줄이기 위해 DTD없이 전송되는 경우가 일반적이다. 그러한 이유로 전송받은 XML 데이터를 효율적으로 저장하고 질의를 최적화하며 또한 관계형 데이터베이스에 저장된 기존의 데이터를 XML 형태로 출력하기 위해서는 DTD가 없는 XML 문서로부터 관계형 스키마의 추출이 필수적으로 요구된다. 따라서 이 논문에서는 반구조적 데이터의 스키마 추출 기법인 최대/최소 경계 스키마 추출 기법을 이용하여 DTD가 없는 XML 문서로부터 관계형 스키마를 생성하고 이를 바탕으로 XML 데이터를 저장하는 방법을 제시한다. 특히, 반구조적 데이터의 최소 경계 스키마를 추출하는데 있어서 기존의 데이터로그보다 효율적인 방법인 시뮬레이션을 제안함으로써 관계형 스키마를 생성하는데 있어서 보다 향상된 방법을 보여준다.

An Efficient Technique for Storing XML Data without DTD

Kyoung-Hyun Park† · Kyung-Hyu Lee†† · Keun-Ho Ryu†††

ABSTRACT

XML makes it possible for data to be exchanged regardless of the data model in which it is represented or the platform on which it is stored, serving as a standard for data exchange format on the internet. Especially, it is natural to send XML data without DTD on the internet when XML is data-centric. Therefore it is needed to extract relational schema to store XML data efficiently, to optimize queries, and to publish data which have been stored in the relational database in the XML format. In this paper, we propose a method to generate relational schema from XML documents without DTD and store XML data using upper/lower bound schema extraction technique for semistructured data. In extracting a lower bound schema, we especially show an efficient technique for creating relational schema by using simulation which is more advanced than the datalog method.

키워드 : XML, 반구조적 데이터(Semistructured Date), 시뮬레이션(Simulation), 데이터가이드(DataGuide), 데이터로그(DataLog)

1. 서 론

최근 인터넷이 발전함에 따라 다양한 형태의 정보들이 인터넷상에서 기하급수적으로 급증하면서 HTML로 이루어진 정보를 보다 효과적으로 사용하고자 하는 연구가 진행되었다. 그러나 HTML이 화면상에 어떠한 형태로 보여지는 기능 이외에는 별다른 기능을 제공하고 있지 않을 뿐만 아니라 고정된 태그만을 사용하고 있어 정보를 표현하는데 한계점을 드러내었다. 이에 따라서 W3C는 인터넷상에서 데이터 교환을 목적으로 HTML을 보완하기 위해 XML을 표준 마크업 언어로 채택하였다.

XML은 기존의 HTML이 갖는 문서 구조의 확장성과 문서의 검사 기능에 대한 문제점을 극복하고 SGML에서 처리가 난해했던 태그의 최소화, 생략등을 제거함으로써 문서를 처리하기 쉽고 명확하게 하였으며 문서 검색이나 재변환을 용이하게 하였다.

XML의 이러한 특징은 인터넷상에서 데이터 교환의 매개체 역할을 가능하게 하였고 이로써 인터넷상의 XML 데이터를 관계형 데이터베이스에 효율적으로 저장하고 기존의 관계형 데이터를 XML 형태로 출력하는 시스템에 대한 요구가 점점 증가하였다. 그에 따라 관계형 데이터베이스를 기반으로 하여 XML 데이터를 저장하고 질의하는 연구가 진행되어 왔는데 그 대표적인 시스템으로 XML-DBMS[1], SilkRoute[2], XPERANTO[3]등이 있다.

이러한 시스템들의 공통적인 특징은 DTD나 관계형 스키마중 적어도 하나는 존재하는 환경에서 XML 데이터를 저장하고 질의한다는 것이다. 즉 DTD를 바탕으로 관계형 스키마를 생성하여 XML 데이터를 저장하고 질의하며 관계형 스키마가 존재할 경우는 이를 바탕으로 DTD를 추론하여 XML 데이터를 처리하고 있다. 따라서 만약 XML 문서가 DTD를 가지고 있지 않고 관계형 데이터베이스도 관계형

† 정 회 원 : 한국전자통신연구원
†† 정 회 원 : 한국전자통신연구원 책임연구원
††† 종신회원 : 충북대학교 전기전자컴퓨터공학부 교수
논문접수 : 2001년 2월 26일, 심사완료 : 2001년 8월 7일

스키마가 존재하지 않는다고 가정한다면 XML 데이터의 저장은 불가능하게 된다.

이 논문에서는 이와 같이 DTD와 관계형 스키마가 모두 존재하지 않는 환경하에서 XML 데이터를 효율적으로 저장하기 위한 XML 데이터와 관계형 스키마간의 데이터 매핑 기법을 제시한다.

특히, XML 데이터의 공통구조를 추출해내기 위해 반구조적 데이터의 최대/최소 경계 스키마 추출 기법을 적용한다. 반구조적 데이터의 스키마 추출 기법은 그 동안 많은 연구[4-7]가 이루어져 왔는데 그중 대표적인 스키마 추출 방법으로 Lore 프로젝트의 데이터가이드(DataGuide)[6]와 데이터로그 규칙(DataLog Rule)[7]을 이용하는 방법이 있다.

이 논문에서도 XML 데이터의 최대 경계 스키마를 추출하기 위해 데이터가이드를 이용한다. 그러나 최소 경계 스키마를 추출하기 위해서 기존의 데이터로그 규칙을 이용하지 않고 새로운 추출 기법인 시뮬레이션을 이용한 추출 기법을 제안하고 적용한다. 그 이유는 데이터로그를 이용한 스키마 추출이 주어진 데이터 그래프의 노드를 방문할 때 모든 객체들이 분류된 초기 타입 릴레이션으로부터 그 릴레이션이 고정점에 도달할 때까지 반복하여 검사하기 때문에 $O(n^2)$이라는 시간을 요구하여 상대적으로 시스템에 많은 부담을 주기 때문이다.

시뮬레이션은 이전부터 데이터 그래프와 스키마 그래프사이의 관계에 대한 유효성을 검사하는데 사용되어져 왔다. 그러나 스키마 그래프가 생성되기 전에 주어진 데이터 그래프에 대한 스키마 그래프를 생성하기 위해 데이터 그래프내의 각 노드에 대해서 시뮬레이션의 관계에 있는 노드를 판단함으로써 데이터 그래프에 대한 스키마 그래프의 추출이 가능해진다.

2. 관련 연구

관계형 데이터베이스를 기반으로 XML 데이터를 저장하고 질의하는 시스템[1-3]들은 관계형 데이터와 XML 데이터간의 데이터를 매핑하는데 있어서 서로 다른 매핑 기법을 사용하고 있다.

XML-DBMS[1]는 XML 문서와 관계형 데이터베이스 상호간의 데이터를 전송하기 위한 시스템으로 기본적으로 XML 문서를 저장하기 위해서 입력되는 XML 문서를 객체 트리(object tree)로 표현되는 객체 뷰(object view)로 변환한 다음 객체-관계 매핑 기법을 이용하여 객체 뷰내의 객체들을 테이블로 매핑하는 방법을 사용한다. XML 문서를 객체 뷰로 매핑하고 객체 뷰를 관계형 테이블로 매핑하는 정보는 매핑 언어(XML-DBMS mapping language)를 사용하여 XML 문서 형태로 작성되어지기 때문에 데이터를 처리하는데 있어서 일관성을 유지할 수 있게 된다.

그러나 XML-DBMS가 상호간의 데이터 교환을 수행하기 위해서는 DTD나 관계형 스키마중 적어도 하나는 반드시 존재해야 한다. 즉 DTD만 존재할 경우 DTD를 사용하여 관계형 스키마를 생성한 후 XML 데이터를 저장하고 관계형 스키마만이 존재할 경우 관계형 스키마를 바탕으로 DTD를 생성하여 DTD에 맞게 관계형 데이터를 XML 형태로 출력한다. 따라서 관계형 스키마가 존재하지 않는 환경에서는 DTD가 없는 XML 데이터를 저장할 수 없다는 문제점이 발생한다.

XML-DBMS가 관계형 데이터와 XML 데이터간의 상호 전송을 위한 시스템이라면 SilkRoute[2]는 기존의 관계형 데이터를 XML 형태로 질의하고 질의 결과를 XML 문서로 생성하기 위한 시스템이다. XML 데이터와 관계형 데이터의 매핑에 있어서 XML-DBMS와 비교하여 가장 큰 차이점은 XML 데이터와 관계형 데이터 사이에 RXL(Relational to XML transformation Language)을 이용하여 질의를 위한 뷰를 작성한다는 것이다. 또한 관계형 데이터베이스와 XML 데이터와의 매핑 정보는 뷰 트리(view tree)안에 유지되는데 문서의 구조를 나타내는 템플릿과 데이터로그 규칙 형태로 나누어 매핑 정보를 나누어 저장한다. SilkRoute가 RXL을 이용하는 이유는 RXL이 SQL의 from절과 where절 그리고 XML-QL의 construct절을 합성하여 이루어져서 RXL내에 관계형 데이터의 스키마와 XML 데이터의 스키마 정보를 모두 포함할 수 있기 때문이다. 그러나 RXL 자체가 XML과 또다른 새로운 언어이기 때문에 Silk Route를 사용하기 위해서는 RXL을 다시 익혀야 하는 불편함이 발생한다.

이에 반해 XPERANTO[3]는 관계형 데이터뿐만 아니라 객체-관계형 데이터를 XML 형태로 질의하는 시스템으로 특히 관계형 데이터뿐만 아니라 메타 데이터도 질의가 가능한 시스템이다. 다른 시스템에 비해 XPERANTO의 가장 큰 특징은 XML-DBMS와 SilkRoute가 관계형 데이터와 XML 데이터의 매핑을 위해서 시스템에 종속적인 매핑 언어나 RXL을 사용한 반면 XPERANTO는 객체-관계형 데이터를 XML 형태의 뷰를 생성하기 위해 XML Schema를 사용한다는 점이다. 즉 질의를 위한 뷰를 XML Schema 형태로 제공하고 XQGM(Xml Query Graph Model)을 이용함으로써 객체-관계형 데이터를 XML 형태로 출력한다.

요약하면 위에서 언급한 시스템들은 모두 매핑 언어, RXL, XML Schema같은 중간적인 언어를 이용하여 XML 데이터와 관계형 스키마 사이의 매핑 정보를 유지하는 공통적인 특징을 가지고 있다

3. 최대경계 스키마

3.1 스키마 추출 모델

반구조적 데이터는 레이블과 방향성이 있는 그래프(la-

beled directed graph)로 표현될 수 있고 이 논문도 이러한 그래프 모델에 기반을 두고 있다. 그러나 XML 문서는 기본적으로 구조적 문서를 정의하는 모델로부터 시작되었기 때문에 두 모델사이에는 분명한 차이가 존재하게 된다. 따라서 XML 문서의 스키마를 추출하는데 반구조적 데이터의 스키마 추출 기법을 적용하기 위해서는 XML 데이터 모델을 레이블과 방향성이 있는 그래프 모델로 변경해 주어야 하는 작업이 필요하게 된다.

```
<est>
  <map>
    <identifiers>
      <dbestid> 6002866 </dbestid>
      <estname> 601572602T1 </estname>
      <genbackacc> BE738181 </genbackacc>
      <genbackgi> 10152173  </genbackgi>
    </identifiers>
    <cloneinfo>
      <cloneid> IMAGE : 3839544 <cloneid>
      <plate> LLCM528 </plate>
      <dnatype> cDNA </dnatype>
    </cloneinfo>
    <primers>
      <polyatail> Unknown </polyatail>
      <sequence> TGCTTTTTACC … </sequence>
      <quality> High quality sequence stops at base · 630 </quality>
      <entrycreated> Sep 14 2000 </entrycreated>
      <lastupdated> Sep 15 2000 </lastupdated>
      <comments> Tissue Procurement : ATCC </comments>
      …
    </primers>
    <library>
      <dbestlibid> 5949 </dbestlibid>
      <libname> NIH_MGC_57  </libname>
      <organism> Home sapiens </organism>
      <organ> brain </organ>
      <tissuetype> glioblastoma  </tissuetype>
      …
    </library>
    <submittrer>
      <name> Robert Strausberg. ph. D  </name>
      <tel> (301) 496-1550 </tel>
      <email> Robert_Strausberg@nih.gov </email>
    </submittrer>
    <citations>
      <title> National Institues of Health, Mammalian Gene
              Collection (MGC) </title>
      …
    </citation>
</map>
<map>
…
```

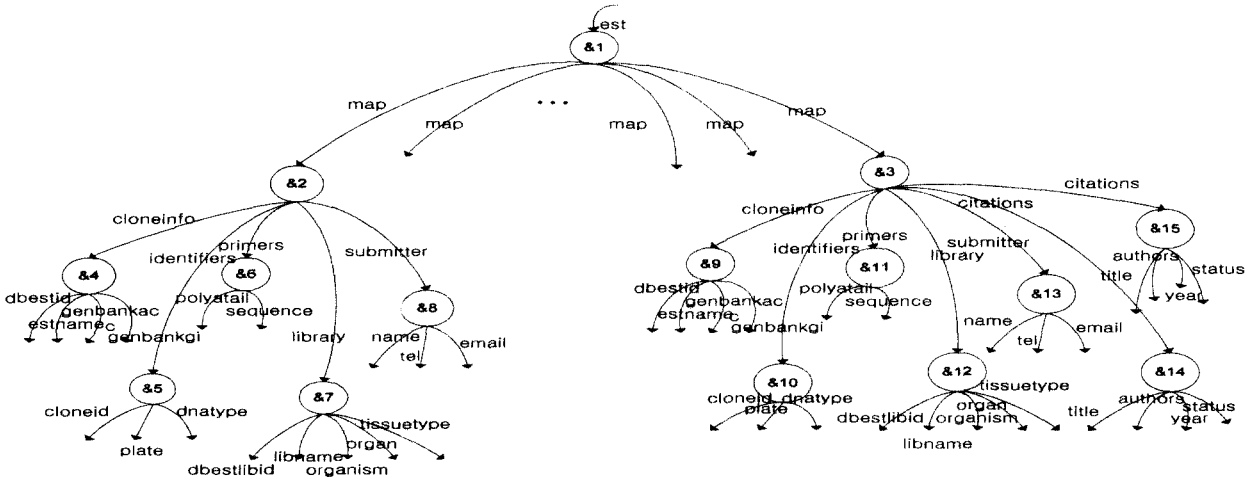
(그림 3-1) XML 문서

(그림 3-1)과 (그림 3-2)는 XML 문서와 이에 대응되는 반구조적 데이터의 데이터 모델을 보여주고 있다. XML 데이터 모델은 레이블이 노드상에 표현되고 반구조적 데이터에서는 간선상에 표현된다. 따라서 XML 데이터 모델을 반구조적 데이터 모델로 변형하기 위해서는 단지 XML 데이터의 노드에 있는 레이블을 노드로 들어오는 간선상에 표현해 주면 된다.

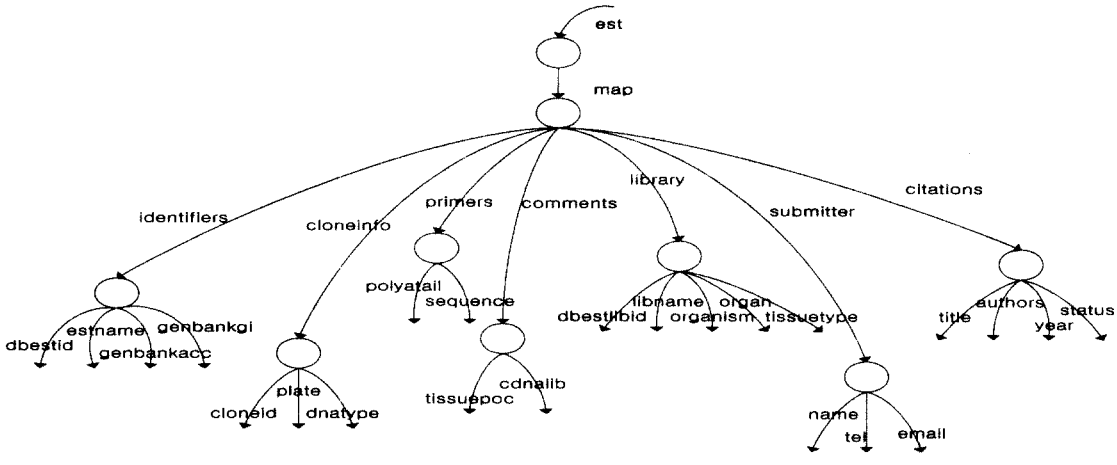
따라서 위의 방법을 통해 XML 문서로부터 생성된 반구조적 데이터 모델을 대상으로 하여 기존의 스키마 추출 기법과 이 논문에서 제안하는 시뮬레이션을 이용한 스키마 추출 기법을 적용하여 XML 문서로부터 최대 경계 스키마와 최소 경계 스키마를 추출할 수 있고 이 두 스키마를 바탕으로 스키마 트리의 추출이 가능해진다.

3.2 데이터가이드를 이용한 최대 경계 스키마 추출

데이터가이드는 데이터베이스 구조를 간결하고, 정확하게 나타내는 스키마로 정의된다[8]. 즉, 데이터가이드는 데이터 소스의 모든 유일한 레이블 경로를 데이터 소스에 나타나는 빈도에 상관없이 한번만 기술한다. 데이터 소스에 나타나지 않는 경로는 데이터가이드에 나타나지 않는다. 데이터



(그림 3-2) 반구조적 데이터 모델



(그림 3-3) 최대 경계 스키마

가이드에 대한 이러한 특성은 반구조적 데이터의 최대 경계 스키마 추출을 가능하게 해준다.

(그림 3-2)에서 보여지는 데이터 그래프의 경우에, root노드 {&1}과 &dg1로 초기화되어 있는 데이터가이드에서 시작하여 &1의 모든 자식노드에 대해서 < label, oid > 순서쌍을 원소로 하는 집합 P를 생성하고 P집합의 원소들에 대해서 label이 같은 oid들의 집합들 즉, < label, { oid, oid … }>가원소가 되는 집합 T를 생성한다. 집합 T의 모든 원소들은 새롭게 데이터가이드에 삽입될 노드(예 : &dg 1, &dg 2, &dg 3 …)를 생성하고 이것을 데이터가이드에 삽입한다.

이때 삽입 노드를 생성하기 전에 T의 원소인 < label, { oid, oid, … } > 순서쌍에서 {oid, oid, … }가 해쉬 테이블에 존재하지 않는다면 해쉬 테이블에 삽입하고 노드를 생성하여 현재 노드에 연결한다. 그러나 이미 존재한다면 새로운 노드를 생성하지 않고 해시 테이블에 존재하는 노드 값에 해당하는 데이터가이드의 노드를 현재 노드에 연결한다. 위와 같은 방법을 현재 상태가 단말 노드의 집합으로 이루어질 때까지 반복한다. (그림 3-3)은 이러한 과정을 거쳐 추출한 최대 경계 스키마를 보여준다.

4. 최소 경계 스키마

4.1 시뮬레이션

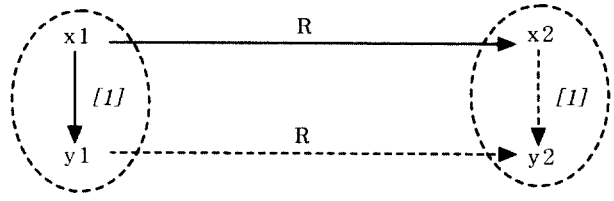
정점의 집합을 V라 하고, $E \subseteq V^2$을 만족하는 간선의 집합을 E, 그리고 정점 v 를 label 《v》로 매핑하는 함수를 《·》: V → A이라 할 때 레이블이 있는 그래프(labeled graph) G = (V, E, A, 《·》)에 대한 시뮬레이션의 정의는 다음과 같다.

정의 4.1 정점 v를 계승하는 정점(successor)들을 post (v) = {u | (v, u) ∈ E}라 하면 정점들의 집합상에서 이진 릴레이션(binary relation) $\leq \subseteq V^2$인 이진 릴레이션에 대해 u ≤ v 은 다음의 두 조건을 만족할 때 정점 v 는 정점 u 를 시뮬레이트(simulate)한다

(1) 《u》 = 《v》

(2) u'∈ post(u)인 모든 정점에 대해 u' ≤ v'이고 v'∈ post(v)인 정점 v'가 존재한다.

이런 시뮬레이션을 이루기 위한 조건은 아래 (그림 4-1)에서 잘 보여진다.



(그림 4-1) 시뮬레이션 다이어그램

(그림 4-1)에서 직선으로 이루어진 간선들이 존재할 때 점선들로 이루어진 간선들이 존재하고 이에 대응하는 정점 y2가 존재하면 이진 릴레이션 R은 시뮬레이션이 된다. 다시 말해서, 간선 x2[l]y2는 x1[l]y1을 시뮬레이트 한다. 이러한 시뮬레이션을 반구조적 데이터의 스키마를 추출하는데 적용하기 위해서는 기존의 정의에 아래의 조건이 더 필요하다.

두 그래프를 반구조적 데이터의 인스턴스를 나타내는 데이터 그래프와 이에 대한 스키마 그래프라고 하면

조건 1. 데이터 그래프의 간선 x1[l]y1이 간선 x2[l']y2에 시뮬레이트 되어야 한다.
이때 l'는 레이블 l에 대응되는 레이블이나 와일드카드(_)를 나타낸다.

조건 2. 시뮬레이션 관계의 두 그래프는 루트가 존재해야 한다. 즉 데이터 그래프와 스키마 그래프의 루트 r과 r'에 대해서 rRr'가 존재해야 한다. 마지막으로,

xRy에서 y가 원자 타입의 노드이고 스트링이나 정수형 같은 타입의 값을 가지면 x도 반드시 원자 타입의 노드이고 같은 타입을 값을 가져야 한다.

이러한 시뮬레이션의 개념은 데이터 그래프와 스키마 그래프 사이에 대한 관계의 유효성을 검사하는 데 이용할 수 있다. 하지만 스키마 그래프가 생성되기 이전에 주어진 데이터 그래프에 대한 스키마 그래프를 생성하기 위해서는 주어진 데이터 그래프의 어떤 노드가 같은 그래프내의 다른 노드와 시뮬레이션 되는지를 판단하여 스키마 추출에 이용할 수 있다.

4.2 시뮬레이션을 이용한 최소 경계 스키마 추출

시뮬레이션을 이용하여 스키마를 추출하기 위해서는 먼저 몇 가지 정의가 요구된다. 먼저, 주어진 그래프 G의 임의의 정점 v에 대해 시뮬레이션 관계에 있는 정점들의 집합을 sim(v)라고 정의한다. 즉, sim(v)는 v가 가지고 있는 출력 간선을 포함하는 정점들을 의미한다. 또한, 주어진 임의의 정점 v에 대해서 자식 정점들과 부모 정점들을 post(v) = {u | (v, u) ∈ E}와 pre(v) = {u | (u, v) ∈ E}로 각각 정의할 수 있는데 여기서 E는 그래프에 속하는 간선들의 전체 집합을 의미한다.

데이터로그를 이용하여 스키마를 추출할 경우에 스키마를 얻기 위한 순환조건으로 객체의 내향 프리디킷의 확장에 대해서 만족하는지를 검사한다. 마찬가지로 시뮬레이션을 이용하는 경우에도 이러한 순환 조건이 수행되어져야 하기 때문에 임의의 정점 v에 대해서 remove(v)라는 함수를 정의해야 한다. 이것은 어떤 정점 v에 대해서 v의 pre(v)에 속하지 않는 객체들의 집합을 의미한다. 만약 어떤 정점 v에 대해서 초기에 sim(v)의 집합을 얻었을때 이 sim(v)에 속하는 모든 객체들이 간선을 통한 객체들간의 관계가 고려되지 않은 상태가 된다. 그러므로 어떤 정점 v가 주어지면 그 정점 v의 pre(v)의 원소 u에 대한 sim(u)를 구하고 sim(u)에서 remove(v)에 해당하는 객체들을 제거함으로써 간선을 통한 객체들 상호간의 관계를 고려할 수 있게 된다.

(그림 4-3)은 이와 같은 이론을 기반으로 스키마를 추출하는 알고리즘을 보여준다. (그림 4-3)에 나타나는 알고리즘은 두 단계의 과정을 통해서 주어진 데이터 그래프 G에 대한 스키마를 추출해 낸다. 여기에서는 시뮬레이션 알고리즘이 (그림 3-2)의 트리 형태의 데이터뿐만 아니라 그래프 형태의 XML 데이터에서도 스키마 추출이 가능함을 보이기 위해 (그림 4-4)를 대상으로 스키마를 추출하는 과정을 보인다.

첫 번째 단계에서는 데이터 그래프 G에 존재하는 복합객체 v에 대한 유사 집합 sim(v)를 얻는다. 예를 들어 노드 &p2는 work-for와 name 그리고 managed_by의 레이블을

```
foreach (node in graph G) {
  labels = getNodeLabels (v) ;
  foreach (v'in graph G) {
    labels' = getNodeLabels (v) ;
    if(labels ⊆ labels')
      sim (v). add (v') ;
  }
  remove (v) = pre (v) - pre (sim (v)) ;
}
prevsim (v) = v ;
while (vertex v such that remove (v) ≠ φ) {
    {assert for v, remove (v) = pre (prevsim (v)) - pre (sim (v)) ; }
    foreach (node in graph G) {
      u = pre (v) ;
      foreach (p in remove(u)) {
        foreach (w in remove(v)) {
          if (w ∈ sim (p)) {
            sim (p) = sim (p) - w ;
            foreach (w" in pre(w)) {
              if (post(w") ∩ sim(p) = φ)
                remove (p) = remove (p) ∪ w" ;
            } // foreach
          } // if
        } // foreach
      } // foreach
    } // foreach
    prevsim (v) = sim (v) ;
} // while
```

(그림 4-3) 최소 경계 스키마 추출 알고리즘

가지는 출력간선을 포함하고 있다. 이러한 출력간선을 포함하고 있는 노드들의 집합은 {&p2, &p3, &p5, &p7, &p8}이 되고 이 집합은 sim(&p2)를 나타낸다. 이러한 방식으로 sim(&r) = {&r}, sim(&p1) = {&p1}, sim(&p2) = {&p2, &p3, &p5, &p7, &p8} = sim(&p3, &p5, &p7, &p8}, sim(&p4) = {&p1, &p4, &p6}, sim(&p5) = {&p3, &p5, &p7, &p8}, sim(&p6) = {&p1, & &p4, &p6}, … 등의 sim(v)를 얻을 수 있다.

그러나 위에서 어떤 정점 v에 대해서 구해진 sim(v)는 단지 레이블의 비교를 통해서 공통된 최소 레이블에 따라 객체를 분류한 것이다. 따라서 이것은 객체들간의 간선에 의한 관계를 표현하지 못한다. remove(v)는 어떤 정점v의 pre(v)에서 이 정점의 sim(v)의 선행자에 해당하는 정점들을 뺀 집합으로 이러한 문제를 고려하고 있다. 예를 들어 정점 &p1의 경우 pre(&p1) = {&r, &p2, &p3}이며 pre(sim(&p1)) = {&r, &p2, &p3}가 된다. 그러므로 remove(&p1)은 φ이 되고 remove (v) = pre (prevsim (v)) - pre (sim (v))에 의해 결국 remove (&p1) = {&p1, &p4, &p5, &p6, &p7, &p8}이 된다. 이 remove(v)는 다음 단계에서 객체들 사이의 간선을 통한 관계를 설명하는데 이용된다.

두 번째 단계에서는 첫 번째 단계에서 얻은 sim(v)의 원소에 대해서 remove(v)를 이용하여 불필요한 원소들을 제거하고 prevsim(v)의 집합을 갱신하는 과정을 통해서 최종 sim(v)의 집합을 구할 수 있다. 예를 들어 정점 &p1의 경우에 pre(&p1) = {&r, &p2, &p3}이기 때문에 u의 집합이 {&r, &p2, &p3}가 된다. 일단 u의 집합이 결정되면 sim(&r), sim(&p2), sim(&p3) 각각의 집합에 대한 갱신을 하게 된

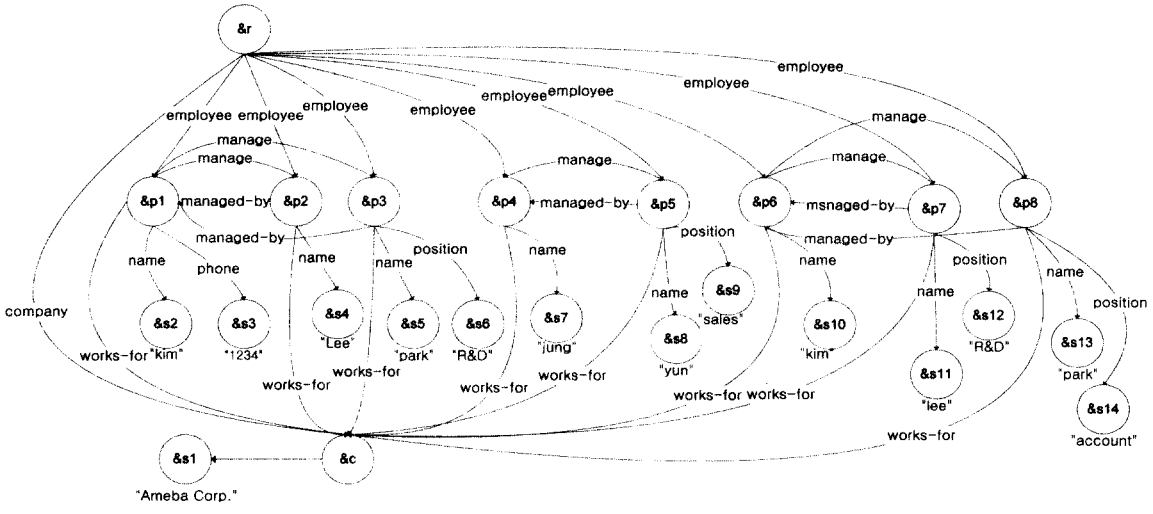
다. 첫 번째 단계에서 구한 u에 대한 시뮬레이션은 각각 sim(&r) = { }, sim(&p2) = {&p2, &p3, &p5, &p7, &p8} 그리고 sim (&p3) = {&p3, &p5}이며 이러한 sim(u)에 대한 집합에 대해서 remove(&p1) = {&p1, &p4, &p5, &p6, &p7, &p8}에 해당되는 객체들을 제거하면 sim(&r) = { }, sim(&p2) = {&p2, &p3} 그리고 sim(&p3) = {&p3}이 된다. 이렇게 sim(u)를 갱신하면 초기에 sim(u)에서 삭제된 개체들 때문에 sim(u)에 대한 remove(u)도 갱신되어야 한다. 즉 제거로 인하여 sim(u)의 객체수가 감소했기 때문에 remove(u)에 속하는 객체들의 개수는 당연히 늘어난다.

예를 들어 v가 &p1을 때 sim(&p2)가 갱신된다. 이 때 w"에 해당되는 집합은 {&p5, &p7, &p8}이다. 이 각각의 객체에 대해서 pre(&p5), pre(&p7), pre(&p8)을 구하고 pre(w)에 속하는 객체의 w"의 post(w")를 구하여 sim(u)와 공통된 객체가 있는지 없는지를 검사하여 만약 공통된 객체가 없으면 w"를 remove(u)에 포함시킨다. pre(&p5) = {&r, &p4}이 post (&r) = {&c, &p1, ⋯, &p8}, post(&p4) = {&p5, &c}이다. 여기서 post(&r)과 sim(&p2)의 공통객체가 존재하므로 remove (&p2)를 갱신하지 않고 post(&p4)와 sim(&p2)의 공통객체가 존재하지 않으므로 remove(&p2)에 &p4를 추가한다.

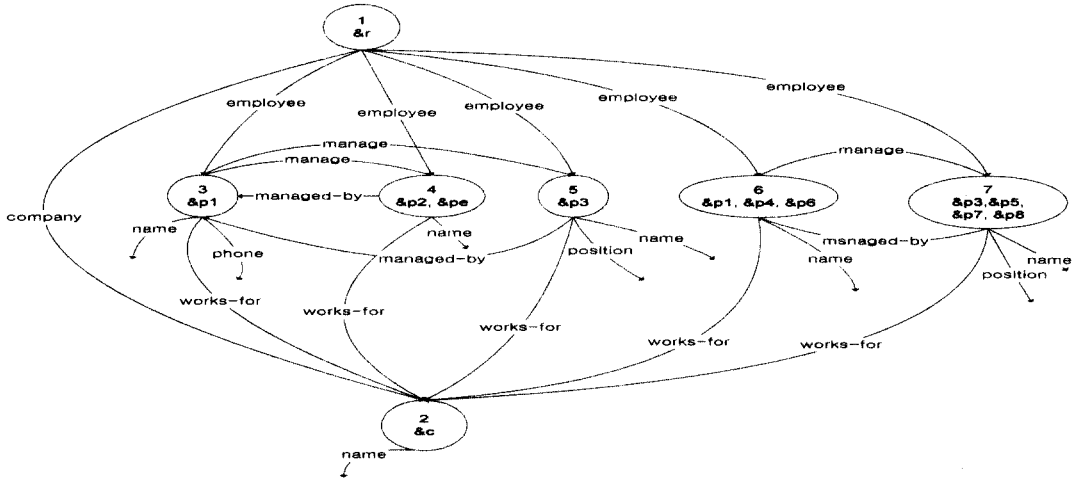
위의 두 단계를 통하여 얻어지는 최종적인 결과는 입력데이터 그래프 G에 있는 모든 복합 객체 v에 대한 sim(v)의 집합들이다. 이러한 시뮬레이션 알고리즘을 통해서 얻은 최소 경계 스키마는 (그림 4-5)에서 보여주고 있고 (그림 3-2)에 대한 최소 경계 스키마는 (그림 4-6)에서 보여주고 있다.

4.3 기존의 최소 경계 스키마 추출 기법과의 비교

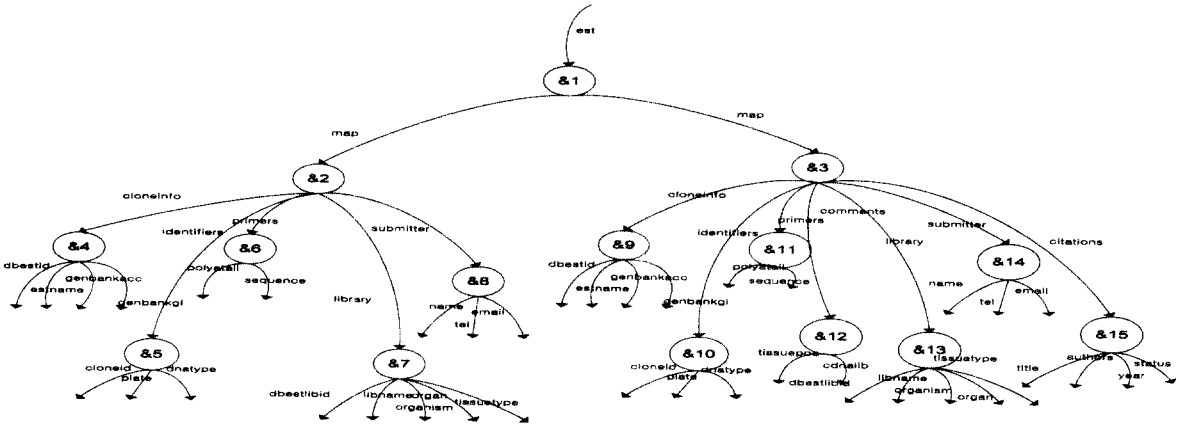
데이터로그는 최대 고정점(the greatest fixpoint)을 이용하여 타입을 분류함으로써 최소 경계 스키마를 추출할 수 있다. 즉, 데이터 그래프가 주어지면 먼저 복합 객체의 수만큼 내향 프리디킷을 생성하고 이 프리디킷들과 개체들의



(그림 4-4) 데이터 그래프



(그림 4-5) 최소 경계 스키마



(그림 4-6) 최소 경계 스키마

관계를 해시 테이블에 저장한다. 각각의 객체에 대해서는 출력 간선에 근거하여 데이터로그 규칙을 생성하고 이를 해시 테이블에 저장한다.

이렇게 생성된 데이터로그 규칙에 최대 고정점을 적용하여 타입을 분류할 수 있다. 하지만 이것은 단순히 공통되는 출력 간선들에 근거하여 타입을 분류한 것이기 때문에 객체들 사이의 간선을 통한 관계는 고려되지 않은 상태이다.

이러한 이유로 데이터로그 규칙에 존재하는 내향 프리디킷에 대한 확장까지 만족하는지에 대한 검사를 해야 한다. 마지막으로 동등한 프리디킷들을 하나의 프리디킷으로 대신하여 프리디킷의 수를 줄여나간다. 이는 데이터 그래프에 대해서 각 원소들의 공통되는 간선들만을 뽑아서 하나의 타입을 생성하는 것을 의미하므로 생성된 타입들의 집합은 결국 최소 경계 스키마의 추출을 의미한다.

그러나 XML 데이터의 모든 객체들이 최대 고정점에 도달하기 위해서는 반복적인 호출을 통해 접근해야 하기 때문에 스키마 추출시에 많은 비용을 요구하며 이에 따라 스키마 갱신의 부담을 증가시킨다.

시뮬레이션은 이러한 데이터로그와 밀접한 관계를 가지고 있다. 우선 시뮬레이션 알고리즘의 초기화는 데이터로그 규칙으로부터 생성되는 타입 릴레이션과 동일하다. 또한 시뮬레이션의 스키마 추출에 대한 그래프 생성 역시 데이터로그의 방법과 동일하다.

결론적으로, 시뮬레이션은 데이터로그 스키마에 대한 최대 고정점 계산을 대신하는 효율적인 알고리즘을 제공하게 되고 이렇게 데이터로그의 최대 고정점을 대신하여 시간 비용을 감소시키는 역할을 한다.

5. 스키마 트리

지금까지 반구조적 데이터의 스키마 추출 기법을 이용하여 XML 문서로부터 최대 경계 스키마와 최소 경계 스키마를 추출하였다. 최대 경계 스키마의 경우 주어진 데이터 그래프 D에 대해서 타입을 구분할 때 모호성이 발생하지 않는 반면 최소 경계 스키마에서는 모호성이 발생하게 된다. 예를 들어 (그림 4-4)에서 &1을 기준으로 map을 통해 도달할 수 있는 노드는 &2, &3으로 간선상의 레이블만을 가지고서는 타입을 구성하는데 있어서 모호성이 발생하게 된다.

따라서 같은 레이블을 가지는 간선들을 통합하여 이러한 모호성을 제거해 주어야 한다. 이렇게 하여 최대 경계 스키마와 최소 경계 스키마를 얻게 되면 우선 주어진 데이터 그래프로부터 중복이 되는 레이블에 대한 정보를 얻어내야 한다.

예를 들어 (그림 3-2)의 XML 데이터는 노드 &3이 citations이라는 두 개의 똑같은 레이블을 가진 간선을 가지고 있다. 이것은 문서상에 해당 엘리먼트가 여러번 중복되어 표현됨을 의미하기 때문에 해당 엘리먼트는 스키마 트리상에서 반드시 "*" 혹은 "+" 연산자로 표현되어야 한다.

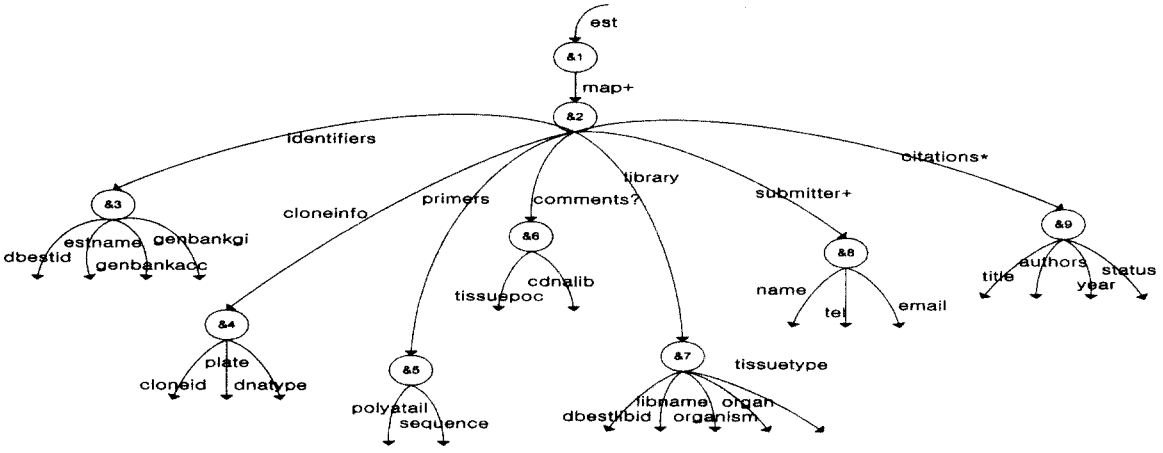
이러한 정보를 얻은 후에는 추출한 최대 경계 스키마와 최소 경계 스키마를 비교하여 중복되는 부분과 중복되지 않은 부분으로 구분한다. 여기서 중복되는 부분은 반드시 문서상에 나타나야 하는 부분을 나타내고 중복되지 않는 부분은 문서상에 나타날 수도 나타나지 않을 수도 있음을 암시한다. 따라서 레이블이 citations인 간선은 입력 간선의 레이블이 map인 노드로부터 여러개의 출력간선으로 표현될 수 있고 또한 존재하지 않을 수도 있기 때문에 스키마 트리상에 citaions*로 표시되어야 한다.

마찬가지로 레이블이 submitter인 경우는 반드시 문서내에 존재해야 하고 여러번 중복되어 존재 할 수도 있기 때문에, 다시 말해서, 한번 이상은 문서에 존재하기 때문에 스키마 트리상에 submitter+로 표시되어야 한다. (그림 5-1)은 이러한 정보를 포함한 스키마 그래프를 보여준다.

6. 스키마 트리와 관계형 스키마간의 매핑

6.1 관계형 스키마 매핑 규칙

XML 문서로부터 추출된 스키마 트리가 XML 문서와 관



(그림 5-1) 스키마 트리

계형 스키마사이의 매핑 정보를 생성하는데 사용되기 위해서는 객체-관계형 매핑 기법을 적용하여 스키마 트리를 각각의 관계형 스키마를 생성하기 위한 단위로 분해해야 한다.

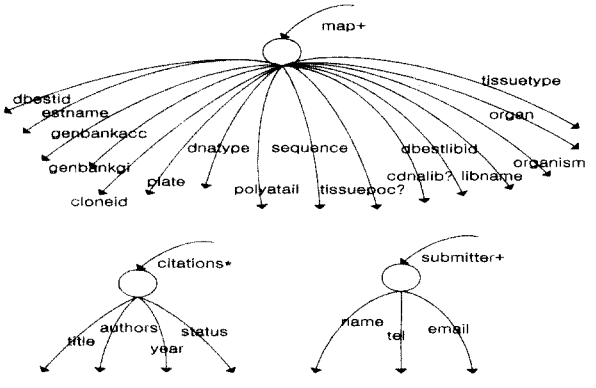
즉, 스키마 트리를 객체 트리로 인식함으로써 아래의 기준들을 기반으로 클래스를 테이블로, 속성은 컬럼으로 그리고 클래스의 상호관계는 후보키/외래키 관계로 매핑된다.

첫째, 객체-관계형 매핑 기법을 적용하기 위해 스키마 트리를 객체 트리로 인식하게 되면 스키마 트리의 각 엘리먼트는 클래스 타입과 속성 타입으로 나뉘어진다. 스키마 트리의 마지막 엘리먼트들 즉 리프 노드들이 속성 타입에 속하고 나머지 엘리먼트들은 클래스 타입에 속하게 된다.

둘째, XML 문서내의 엘리먼트가 계층 구조를 이루기 때문에 XML 문서로부터 추출한 스키마도 트리 형태의 계층 구조를 이루고 있다. 스키마 트리에서의 엘리먼트들간의 부모/자식 관계는 두 엘리먼트들의 타입에 따라 결정되어진다. 만약 두 엘리먼트들의 타입이 모두 클래스 타입이면 클래스-클래스 관계가 이루어져 관계형 스키마에서 후보키/외래키 관계로 매핑되고 만약 자식 엘리먼트가 속성 타입이면 클래스-속성 관계가 이루어져 자식 엘리먼트는 부모 엘리먼트의 속성으로 관계형 스키마에서 테이블과 테이블내의 컬럼으로 매핑된다.

셋째, 단일값 속성(single-valued property)은 클래스 테이블의 컬럼으로 매핑되거나 BLOB로 저장하기 위해서 새로 생성된 테이블의 컬럼으로 매핑되는 반면 다중값 속성(multi-valued property)은 별개의 테이블의 다중 튜플로 매핑된다.

넷째, 스키마 트리내의 계층구조를 줄여 불필요한 테이블의 생성을 막는다. 먼저 XML 문서의 루트 엘리먼트를 제거할 수 있다. 루트 엘리먼트를 제거할 수 있는 이유는 루트 엘리먼트가 XML 문서에는 단 하나의 루트 엘리먼트가 존재해야 한다는 XML의 조건을 만족하기 위해 존재하기 때문이다. 또 하나는 클래스 타입의 엘리먼트를 제거할 수 있다. 클래스타입의 엘리먼트를 제거하게 되면 제거되는 클래스의 속성들은 부모 클래스의 속성으로 취급된다.



(그림 6-1) 분해된 스키마 트리

예를 들어 (그림 5-2)에서 submitter+나 citations*인 레이블을 입력 간선으로 가지는 노드는 다중 튜플값을 가지고 있으므로 새로운 테이블을 할당하여 저장하고 특히, 레이블이 citations*인 노드는 이에 더하여 매핑되는 컬럼이 널값을 가질 수 있도록 설정해야 한다.

스키마 트리를 입력으로 하여 이와 같은 규칙들을 적용하면 테이블 생성을 위해 스키마 트리가 분해되는데 (그림 5-2)의 스키마 트리를 분해한 결과는 (그림 6-1)과 같다.

6.2 관계형 스키마 생성

(그림 6-1)의 분해된 트리는 관계형 스키마로 매핑이 가능한데 이때 서로간의 매핑 정보를 유지해야 한다. 스키마 트리와 관계형 데이형 테이블 사이의 매핑 정보는 XML 데이터를 저장하거나 사용자가 질의를 할 때 질의에 대한 결과를 추출해내는 과정에서 요구되어진다. 이러한 매핑 정보는 XML 형태로 생성되기 때문에 일관성있는 유지가 가능해진다.

```
<table>
  <tablename elementname = "map" > map < /tablename>
  <column elementname = "dbestid" > dbestid < /column>
  <column elementname = "estname" > estname < /column>
  <column elementname = "genbankacc" > genbankacc < /column>
  <column elementname = "genbankgi" > genbankgi < /column>
  <column elementname = "cloneid" > cloneid < /column>
  <column elementname = "plate" > plate < /column>
  <column elementname = "dnatype" > dnatype < /column>
  <column elementname = "polyatail" > polyatail < /column>
  <column elementname = "sequence" > sequence < /column>
  <column elementname = "tissuepoc" > tissuepoc < /column>
  <column elementname = "cdnalib" > cdnalib < /column>
  <column elementname = "dbestlibid" > dbestlibid < /column>
  <column elementname = "libname" > libname < /column>
 <column elementname = "organism" > organism < /column>
  <column elementname = "organ" > organ < /column>
   <column elementname = "tissuetype" > tissuetype < /column>
   <primarykey > mid < /primarykey>
   <intertable elementname = "citations"/ >
   <intertable elementname = "submitter"/ >
 </table>
```

(그림 6-2) map 테이블에 대한 매핑 정보

```
<table>
  <tablename elementname = "citations" > citation < /tablename>
  <column elementname = "title" > title < /column>
  <column elementname = "authors" > authors < /column>
  <column elementname = "year" > year < /column>
  <column elementname = "status" > status < /column>
  <primarykey > cid < /primarykey>
  <intertable elementname = "map">
    <candidatekey > mid < /candidatekey>
    <foreignkey > mid < /foreignkey>
  <intertable>
</table>
```

(그림 6-3) citations 테이블에 대한 매핑 정보

```
<table>
  <tablename elementname = "submitter" > submitter </tablename>
  <column elementname = "name" > name < /column>
  <column elementname = "tel" > tel < /column>
  <column elementname = "email" > email < /column>
  <primarykey > sid < / primarykey>
  <intertable elementname = "map">
    <candidatekey > mid< / candidatekey>
    <foreignkey > mid< / foreignkey>
  <intertable>
</table>
```

(그림 6-4) submitter 테이블에 대한 매핑 정보

(그림 6-2), (그림 6-3), (그림 6-4)는 분해된 스키마 트리에 대한 매핑정보를 나타낸다.

tablename 엘리먼트는 클래스 타입의 노드가 매핑되는 테이블명을 포함하고 column 엘리먼트는 테이블내의 각 컬럼에 대한 정보를 가지게 된다. column 엘리먼트의 속성중 childtable은 새로 생성되는 테이블을 포함하고 candidatekey 엘리먼트는 후보키가 되는 컬럼을 지정한다. 따라서 노드 &2는 map이라는 테이블을 생성하고 테이블내에 dbestid, estname등의 컬럼을 가진다. 레이블이 citations와 submitter인 노드는 각각 citations 테이블과 submitter 테이블로 매핑되고 map 테이블의 mid는 citationas 테이블과 submitter 테이블의 cid, sid와 각각 후보키/외래키 관계가 이루어진다.

(그림 6-2)에서 보면 tablename 엘리먼트는 클래스 타입의 노드가 매핑되는 테이블명을 포함하고 tablename 엘리먼트의 elementname 애트리뷰트는 스키마 트리에서의 입력 레이블명을 나타낸다. 또한 column 엘리먼트는 테이블내의 각 칼럼에 대한 정보를 가지게 되고 intertable 엘리먼트는 스키마 트리에서 하위 노드들중에서 새로운 테이블로 생성되어지는 노드들에 대한 정보를 포함한다. interclass 엘리먼트의 하위엘리먼트인 candidatekey 엘리먼트와 foreignkey 엘리먼트는 후보키와 외래키를 각각 지정한다. 따라서 노드 &2는 xperson이라는 테이블을 생성하고 테이블내에 first, last, street등의 컬럼을 가진다. 그리고 person 테이블의 pid 컬럼은 email 테이블과 job 테이블 각각의 pid와 후보키/외래키 관계를 형성한다. 데이터 타입은 기본적으로 varchar형태로 저장되며 데이터 특성에 따른 데이터 타입을 지정하기 위해서는 개발자가 테이블 매핑 정보를 직접 수정함으로써 가능해진다. <표 6-1>, <표 6-2>,

〈표 6-1〉 map 테이블

mid	dbestid	estname	genbankacc	genbankgi	cloneid	···
01	6002866	601572602T1	BE738181	10152173	3'	···
02	6002867	601572604T1	BE738182	10152174	3'	···
03	6002868	601572605T1	BE738183	10152175	3'	···
04	6002869	601572608T1	BE738184	10152176	3'	···
···	···	···	···	···	···	···

〈표 6-2〉 citations 테이블

cid	mid	title	authors	year	status
01	02	National Institutes of Health	S. Staab	1999	Unpublished
02	02	National Institutes of Health	C. Braun	1999	Unpublished
03	02	National Institutes of Health	I. Bruder	1999	Unpublished
04	03	National Institutes of Health	A. Dusterhoft	1998	Published
···	···	···	···	···	···

〈표 6-3〉 테이블 submitter

sid	mid	name	tel	email
01	01	Robert Strausberg, ph. D.	(301) 496-1550	Robert_Strausberg@nih.gov
02	02	Robert Strausberg, ph. D.	(301) 496-1550	Robert_Strausberg@nih.gov
03	03	J. Simeon, ph. D.	(402) 369-1488	Simeon@nih.gov
04	03	Dallan Quass, ph. D.	(309) 566-2744	Dallan@nih.gov
...	...	...	...	...

<표 6-3>은 (그림 6-1)의 스키마 트리를 바탕으로 생성한 관계형 테이블을 보여준다.

7. 구현 및 실험

7.1 구현 환경

이 논문에서 소개하는 스키마 추출 알고리즘들은 모두 Java를 이용하여 Windows-NT 환경 하에서 구현되었으며 XML 데이터를 파싱하기 위해 Sun Project X Technology Release 2를 이용하였다. 또한 XML 문서로부터 스키마를 추출하여 데이터를 저장하기 위한 시스템으로 관계형 데이터베이스 시스템인 Oracle 7.3.4를 이용하였다.

7.2 시스템 구성

이 논문에서 제안하는 전체적인 시스템의 구성도는 (그림 7-1)과 같다. 시스템 구조는 XML 데이터를 읽어들여 관계형 데이터베이스 내에 테이블을 생성한 후에 XML 데이터를 저장한다. 사용자는 XML 데이터를 저장할 때 추출한 스키마 트리를 대상으로 XML_QL로 질의를 하게되고 XML_QL은 시스템 내에서 SQL로 변환되어 데이터베이스에 질의를 한다. 데이터베이스로부터 얻어낸 결과는 다시 XML 문서로 변화되어 사용자에게 전송되어진다.

시스템 내의 세부 컴포넌트의 기능은 아래와 같다.

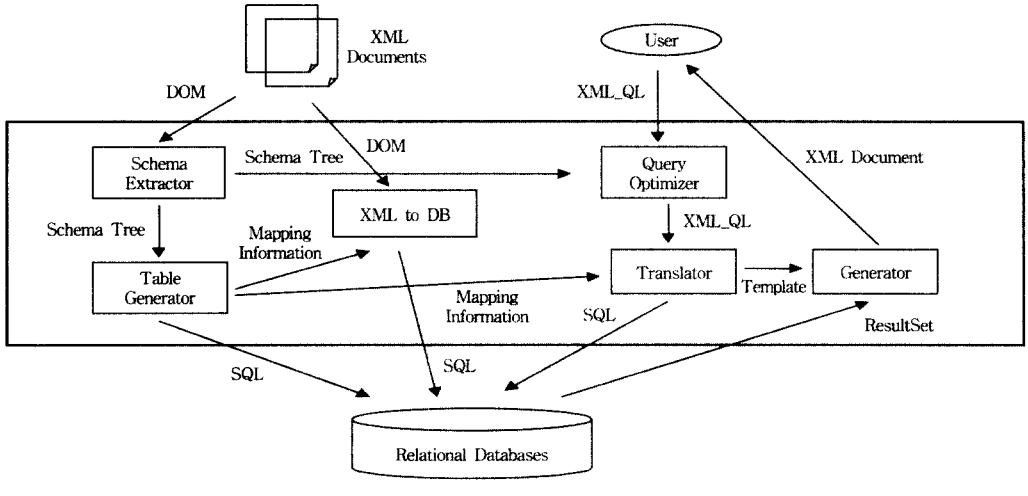
- 스키마 추출기(Schema Extractor) : XML 데이터로부터 최대/최소 경계 스키마를 추출한 후 이를 바탕으로 스키마 트리를 생성한다.
- 테이블 생성기(Table Generator) : 스키마 트리를 대상으로 하여 XML 형태의 매핑 정보를 생성하고 테이블을 생성하기 위한 SQL를 생성한다.
- XMLToDB : XML 데이터를 읽어들여 생성한 테이블에 데이터를 저장한다.
- 질의 최적기(Query Optimizer) : 사용자의 질의에 포함된 *, +, ?등의 연산자를 스키마 트리를 이용하여 제거함으로써 질의를 최적화 시킨다.
- 번역기(Translator) : XML_QL을 매핑 정보를 이용하여 대응되는 SQL문으로 변환한다.
- 생성기(Generator) : 데이터베이스로부터 얻은 데이터를 XML 형태로 변환하여 사용자에게 전송한다.

XML문서로부터 관계형 스키마를 추출하기 위해서는 먼저 XML문서를 받아들여 최대 경계 스키마와 최소 경계 스키마를 추출한 후 XML 문서형태의 두 스키마를 바탕으로 스키마 트리를 구성한다. 이때 최대 경계 스키마 추출은 데이터가이드 알고리즘을 이용하고 최소 경계 스키마 추출은 시뮬레이션 알고리즘을 이용한다.

스키마 트리는 규칙에 따라 분해되어 테이블로 매핑되기 위한 매핑 정보를 생성한다. 매핑정보는 관계형 스키마 생성을 위해 SQL문을 생성하고 테이블이 생성되면 매핑 정보를 참조하여 XML 데이터를 생성된 테이블내에 저장한다.

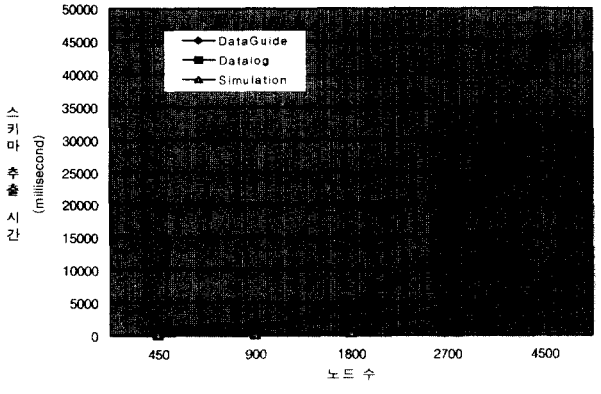
7.3 성능 평가

실험의 목적은 최대 경계 스키마 추출 알고리즘과 최소



(그림 7-1) 시스템 구성도

경계 스키마 알고리즘의 성능을 분석해 보고 특히, 시뮬레이션을 이용한 최소 경계 스키마 추출 알고리즘이 기존의 데이터로그를 이용한 최소 경계 스키마 추출 알고리즘보다 성능이 우수함을 보이는 것이다. 각 알고리즘의 성능을 평가하기 위해 (그림 2-1)과 같이 문서간의 링크를 포함하지 않는 트리 형태의 EST 데이터를 입력 데이터로 사용하였다. 실험은 문서내의 엘리먼트의 개수에 따른 스키마 추출 시간을 비교하였는데 (그림 7-3)에서 보는 것처럼 엘리먼트의 수가 증가할수록 시뮬레이션과 데이터가이드에 비해 데이터로그가 성능이 현격하게 저하됨을 알 수 있다.



(그림 7-3) 스키마 추출 시간

데이터로그가 시뮬레이션에 비해 상대적으로 성능이 낮은 이유는 최대 고정점에 도달할때까지 타입 릴레이션을 계속해서 비교해야 하기 때문이다. 데이터로그는 타입을 초기화할 때 내향 프리디킷의 조건을 체크하지 않지만 타입의 수를 감소시킬 때 내향 프리디킷의 조건을 체크하므로 재귀호출이 발생하게 된다. 이러한 재귀호출에 의해서 데이터의 양이 증가할수록 그리고 트리 형태가 아닌 그래프 형태의 데이터일 때 성능은 더욱 저하된다.

이에 비해 시뮬레이션을 이용한 최소 경계 스키마 추출 알고리즘은 그래프상의 정점을 n이라 하고 간선을 m이라 할 때 O(mn)의 시간을 보장하는 것이 가장 큰 특징이다. 물론 어떤 정점 v에 대해서 sim(v)을 구하는 초기화 단계는 $O(n^2)$의 시간이 소요되는데 이것은 데이터로그 방법에서도 마찬가지이다. 하지만 remove(v)라는 개념을 이용함으로써 어떤 정점 v에서 sim(v)에 속하지 않는 원소를 제거하는 방법을 이용한다. 이러한 remove(v)의 집합은 sim(v)이 초기화된 뒤에 모든 sim(v)에 대해서 v 정점으로 들어오는 간선들에 대한 검색을 이용하여 구할 수 있으므로 remove를 초기화하는데 정점과 간선의 곱인 O(mn)이라는 시간이 소요된다. 일단 remove(v)가 초기화되면 모든 정점에 대해서 sim(v)와 remove(v)를 이용하여 sim(v)를 갱신함으로써 타입화가 진행되는데 여기에 소요되는 시간 역시 O(mn)이 걸린다. 그러므로 시뮬레이션을 이용하여 구조정보를 추출하는데는 O(mn)이라는 시간이 걸린다.

데이터로그를 이용할 경우 초기 데이터 로그 규칙을 생성하고 타입 릴레이션을 만드는 데는 $O(n^2)$의 시간이 소요된다. 또한 최대 고정점에 도달할 때까지 타입 릴레이션을 계속해서 비교하는 동안 $O(n^2)$의 시간이 소요된다. 이는 초기 타입 릴레이션은 모든 복합 객체의 수($c <= n$)만큼 프리디킷이 생성되고 초기 프리디킷에 속하는 객체의 수 역시 n보다 더 큰 값($p >= n$)을 가지므로 $O(cp) \approx O(n^2)$이 된다.

8. 결 론

이 논문에서는 반구조적 데이터의 스키마 추출 기법을 이용하여 XML 문서로부터 스키마 트리를 추출하고 추출된 스키마 트리로부터 매핑 정보를 이용하여 관계형 스키마를 생성하고 생성된 매핑 정보를 유지하는 방법을 살펴보았다. 반구조적 데이터는 기존의 스키마와는 달리 고정된 스키마가 없고 주어진 데이터 인스턴스에 대해 하나 이상의 스키마가 존재한다. 이들 스키마는 크게 최대 경계 스키마와 최소 경계 스키마로 구분할 수 있고 데이터가이드를 이용하여 최대 경계 스키마를 추출할 수 있다. 최소 경계 스키마를 추출하는 기존의 방법은 데이터로그를 이용하는 것이었다. 그러나 이 논문에서는 스키마 추출을 위해 데이터로그를 이용하지 않고 보다 효율적인 방법으로 시뮬레이션을 이용하여 최소 경계 스키마를 추출하는 기법을 제시하였다.

XML 문서로부터 반구조적 데이터의 스키마 추출 기법을 이용한 스키마 트리의 추출은 상당한 장점을 가져온다. 먼저 XML 데이터를 관계형 데이터베이스에 저장할 경우 스키마 트리를 이용하여 효율적인 관계형 테이블 생성을 가능하게 하고 또한 데이터베이스로부터 데이터를 XML 형태로 출력할 경우 스키마 트리를 이용하여 XML 문서의 생성을 용이하게 해준다. 그리고 사용자가 질의를 할 경우에도 사용자에게 편리성을 제공해 준다.

또한 중첩 구조를 가진 스키마 트리를 관계형 스키마로 매핑하기 위해서 클래스를 테이블로 매핑하고 속성을 컬럼으로 매핑하는 객체-관계 매핑 기법을 적용하였다.

이 논문에서의 관계형 스키마 생성의 대상은 데이터 중심의 XML(data centric XML)이기 때문에 문서 중심의 XML(document centric XML)에서 중요시되는 속성(attribute)과 하위 엘리먼트(subelement)의 구분, 엘리먼트간의 순서, 그리고 엘리먼트들간의 링크등은 이 논문에서는 고려하지 않았다.

따라서 향후 연구로는 데이터 중심의 XML문서뿐만 아니라 문서 중심의 XML 문서에 적합한 관계형 스키마 생성에 대한 연구가 필요하고 아울러 매핑 정보의 유지에 있어서 개발자가 정의한 XML 형태의 정보보다 효율적이고 표준화된 방법인 XML Schema를 이용한 매핑 정보의 유지

에 대한 연구가 필요하다.

참 고 문 헌

[1] R. Bourret, C. Bornhovd, A. P. Buchmann. A Generic Load/Extract Utility for Data Transfer between XML Documents and Relational Databases. WECWIS'00, San Jose, California, June pp.8-9, 2000.

[2] Ronald Bourret. XML and Databases. Technical University of Darmstadt, 2000. http : //www.rpbourret.com/xml/XMLAndDatabases.htm.

[3] Michael J. Carey, Daniela Florescu, Zachary G. Ives, Ying Lu, Jayavel Shanmugasundaram, Eugene J. Shekita, Subbu N. Subramanian : XPERANTO : Publishing Object-Relational Data as XML. WebDB (Informal Proceedings), pp.105-110, 2000.

[4] M. Fernandez, WangChiew Tan, Dan Suciu. SilkRoute : Trading between Relations and XML. WWWg, 2000.

[5] P. Buneman, S. Davidson, M. Fernandez, and D. Suciu. Adding structure to unstructured data. In Proc. of the ICDT, 1997.

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

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

[8] D. Calvanese, G. Giacomo, and M. Lenzerini. What can Knowledge representation do for semi-structured data? In Proc. of the 15th National Conf. on Artificial Intelligence (AAAI-98), 1998.

[9] S. Abiteboul. Querying semi-structured data. In Proc. of th e Intl. Conf. on Database Theory (ICDT), 1997.

[10] The World Wide Web Consortium (W3C)'s DOM(Document Object Model) web page, 2000. http : //www.w3c. org/dom.

[11] M. Fernandez and D. Suciu. Optimizing regular path expressions using graph schemas. In Proc. of the Intl. Conf. on Database Theory(ICDT), 1997.

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

[13] 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 Mangement of Data, Dallas, Texas, 2000.

[14] R. Goldman, J. Widom. DataGuides : Enabling Query Formulation and Optimization In Semistructured Databases. In Proc. of the 23rd VLDB Conference Athens, Greece, 1997.

[15] P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu. A query language and optimization techniques for unstructured data. In SIGMOD, pp.505-516, Montreal, 1996.

[16] M. Henzinger, T. Henzinger, and P. Kopke. Computing simulation on finite and infinite graphs. In Proc. of the 20th Symposium on Foundations of Computer Science, pp.453-462, 1995.

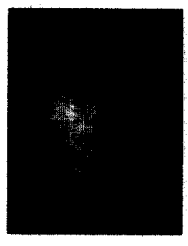
[17] P. Kilpelainen, H. Mannila, and E. Ukkonen. MDL learning of unions of simple pattern languages from positive exmamples. In Proc. of the European Conf. on Computational Learning Theroy, 1995.

[18] 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.

[19] S. Nestorov, J. Ullman, J. Wiener, and S. Chawathe. Representative Objects : Concise Representation of Semistructrued Hierarchical Data. ICDE, 1997.

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

[21] The World Wide Web Consortium (W3C)'s XML web page, 1998. http : //www.w3c.org/XML/.



박 경 현

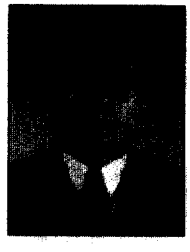
e-mail : hareton@etri.re.kr

1999년 충북대학교 컴퓨터공학과 졸업

2001년 충북대학교 대학원 전산학과 (이학석사)

2001년~현재 한국전자통신연구원

관심분야 : XML, 시공간데이터베이스, Wireless Intelligent Network



이 경 휴

e-mail : khyulee@etri.re.kr

1981년~1992년 숭실대학교/KAIST 전산학

1988년~현재 정보통신 기술사 위원

1994년~1996년 중경공업대학 겸임교수

1982년~현재 한국전자통신연구원 책임연구원

관심분야 : 고속통신망, LAN, 망관리, 성능평가등

류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr

1976년 숭실대학교 전산과 졸업

1980년 연세대학교 산업대학원 전산전공 (공학석사)

1988년 연세대 대학원 전산전공(공학박사)

1976년~1986년 육군군수지원사전산실 (ROTC장교), 한국전자통신연구소 (연구원), 한국방송통신대학교 전산학과(조교수) 근무

1989년~1991년 Univ. of Arizona 연구원(TempIS Project)

1986년~현재 충북대학교 전기전자컴퓨터공학부 교수

관심분야 : 시간지원 데이터베이스, 시공간 데이터베이스, Temporal GIS, 지식기반 정보검색시스템, 데이터마이닝, Bio-Informatics등