Journal of Korea Multimedia Society Vol. 9, No. 7, July 2006(pp. 773-781)

효율적인 릴레이션 생성과 제약조건 보존을 위한 새로운 Inlining 기법

안성철†, 김영웅††

요 약

XML(eXtensible Markup Language)은 웹 상의 데이터를 표현하고, 교환하기 위한 표준 언어로써, XML로 표현된 문서를 관계형 데이터베이스 관리시스템(RDBMS)에 저장하고 관리하는 기법에 대한 연구가 활발히 진행되어 왔다. 이러한 연구들은 입력으로 DTD(Document Type Definition) 문서를 받는 기법을 사용하고, 그것으로부터 관계형 스키마를 추론하는 기법을 사용한다. 하지만, 기존의 연구들은 DTD 간소화 절차때문에 semantic 보존을 고려하지 않는다. 더 나아가서, 기존의 연구들은 XML 데이터의 내용(content)와 구조(structure) 정보만을 저장하는데 초점이 맞춰져 있기 때문에, XML 문서 저장 시 데이터의 무결성을 보장하기 위해 저장 프로시져나 트리거를 사용해야 하는 번거로움이 생긴다. 본 논문에서는 [1]의 연구에서 제시한 Inlining 기법을 기반으로 기존의 Inlining 기법의 문제점인 DTD에서 추론할 수 있는 의미적인 정보의 손실을 관계형 스키마로 보존하는 방법과 효율적인 릴레이션 생성을 위해 개선된 Inlining 기법을 제시한다.

New Inlining Method for Effective Creation of Relations and Preservation of Constraints

Sung-Chul An†, Young-Ung Kim††

ABSTRACT

XML is a standard language to express and exchange the data over the web. Recently, researches about techniques that storing XML documents into RDBMS and managing it have been progressed. These researches use a technique that are receiving the DTD document as an input and generate the relational schema from it. Existing researches, however, do not consider the semantic preservation because of the simplification of the DTD. Further, because existing studies only focus on the preservation technique to store information such as content and structure, there is a troublesomeness that have to use the stored-procedure or trigger for the data integrity during the stores of XML documents. This paper proposes a improved Inlining technique to create effective relations and to preserve semantics which can be inferred from DTD.

Key words: RDBMS(관계형데이터베이스시스템), XML(확장가능마크업언어), DTD(문서형정의), Inlining(인라이닝)

1. 서 론

인터넷 상에서 XML로 표현된 데이터들이 많아짐에 따라 이를 효율적으로 저장하고 관리하는 기법들에 대한 연구가 활발히 진행되었다. 이러한 연구들의 대표적인 기법으로는 연구[1,3]에서 제시된 Inlining

※ 교신저자(Corresponding Author) : 안성철, 주소 : 서울특별시 성북구 삼선동 3가 389(136-792), 전화 : 02)760-4438, FAX : 02)760-4436, E-mail : ascsg21@hansung.ac.kr
접수일 : 2005년 10월 5일, 완료일 : 2006년 3월 16일

† 준회원, 한성대학교 컴퓨터공학과
†† 한성대학교 컴퓨터공학과
(E-mail : yukim@hansung.ac.kr)
※ 본 연구는 2006년도 한성대학교 교내연구비 지원과제임.

기법이 존재한다.

Inlining 기법에서는 Input으로 DTD를 받아서 관계형 스키마를 추론해 내는 알고리즘을 사용한다. 이때, DTD로부터 관계형 스키마에 보존되어야 하는 정보로는 문서의 내용(content), 구조(structure), 의미(semantic)가 있다[5]. 그러나, 기존에 제시된 Inlining 기법들은 세 가지 정보 모두를 다 고려하지 못하고 특정 정보만을 저장하는데 초점이 맞추어져 있기 때문에 다음과 같은 문제점을 가지게 된다.

① DTD 간소화 절차를 거치면서 DTD에서 추론할 수 있는 NULL, NOT NULL, Cardinality Constraints와 같은 DTD의 의미적인 정보들의 손실이 발생할 수 있다.

해결책 - 기존에 제시된 DTD 간소화 절차를 의미적인 정보를 보존할 수 있도록 수정 제시한다.

② 연구 [1], [3]에서는 DTD의 의미적인 정보를 관계형 스키마로 매핑시킬 수 있는 기준을 제시하지 않고 있다.

해결책 - DTD의 의미적인 정보를 SQL의 제약조건으로 매핑할 수 있는 기준을 제시한다.

③ 연구 [2]에서는 DTD로부터 추론될 수 있는 의미적인 정보를 보존하고 있지만, 효율적인 릴레이션 생성에 관한 고려가 없다.

해결책 - 생성되는 릴레이션 수의 감소, 중복 데이터의 제거에 초점을 맞추어 개선된 기법을 제시한다.

본 논문에서는 위의 문제점을 해결하기 위해 그림 1과 같은 접근 방법을 보이는 NCPI-MDS(New Constraints-Preserving Inlining with Modified DTD Simplification) 기법을 제시한다. NCPI-MDS 기법에서는 의미적인 정보를 보존하기 위해 수정된 DTD 간소화 절차를 거치게 되고(ⓐ), 데이터의 중복을 줄이고, 효율적인 릴레이션 생성을 하기 위해 수정된 EHI(Efficient Hybrid Inlining) 알고리즘을 적용하고(ⓑ), DTD 내의 의미적인 정보들을 추론해 냄으로써 이를 결과 관계형 스키마에 재작성하게 되는 과정을 거친다(ⓒ).

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고, 3장에서는 Hybrid Inining 기법의 문제점을 개선한 NCPI-MDS 기법을 제안하고, 4장에서는 두 가지 기법에 의해 생성된 릴레이션에 대한 비교분석을 보여준다. 마지막으로 5장에서는 본 논문의 결론을 맺는다.

2. 관련 연구

[3]의 연구에서는 Basic, Shared, Hybrid Inlining의 세 가지 기법을 제안하고 있다. 각 Inlining 기법을 요약하면 다음과 같다.

(1) Basic Inlining

● DTD 그래프에서 모든 노드들에 대해 자식 노드들을 inline 시킨 후, 각 노드들에 대해서 릴레이션을 생성한다.

● "*" edge(부모 엘리먼트와 자식 엘리먼트가 "*" 연산자를 사용해서 정의되어 있는 경우)로 연결되는 자식노드는 새로운 릴레이션으로 생성한다.

● 두 개의 노드 사이에 recursive가 존재하면 둘 중 하나의 엘리먼트를 새로운 릴레이션으로 생성한다.

(2) Schared Inlining

● in-degree edge(DTD 그래프에서 노드로 들어오는 간선)의 수가 1보다 큰 노드는 새로운 릴레이션으로 생성한다(다수의 엘리먼트에 의해 공유되는 엘리먼트).

● in-degree edge가 0인 노드는 새로운 릴레이션으로 생성한다(root 엘리먼트).

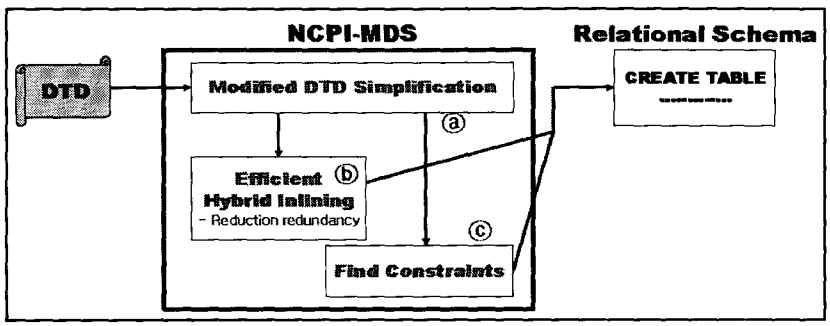


그림 1. NCPI-MDS 기법

● "*" edge로 연결되는 자식노드는 새로운 릴레이션으로 생성한다.

● 두 개의 노드 사이에 recursive가 존재하면 둘 중 하나의 엘리먼트를 새로운 릴레이션으로 생성한다.

● 나머지 in-degree가 1인 노드는 모두 부모 노드로 inline 시킨다.

(3) Hybrid Inlining

● "*" edge나 recursive가 존재하지 않고, in-degree edge가 1보다 큰 노드들을 부모 노드로 inline 시킨다는 것을 제외하고는 Shared Inlining 기법과 동일하다.

위의 [3]의 연구에서 제시하는 Inlining 기법은 특정 DTD로부터 XML 문서의 내용과 구조를 추론하는데 초점을 맞춘 기법이다. 이에 반해, 연구 [2,5]에서 제시하는 기법은 연구 [3]에서 제시한 Inlining 기법을 기반으로 DTD 문서에서 의미적인 정보들을 생성되는 릴레이션으로 보존하는 기법을 보여주고 있으며, 요약하면 다음과 같다.

(1) X2R Inlining

● Inlining 기법은 연구 [3]에서 제시한 Hybrid Inlining 기법을 사용한다.

● XML key와 keyref 정보를 보존하는 constraint relations라는 개념을 도입하여 키 정보를 보존한다.

● XML 문서의 엘리먼트들 사이의 부모-자식 관계의 정보를 보존하기 위해 각 자식 릴레이션에 parent-id 속성을 추가하여 보존한다.

(2) CPI(Constraints-Preserving Inlining)

● Inlining 기법은 연구 [3]에서 제시한 Shared Inlining 기법을 사용한다.

● DTD로부터 Domain, Cardinality Constraints와 Inclusion, Equality-Generating, Tuple- Generating, Dependencies와 같은 의미적인 제약조건들을 추론해낸다.

● 추론된 제약조건들을 기존 Inlining 기법을 통해 추론한 릴레이션에 rewrite한다.

X2R Inlining과 CPI는 DTD로부터 의미적인 정보를 찾아내는데 초점이 맞추어져 있기 때문에 효율적인 릴레이션 생성에 대해서는 고려하지 못하고 있다. 또한, DTD의 간소화 절차가 생략되어 있는데, 이는 각 기법들이 DTD의 복잡성을 다루지 못함을 알 수 있다.

3. NCPI-MDS(New Constraints-Preserving Inlining with Modified DTD Simplification)

3.1 DTD 간소화(Simplification)

DTD는 XML 문서의 구조 정보를 기술하기 위해 제시된 방법 중의 하나이다. 그러나, DTD 간소화 규칙을 거치면서 릴레이션 스키마로 보존되어야 하는 정보들이 손실되고, 이로 인해 효율적이지 못한 스키마를 생성하게 된다. 그림 2는 연구 [1]에서 제시되고 있는 DTD 간소화 규칙이다.

그림 2에서 5가지의 DTD 간소화 규칙이 제시되어 있는데, 그 중 DTD에서 추론할 수 있는 정보임에도 불구하고 간소화 규칙을 거치면서 정보의 손실을 초래하는 규칙은 규칙 1, 2, 3이다. 각각 "+", "?", "|" 연산자가 다른 연산자로 변환됨으로써 대응수 제약조건의 의미가 손실되거나, 효율적이지 못한 릴레이션을 생성하게 되는 결과를 초래한다. 이러한 의미를 스키마 생성 시에 보존하기 위해 수정된 DTD 간소화 절차는 그림 3와 같다.

그림 3의 수정된 DTD 간소화 절차에서 남겨진 규칙들 또는 추가된 규칙들은 자명한 규칙들만 남게 된다. 예를 들어, 규칙 7에서 볼 수 있듯이 {0, ...}의 발생이 가능한 엘리먼트와 {0, ...}의 발생이 가능한 엘리먼트를 더하면 당연히 해당 엘리먼트는 {0, ...}의 발생이 가능하다는 하나의 엘리먼트 정의로 간소화될 수 있다. 또한, {0, ...}의 발생이 가능하다는 하나의 엘리먼트로 간소화됐음에도 원래의 의미적인 정보인 {0, ...}의 발생이 가능하다는 것이 손실되지 않는다.

3.2 EHI 알고리즘

기존의 Hybrid Inlining 알고리즘을 기반으로 하면서 효율적인 릴레이션 생성을 위해 수정된 NCPI-MDS 알고리즘은 다음과 같다.

① DTD 그래프에서 새로운 릴레이션으로 만들어져야 하는 top nodes[2]를 시작으로 DFS 탐색을 한다. top nodes의 식별은 다음과 같다.

- DTD 그래프 내에서 어떠한 노드들로부터 도달할 수 없는 노드(최상위 노드)
- 부모 노드에서 자식 노드로 "*" 연산자나 "+"

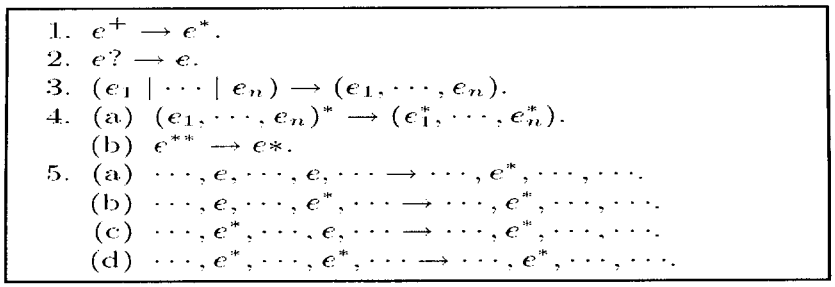


그림 2. DTD 간소화 규칙

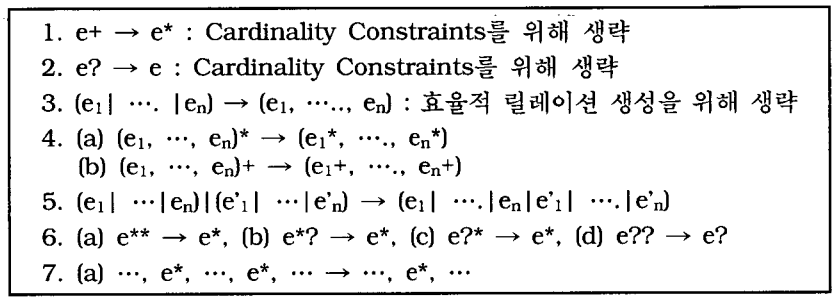


그림 3. 수정된 DTD 간소화 절차

연산자에 의해 연결된 자식 노드

- recursive가 존재하는 두 노드 중 하나의 노드

② top node를 시작으로 DFS 탐색을 하는 동안 다른 top node를 만나지 않는 한 모든 도달 가능한 leaf nodes들을 최초 top node로 inline 시킨다. DTD에 정의된 어떠한 엘리먼트도 root 엘리먼트가 될 수 있으므로 생성된 테이블에 저장된 튜플을 구분하기 위한 nodetype이라는 속성을 스키마에 추가한다.

③ DFS 탐색 중 "|" 연산자를 사용하는 노드를 만나면, 이 연산자로 연결된 엘리먼트가 leaf node일 경우는 모든 엘리먼트를 하나의 칼럼에 저장하기 위해 pcdata 칼럼을 추가하고, 실제 저장된 엘리먼트를 구별하기 위해 nodetype 칼럼을 추가한다. 또한, leaf node가 아닌 중간 노드이고, 새로운 릴레이션으로 만들어져야 한다면 모든 엘리먼트를 저장하기 위한 통합된 하나의 릴레이션을 생성한다. ③의 방법을 통해 필요 없는 칼럼의 생성을 막고, 많은 NULL 값의 발생과 줄일 수 있다.

④ ② ~ ③까지의 절차를 반복해서 실행한 후 추론된 릴레이션 중 T1(ID), T2(ID) 또는 T1(ID, value1), T2(ID, value1) 형태를 갖는 릴레이션이 적어도 두 개 이상 존재한다면, 이들 릴레이션을 TABLE (ID, nodetype) 또는 TABLE(ID, nodetype, value) 형태의 하나의 릴레이션으로 통합시킨다. ④의 방법을 통해서는 ID 값만을 갖는 릴레이션을 통합, 관리함으로써 XML 문서에 특정 질의 시에 조인 비용을 감소시킬 수 있다.

⑤ 마지막으로, "*" 연산자에 의해 만들어지는 릴레이션과 부모 릴레이션 사이의 관계정보를 저장하는 asterIsk(parentID, childID, parentType, chIldType)의 릴레이션을 추가한다. 이 릴레이션을 이용해서 DTD 내에서의 "*" 연산자로 연결된 부모, 자식 노드의 관계정보를 저장한다. 이러한 스키마 생성은 XML 데이터 저장 시에 중복을 제거해 줄 수 있다. 또한, 일반적인 기본키와 외래키의 조인에 의한 부모 엘리먼트에서 자식 엘리먼트를 탐색하는 하향식 탐색에 있어서의 탐색비용을 단축시켜 줄 뿐만 아니라, 자식 엘리먼트에서 부모 엘리먼트를 탐색하는 상향식 탐색에 있어서도 탐색비용을 줄이는데 도움이 된다[1].

3.3 DTD 내의 의미적인 정보[2]

DTD에서 추론할 수 있는 의미적인 정보와 관계형 스키마에서 매핑될 수 있는 제약조건들은 다음과 같다.

도메인 제약조건 : DTD에서 추론할 수 있는 도메인 제약조건은 속성의 값이 특정 집합의 값으로써

한정될 때 확인할 수 있다. 예를 들어, <!ATTLIST author gender (male | female) #REQUIRED married (yes | no) #IMPLIED>와 같은 애트리뷰트 정의에서 author 엘리먼트는 속성으로 gender와 married를 가지고, gender라는 속성은 값으로써 "male" 또는 "female" 중 한 값을 갖고, married라는 속성은 값으로써 "yes" 또는 "no" 중 한 값을 가질 수 있다. 이러한 특성은 관계형 스키마에서 CHECK 절을 사용하여 보존될 수 있다. 또한, #REQUIRED 와 #IMPLIED와 같은 속성은 NULL, NOT NULL로 보존이 될 수 있다.

대응수 제약조건 : DTD 내에서 추론이 될 수 있는 대응수는 표 1과 같다.

Inclusion Dependencies : IDs는 XML 문서 내의 특정 속성의 값이 다른 속성의 값으로 나타나야만 한다라는 의미로써, 참조무결성의 일반화된 개념이라 할 수 있다[2]. DTD에서 IDs를 추론할 수 있는 요소는 ID, IDREF, IDREFS 속성이 있다. 이는 관계형 스키마의 기본키와 외래키의 개념으로 보존이 될 수 있다. 하지만, DTD 고유의 문제, 즉 IDREF와 ID 속성이 하나의 DTD에 여러 개가 존재할 경우 어떤 IDREF가 ID를 참조하는지를 명확히 표현할 수 없다는 문제가 존재하는데, 본 논문에서는 이를 위한 해결책은 차후의 연구로 진행하고자 한다. 본 논문에서의 기법은 DTD 문서 내에 하나의 ID가 존재하고 이를 참조하는 IDREF, IDREFS를 가정한다.

4. 비교 분석

본 장에서는 그림 4의 DTD를 기반으로 NCPI-MDS 기법을 사용해서 생성되는 릴레이션과 기존의 Inlining 기법으로 생성되는 릴레이션을 비교 분석한다.

4.1 릴레이션 생성

그림 4의 DTD를 따르는 XML 문서를 저장하기 위해 기존의 Hybrid Inlining과 NCPI-MDS 기법에서 DTD 문서를 파싱하면 그림 5, 그림 6과 같은 DTD 그래프를 생성할 수 있다.

그림 5에서 점선 노드는 DTD에서 애트리뷰트를 의미하고, 노드와 노드 사이의 edge는 DTD에서의 "*", "+", "?" 연산자를 이용하여 표시하고, 연산자가 나타나지 않는 edge는 DTD에서 연산자 없이 정의된 엘리먼트, 즉 반드시 한번은 나타나야만 하는 엘리먼트를 연결하는 edge이다. 본 논문에서 제시하는 NCPI-MDS 기법에 의해 보존된 정보들을 DTD 그

표 1. 가능한 Cardinality Relationship

타입	대응수	연산자	제약조건
A	1 대 {0, 1}	?	NULL, UNIQUE
B	1 대 {1}		NOT NULL, UNIQUE
C	1 대 {0, ...}	*	NULL
D	1 대 {1, ...}	+	NOT NULL

```
<!ELEMENT root          (city*)>
<!ELEMENT city          (state?, restaurants, reviews*, name)>
<!ELEMENT state         (#PCDATA)>
<!ELEMENT reviews       (review*)>
<!ELEMENT review        (#PCDATA)>
<!ATTLIST review        rids    IDREF   #IMPLIED>
<!ELEMENT restaurants   (cuisine*)>
<!ELEMENT cuisine       (restaurant*)>
<!ATTLIST cuisine type  (French | Chinese | American | Korean)
                                #REQUIRED>
<!ELEMENT restaurant    ((appetizer | salad | desert | entree)*, name)>
<!ATTLIST restaurant    id      ID      #REQUIRED>
<!ELEMENT appetizer     (name, price)>
<!ELEMENT salad         (name, price)>
<!ELEMENT desert        (name, price)>
<!ELEMENT entree        (name)>
<!ATTLIST entree        spicy   CDATA   #IMPLIED>
```

그림 4. DTD 문서

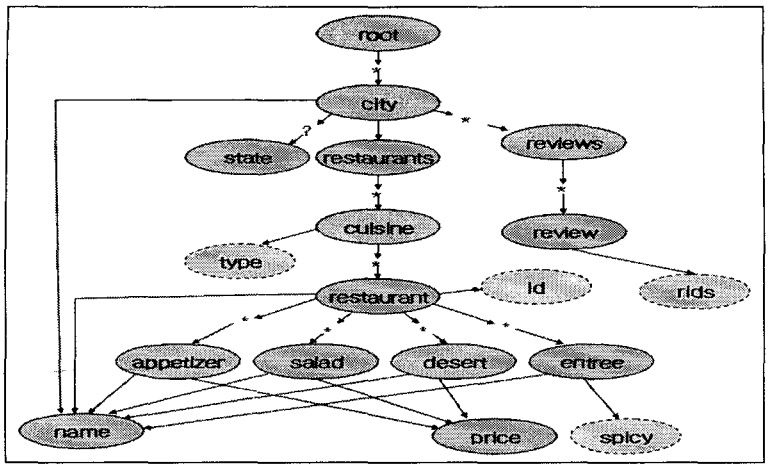


그림 5. DTD 그래프

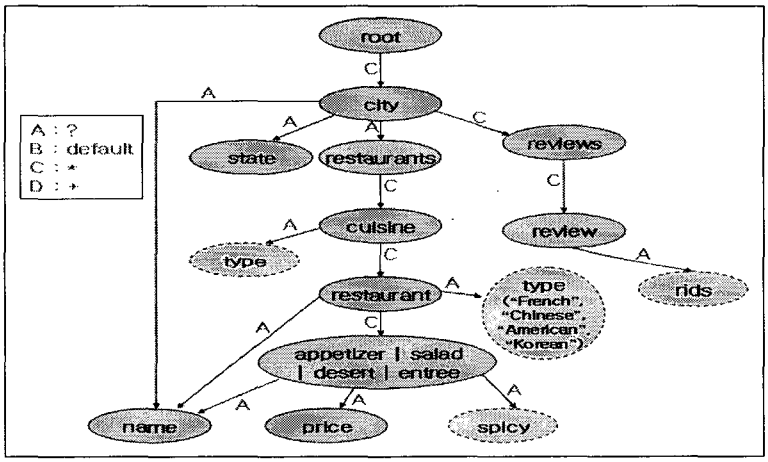


그림 6. Annotated DTD 그래프

래프에 표현하기 위해 그림 5에서 본 DTD 그래프를 사용하지 않고 연구 [2]에서 보여주고 있는 Annotated DTD 그래프(ADG)를 사용하여 표현한다. 또한, 연구 [2]의 원래의 ADG와는 달리 "|" 연산자로 정의된 엘리먼트는 그래프에서 하나의 노드로 표현하고, 도메인 제약조건을 가지는 노드는 해당 노드에 한정된 값을 표현한다. 이는 실제 구현 시 내부적인 자료구조로써 표현될 수 있다.

다음으로, 그림 5의 DTD 그래프를 기반으로 엘리먼트 그래프를 생성해 내고, 엘리먼트 그래프로부터 그림 7과 같은 릴레이션을 추론해 낼 수 있다.

또한, 그림 6의 DTD 그래프를 기반으로 4장에서 언급한 NCPI-MDS 기법에 따라 릴레이션을 추론해 내면 그림 8과 같은 릴레이션을 만들 수 있다.

4.2 정성적 비교 분석

기존의 Hybrid Inlining 기법으로 생성된 릴레이션은 그림 7에서 볼 수 있듯이 DTD에서 추론할 수 있는 NULL, NOT NULL, CHECK 절과 같은 제약조건을 보존하지 못하고 있기 때문에 XML 문서의 저장 시 데이터 무결성을 위해 저장 프로시져나 트리거를 이용해야 하는 번거로움이 생기게 된다. 또한, 그림 4의 DTD에서 appetizer, salad, desert, entree 엘리먼트는 "|", "*" 연산자를 이용하여 정의가 되어 있는데, 이는 결과적으로 그림 7에서 볼 수 있듯이 많은 릴레이션을 생성하게 되고, 조인 연산의 증가를 가져오므로 탐색비용을 증가시키게 된다.

이에 반해, 본 논문에서 제시하는 NCPI-MDS 기법은 DTD의 의미적인 정보를 관계형 스키마로 보존

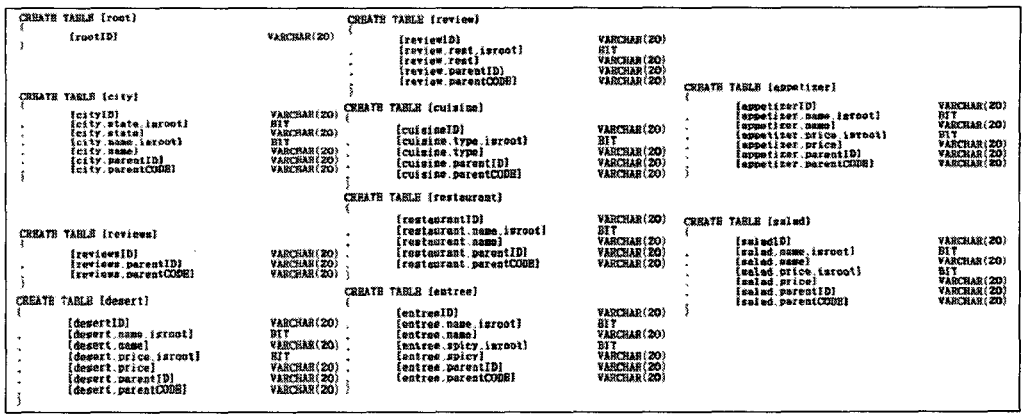


그림 7. Hybrid Inlining 기법으로 추론된 릴레이션

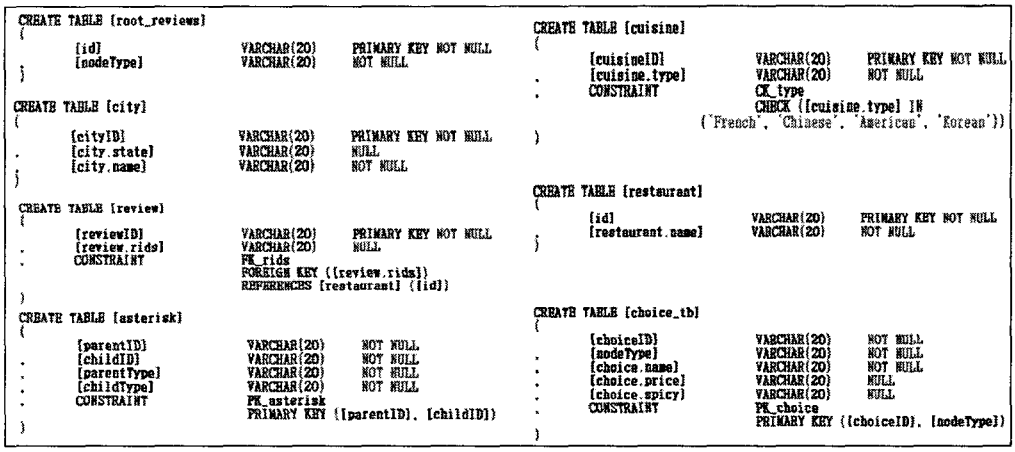


그림 8. NCPI-MDS 기법으로 추론된 릴레이션

함을 그림 8을 통해서 알 수 있다. 그림 8의 릴레이션에서 보존된 의미적인 정보와 효율적인 릴레이션 생성과 관련된 부분은 정리하면 표 2와 같다.

4.3 MS-SQL SERVER를 이용한 질의처리 성능 비교

그림 4의 DTD의 문서에 대한 질의 중 '레스토랑 이름이 restaurants-r1인 레스토랑의 모든 요리의 이

표 2. NCPI-MDS 기법의 결과

종류	예시
도메인 제약조건	cuisine 엘리먼트의 type 속성이 도메인 제약조건을 가짐. 이는 [cuisine] 릴레이션에서 볼 수 있듯이 SQL CHECK 절로 보존이 됨.
대응수 제약조건	DTD 그래프에서 대응수의 타입에 따라 릴레이션 생성 시 NULL, NOT NULL 제약조건으로 보존됨.
Inclusion Dependencies	review 엘리먼트의 rids 속성이 Inclusion Dependencies를 가짐. 이는 [review] 릴레이션에서 기본키, 외래키 제약조건으로 보존됨.
릴레이션 수 감소	[choice_tb] 릴레이션이 appetizer, salad, desert, entree 엘리먼트의 모든 데이터를 저장함. 기존의 기법에서 많은 릴레이션이 생성되던 것과는 달리 하나의 릴레이션에 데이터를 저장함으로써 탐색비용을 줄일 수 있음.
중복 데이터의 제거	[asterisk] 릴레이션을 통해 중복 데이터를 제거할 수 있고, 상향·하향 탐색에 도움을 줄 수 있음.

```
SELECT [appetizer.name], [appetizer.price], null FROM [appetizer]
        WHERE [appetizer.parentID] = (SELECT [restaurantID] FROM [restaurant]
                                      WHERE [restaurant.name] = 'restaurant-r1')
UNION
SELECT [desert.name], [desert.price], null FROM [desert]
        WHERE [desert.parentID] = (SELECT [restaurantID] FROM [restaurant]
                                   WHERE [restaurant.name] = 'restaurant-r1')
UNION
SELECT [salad.name], [salad.price],null FROM [salad]
        WHERE [salad.parentID] = (SELECT [restaurantID] FROM [restaurant]
                                  WHERE [restaurant.name] = 'restaurant-r1')
UNION
SELECT [entree.name], null, [entree.spicy] FROM [entree]
        WHERE [entree.parentID] = (SELECT [restaurantID] FROM [restaurant]
                                   WHERE [restaurant.name] = 'restaurant-r1')
GO
```

그림 9. Hybrid Inlining 기법을 위한 SQL syntax

```
SELECT [choice.name], [choice.price], [choice.spicy]
        FROM [NCPI-choice_tb]
        WHERE [choiceID] IN
                (SELECT [childID] FROM [NCPI-asterisk]
                        WHERE [parentID] = (SELECT [id] FROM [NCPI-restaurant]
                                WHERE [restaurant.name]='restaurant-r1'))
GO
```

그림 10. NCPI-MDS 기법을 위한 SQL syntax

표 3. MS-SQL 실행 계획 비교

	JOIN 수	Table Scan 수	Index Scan 수	Index Seek 수
Hybrid Inlining	4	8	0	0
NCPI-MDS	2	0	1	2

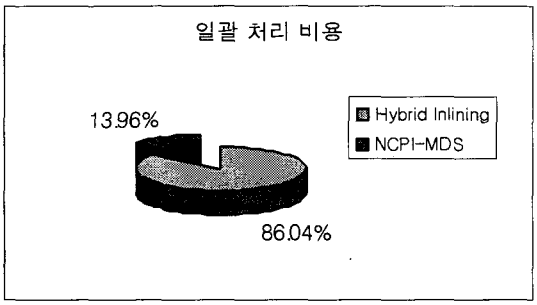


그림 11. 질의어 일괄 처리 비용

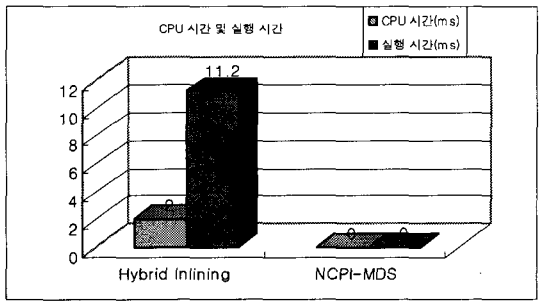


그림 12. CPU 시간 및 실행 시간

름과 가격, spicy는?'이라는 질의를 고려해 볼 경우, 이는 SQL 문장으로 그림 9, 그림 10과 같이 변환될 수 있다.

이를 MS-SQL SERVER를 이용하여 실제 XML 문서의 데이터를 로드하고, 위의 쿼리문을 실행하였을 경우를 비교하면 표 3, 그림 11, 그림 12와 같다.

5. 결 론

XML 문서를 RDBMS에 저장하기 위한 기법 중 Hybrid Inlining 기법은 DTD로부터 보존이 될 수 있는 많은 의미적인 정보들을 손실할 뿐만 아니라, 실제 XML 문서를 저장할 때 많은 데이터의 중복이 발생하는 문제점을 보인다.

이에 본 논문에서는 DTD로부터 추론할 수 있는 의미적인 정보들을 보존하기 위해 수정된 DTD 간소화 절차를 제시하고, 이를 관계형 릴레이션으로 매핑할 수 있는 방법을 제시하였으며, 데이터의 중복을 방지하고, 데이터 질의 시 탐색에 도움이 될 수 있는 효율적 릴레이션 생성을 다룬 NCPI-MDS 기법을 제시하였다.

향후 연구 방향은 본 논문에서 제시한 NCPI-

MDS 기법에 의해 생성된 릴레이션에 대해 더 많은 종류의 XML 문서를 고려한 질의처리 능력을 검증하는 과정이 이루어져야 할 것이다. 또한, 릴레이션을 추론하기 위해 DTD를 사용하는 것이 아니라 DTD의 문제점을 극복하기 위해 제시된 XML Schema를 사용하는 릴레이션 추론기법도 연구되어야 한다.

참 고 문 헌

[1] S. Lu, Y. Sun, M. Atay, and F. Fotouhi, "A New Inlining Algorithm for Mapping XML DTDs to Relational Schemas," *Proc. of the 1st International Workshop on XML Schema and Data Management*, 2003.

[2] D. Lee, and W. W. Chu, "Constraints-Preserving Transformation from XML Document Type Definition to Relational Schema," *International Conference on Conceptual Modeling / the Entity Relationship Approach*, 2000.

[3] J. Shanmugasundaram et al. "Relational databases for querying XML documents: Limitations and opportunities," *In Proc. of VLDB*, Edinburgh, Scotland, 1999.

[4] S. A. T. Lahiri and J. Widom, "Ozone : Integrating Structured and Semistructured Data," *In proceedings of DBPL Conf.*, 1999.

[5] Y. Chen, S. Davidson, and Y. Zheng, "Constraint Preserving XML Storage in Relations," *In WebDB*, 2002.

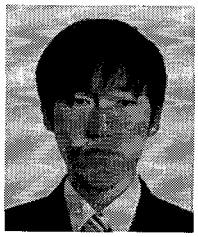
[6] H. Schoning, "Tamino - a DBMS Designed for XML," *In Proceedings of IEEE ICDE*, 2001.

[7] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)," *W3C Recommendation*, http://www.w3.org/TR/2004/REC-xml-20040204/, 2004.

[8] J. Clark and S. DeRose, "XML Path Language (XPath) Version 1.0," *W3C Recommendation*, http://www.w3.org/TR/xpath, 2004.

[9] S. Boag, D. Chamberlin, M. F. Fernandez, D. Florescu, J. Robie, and J. Simeon, "XQuery 1.0 : An XML Query Language," *W3C Working Draft*, http://www.w3.org/TR/query/, 2005.

[10] D. Florescu and D. Kossmann, "Storing and Queryring XML Data using an RDBMS," *Bulletin of the IEEE Computer Society Technical Commitee on Data Engineering*, 22(3):27-34, 1999.



안 성 철

2004년 한성대학교 산업공학과 졸업(B.S.)
2006년 한성대학교 컴퓨터공학과 졸업(M.S.)
2004년~현재 한성대학교 컴퓨터공학과 조교
관심분야 : 객체지향 데이터베이스, XML 데이터베이스, 정보보안, DB보안

김 영 웅

1981년 경북대학교 전자공학과 졸업(B.S.)
1984년 한국과학기술원 전산학과 졸업(M.S.)
1993년 한국과학기술원 전산학과 졸업(Ph.D.)
1984년~1997년 한국통신 연구개발원 팀장
1997년~현재 한성대학교 컴퓨터공학과 교수
관심분야 : 객체지향 데이터베이스, XML 데이터베이스, 공간 데이터베이스, 멀티미디어 정보검색