Inlining 알고리즘을 이용한 XML DTD 매칭 방법에 관한 연구

허보진*, 김형석*, 김창석*
*공주대학교 대학원 멀티미디어학과
e-mail:{bobe04, replay76, csk}@kongju.ac.kr

A Study for XML DTD Matching Method using Inlining Algorithm

Bo-Jin Heo*, Hyeong-Seok Kim*, Chang Suk Kim*
*Dept of Multimedia, Kongju National University

요 약

XML DTD 매칭은 데이터 통합이나 데이터 웨어하우스, 웹 마이닝, 전자상거래, 의미적 질의 처리등과 같은 데이터베이스 관련 응용분야에서 수행해야 할 근본적인 연구 분야이다. 웹이 발전됨에 따라 웹 상의 데이터 교환의 표준인 XML로 많은 데이터를 표현하게 되었고, 이 XML DTD에 대한 매칭이 주된 연구 분야로 대두되었다. XML 스키마는 플랫 구조인 기존의 관계형 데이터베이스 스키마와는 달리 계층적인 트리 구조로 이루어져 DTD를 직접 비교하기가 어렵다. 본 논문에서는 계층적 구조인 XML DTD의 계층적 구조 정보와 무결성 제약조건을 추출하여 일차원적인 직렬 구조로 변환한 후, 유사한 DTD를 매칭하는 방법을 제안한다.

1. 서론

XML DTD 매칭은 데이터 통합이나 데이터 웨어하우스, 웹마이닝, 전자상거래, 의미적 질의처리 등과 같은 데이터베이스 관련 응용 분야에서 수행해야 할 근본적인 연구 분야이다. 웹이 발전됨에 따라 웹상의 데이터 교환의 표준인 XML로 많은 데이터를 표현하게 되었고, 이 XML DTD에 대한 매칭이 주된 연구 분야로 대두되었다.

XML DTD는 플랫 구조인 기존의 관계형 데이터베이스 스키마와는 달리 계층적이며 집합과 회귀적인 성질을 가진 트리 구조로 이루어져 스키마를 직접 비교하기가 어렵다.

따라서, 본 논문에서는 직렬식(Inline) 알고리즘을 이용하여 이것을 일차원적인 구조로 변환한다. 이 변환된 구조는 DTD의 구조적인 정보와 의미적인 제약조건 등이 포함된 일련의 연속된 값의 리스트인 특징 값으로 표현된다. 이 특징 값 사이의 유사도를 측정함으로서 두 DTD 간의 스키마 매칭 정도를 정량화된 값으로 표현하는 방법을 제안한다.

2. 관련 연구

2.1 스키마 매칭 접근 방법들

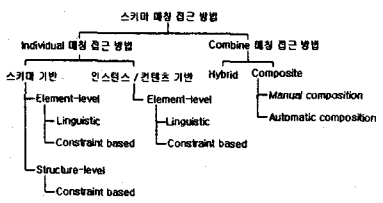

그림 1. 스키마 매칭 접근 방법 분류

위의 분류는 스키마 매칭에 대한 분류이다. 스키마 구조만 고려했느냐 혹은 저장된 데이터도 고려했느냐를 구분하며, 어휘의 유사도와 스키마의 제약사항을 했는지 여부에 따라 구분하여 분류한다[2].

2.2 XML DTD의 특성

DTD(Document Type Definition)는 XML 문서의 구조와 문법과 어휘(tag이름, attribute이름 등)에 대한 정의이다. DTD는 문서 형식 선언(Document Type Declaration), 요소 선언(Element Declarations), 속성 선언(Attribute Declarations), 노테이션 선언(Notation Declarations), 엔터티 선언(Entity Declarations) 등과 같이 다섯 부분으로 나누어 볼 수 있다.

문서형식선언(DOCTYPE 선언)은 DTD를 사용하겠다는 것을 선언한 다음, DTD의 정의들을 찾을 수 있는 위치를 알려주는 역할을 한다.

어휘집과 비교하여 문서들의 유효성을 검증할 수 있어야 한다는 것은 마크업 언어들의 공통의 요구사항이다. 어떤 하나의 XML 문서는 그 안에 들어 있는 XML 내용이 미리 정의된 허용 가능한 요소, 속성, 그리고 그 밖의 것들을 따르고 있는 경우에 유효한(valid) 것이 된다.

특별한 문서 형식 정의(Document Type Definition)문법, 즉 DTD를 이용함으로써 특정 파서들을 통해 한 종류의 문서 형식에 일치하는 내용인지를 검사할 수 있게 된다. XML 권고안은 파서들을 두 종류로 나누고 있는데, 하나는 유효성 검증(validating) 파서이고 다른 하나는 비유효성 검증(nonvalidating) 파서이다. 권고안에 따르면 유효

성 검증 파서들은 반드시 DTD를 사용하여 유효성 검증을 할 수 있도록 구현되어야만 한다. 그러므로 유효성 검증 파서만 있으면 애플리케이션 안에 앞에서 본 것과 같은 유효성 검증을 위한 코드를 작성하지 않고도 외부 프로세서를 이용하여 유효성 검증을 하는 것이 가능하게 된다.

2.3 CPI Algorithm

CPI(Constraints-preserving Inlining) 알고리즘은 DTD에서 발견할 수 있는 의미적인 제약조건들을 추출하여 관계형 스키마로 변환할 때 이 제약조건들을 반영하는 것이다. Cardinality Constraints와 Inclusion Dependencies (INDs), Equality-Generating Dependencies(EGDs), Tuple-Generating Dependencies(TGDs) 제약조건들이 있다.

(1) ADG(Annotated DTD Graph)로부터 top node를 판정한다.

(2) Annotated DTD 그래프에서 관계형 스키마를 생성한다.

(3) 출력 스키마는 모든 스키마 테이블과 의미적인 제약조건들을 결합한다.

3. 전체적인 구조

```
<!DOCTYPE publication[
   <!ELEMENT paper (title, contact?, author, cite?)>
   <!ATTLIST paper id ID #REQUIRED>
   <!ELEMENT title (#PCDATA)>
   <!ELEMENT contact EMPTY>
   <!ATTLIST contact aid IDREF #REQUIRED>
   <!ELEMENT author (person+)>
   <!ATTLIST author id ID #REQUIRED>
   <!ELEMENT person (name, (email|phone)?>
   <!ATTLIST person id ID #REQUIRED>
   <!ELEMENT name EMPTY>
   <!ATTLIST name fn CDATA #IMPLIED
                  ln CDATA #REQUIRED>
   <!ELEMENT email (#PCDATA)>
   <!ELEMENT phone (#PCDATA)>
   <!ELEMENT cite (paper*)>
   <!ATTLIST cite pid ID #REQUIRED
                  format (ACM|IEEE) #IMPLIED>
]>
```

그림 2. publication에 대한 DTD

```
<paper id= "TR-2003-009" >
    <title>XML Schema Matching</title>
    <contact>aid= "철수" </contact>
    <author>
        <person id= "철수" >
            <name fn= "철수" ln= "박" />
            <email>cspark@hanmail.net</email>
            </person>
    </author>
</paper>
```

그림 3. XML 문서

그림 4의 주석 DTD 그래프를 직렬식 알고리즘(Inline algorithm)[4,5]을 이용하여 플랫한 구조로 바꾸면 다음 표 1과 같다.

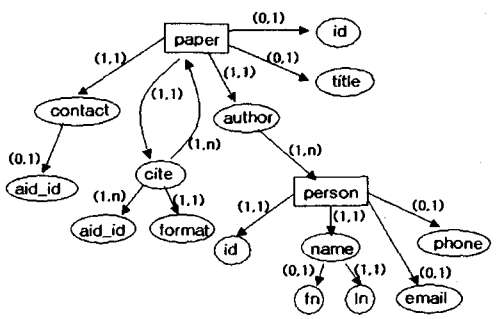

그림 4. Publication에 대한 주석 DTD 그래프

표 1에서 애트리뷰트는 주석 DTD 그래프에서 엘레먼트 paper에서 각 노드를 방문하면서 추출한 것 이며, 계층적인 성질을 플랫하게 만든다고 하여 인라인 알고리즘이라고 부른다.

4. Inline 알고리즘을 이용한 XML DTD 변환

4.1 무결성 제약조건의 도출

◉ 도메인 제약조건의 도출

DTD의 도메인 무결성 제약조건은 다음과 같은 경우이다.

<!ATTLIST author gender (male | female)

표 1. paper 특징 값의 추출

속성	P.Key	F.Key	자료형태	길이	Nullable	특징값
id	yes	no	numeric	4	no	(1,0,0,0.2,0)
title	no	no	string	20	no	(01,0,1,0.7,0)
person_id	yes	yes	numeric	4	yes	(1,1,0,0.2,0)
person_fn	no	no	string	10	yes	(0,0,0,0.4,0)
person_ln	no	no	string	10	no	(0,0,0,0.4,0)
person_email	no	no	srting	20	yes	(1,0,0,0.2,0)
person_phone	no	no	numeric	20	yes	(0,0,1,0.7,0)
citeaid	no	no	string	20	yes	(0,0,1,0.7,0)
dcontact_aid	no	yes	string	20	yes	(0,0,1,0.7,0)

#REQUIRED>

키워드 #REQUIRED는 관계형 모델에서 NOT NULL과 같이 NULL을 허용하지 않는 개념이다. 그래서 이런 요소에서는 NOT NULL 조건을 추출할 수 있다.

◉ 카디널리티 제약조건의 도출

XML 스키마인 DTD는 계층적이며, 집합(set)과 회귀적인(recursive) 성질을 가지고 있다. 그러므로 '?', '*', '+' 등의 카디널리티가 표현되어 있으면 비교가 매우 어렵기 때문에 이것을 반영한 그래프 형태로 DTD를 변환하는 것이 필요하다.

일반적인 DTD 그래프에 '?', '*', '+' 들로 표현되는 카디널리티 관계(cardinality relationship)를 부과하여 주석 DTD를 생성할 수 있다. 이것에 대한 표기법은 Lee[4]의 알고리즘에 따른다.

카디널리티 관계는 4가지로 구분할 수 있으며, 각각의 의미성은 다음과 같다.

1-to-{1} mapping("only"):NOT NULL (A Type)
1-to-{0,1} mapping("at most")
:NOT NULL (B Type)
1-to-{1,…} mapping("at least")
:NOT NULL (C Type)
1-to-{0,…} mapping("any"):NULLable (D Type)

◉ 외래 키 제약조건의 도출

DTD의 속성이 다음과 같은 경우 외래 키를 추출할 수 있다.

<!ELEMENT contact EMPTY>
<!ATTLIST contact aid IDREF #REQUIRED>

키워드 IDREF는 관계형 모델에서 참조 무결성과 같은 개념이다. 그래서 요소 contact에서 외래 키 개념을 추출할 수 있다.

4.2 Inlining Algorithm을 이용해 XML DTD를 Flat한 구조로 변환

위의 Inline 알고리즘과 무결성 제약조건 도출 알고리즘을 이용하여 애트리뷰트의 특징 구분자(feature discriminator)는 주 키(P.key), 외래 키(F.key), 자료형태(Data type), 길이(Length), 널 허용 여부(Nullable) 등으로 구성된다. 이들은 주석 DTD 그래프에서 플랫 구조로 변환될 때 제약조건에서 추출된 결과이다. 이때 길이는 XML 문서의 데이터 길이를 의미하며 여기서는 임의의 값을 할당 했다.

특징 구분자의 값들은 비교를 용이하게 하기 위해 정규화된 값으로 바꾸는데, 이것을 특징 값(feature value)이라고 한다. 특징 값은 0과 1사이의 값을 가지는데 [0,1]로 표현한다.

주 키 특징이 있으면 1, 없으면 0을 부여한다. 마찬가지로 외래 키 특징이 있으면 1, 없으면 0을 부여한다. 자료형태는 문자형이면 1을, 숫자형이면 0을 부여한다. 자료의 길이는 XML 문서에 있는 데이터를 참조하여 평균 길이

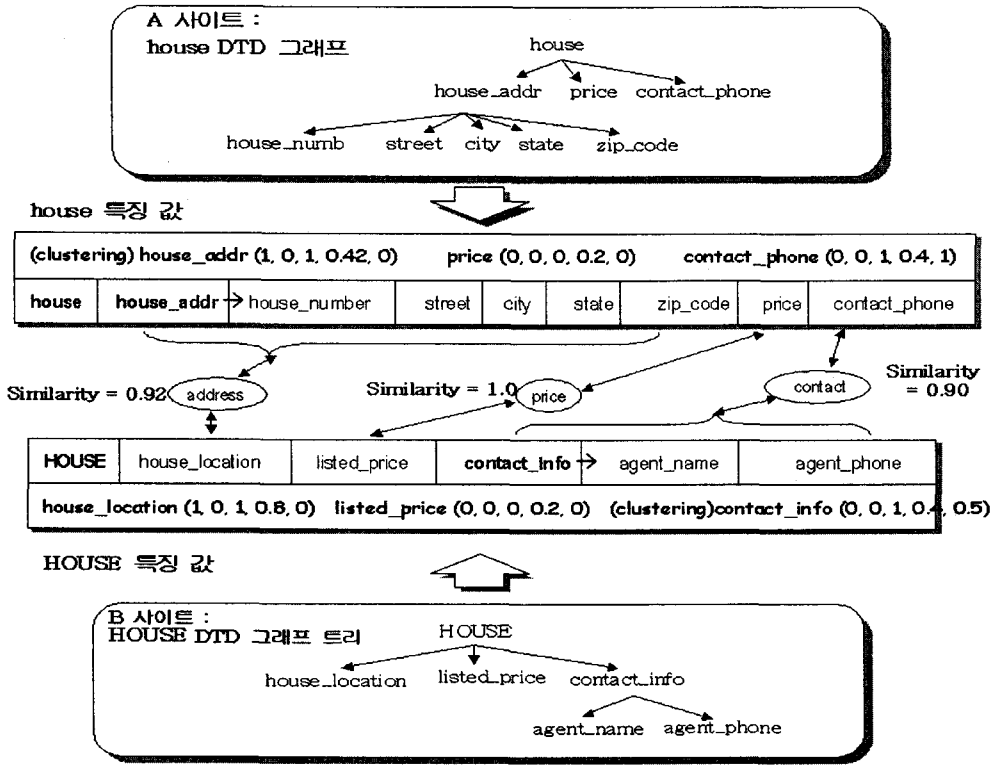

그림 5. 두 스키마 사이의 특징 값 비교 과정

를 구하여 정규화한 값을 표현한다.

5. Matching 방법을 이용한 비교 분석

어떤 애트리뷰트의 특징 값은 다른 애트리뷰트와 매칭하기 위한 속성을 모두 가지고 있다. 그림 5는 웹 상에서 두 사이트에 있는 **house**와 **HOUSE**라는 DTD간의 매칭 유사도를 구하는 과정을 나타낸 것이다.

◉ 알고리즘: DTD Matching

Step 1: 주어진 DTD를 직렬식 알고리즘을 이용하여 계층적인 구조를 직렬 구조로 바꾼다. 이때 각 애트리뷰트는 5가지 특징 구분자에 대한 값을 구한다.

Step 2: 특징 구분자에 대한 특징 값을 구한다. 이때 정규화 함수는 선형 함수를 사용한다.

Step 3: 애트리뷰트 중에 매치 카디널리티 1:n 인 경우, 클러스터링 특징 값을 구한다.

Step 4: 각 특징 값을 Sim 연산자를 이용하여 애트리뷰트 간의 유사도를 구한다.

Step 5: 각 애트리뷰트 간의 유사도를 산술 평균하여 최종적인 유사도를 구한다

A 사이트의 어떤 DTD **house**와 B 사이트의 임의 DTD **HOUSE**는 의미적으로 비슷하지만, 구조는 상이하다. 이것을 DTD 그래프를 직접 비교하는 것은 어려움이 많다.

여기서는 이 DTD 그래프를 3장에서 서술한 방법대로 계층적인 성질을 플랫하게 만드는 직렬식 알고리즘을 이용하여 플랫하게 만든다. 이 과정의 결과물로 특징 값을 가지는 애트리뷰트가 생성된다. 이것은 이차원적인 계층구조를 일차원적인 평면구조로 바꾸어 직접 비교가 가능하다.

애트리뷰트의 종류는 단순 타입(simple type)과 클러스터링 타입(clustering type)으로 나눌 수 있다.

단순 타입은 price, listed_price, house_location 처럼 단일 애트리뷰트를 가진다. 클러스터링 타입은house_addr, contact_info와 같이 자식 애트리뷰트를 가졌던 것이다. 이것은 Sim 연산자를 통하여 클러스터링된 특징 값을 구해야 한다[6, 7].

$$SIM(A,b) = \frac{1 - \sum |feature\ value(M)\ feature\ value(S)|}{|X|}$$

SIM 연산자는 두 특징 값 사이의 유사도를 구하는 연산자이다. 여기서는 house_addr와 house_location 사이의 유사도가 0.92임을 보여 준다. 또한 contact_phone과 contact_info의 유사도는 0.90이다. 이것들은 완전히 일치하지는 않지만 의미적으로 거의 일치함을 나타낸다. 유사도가 1.0인 price와 listed_price는 완전히 같은 의미의 애트리뷰트임을 나타낸다.

각 애트리뷰트의 유사도가 나오면 최종적으로 A 사이트의 DTD **house**와 B 사이트의 DTD **HOUSE**의 매칭 유사도를 구한다. 여기서는 다양한 방법 중에 산술평균 식을 이용한다. (0.92 + 0.90 + 1.0) / 3 = 0.94가 두 DTD간의 의미적 유사도이다.

6. 결론 및 향후 연구

본 논문에서는 계층적인 형태의 XML DTD들 간의 스키마 매칭 방법을 제안하였다. 계층적이고 내포적인 구조인 XML DTD를 직접적으로 비교한다는 것은 자동화하기가 어려워 직렬식 알고리즘을 이용하여 이것을 일차원적인 구조로 변환하였다. 이 변환된 구조는 DTD의 구조적인 정보와 의미적인 제약조건 등이 포함된 일련의 연속된 값의 리스트인 특징 값으로 표현하였다. 이 특징 값 사이의 유사도를 측정함으로서 두 DTD 간의 스키마 매칭 정도를 정량화된 값으로 표현하였다.

본 연구의 의의는 복합적(complex)이고 계층적인 스키마를 일차원적인 리스트 형태의 수치로 단순화하여 스키마 매칭을 자동화한 것이라고 할 수 있다.

7. 감사의 글

본 연구는 한국과학재단 목적 기초연구(R01-2002-000-00068-0)의 지원으로 수행되었음.

참고문헌

[1] W.Li and C.Clifton, "SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks", Data & Knowlwdge Engineering 33, pp.49-82, 2000.

[2] E.Rahm and P.Bernstein, "A survey of approaches to automatic schema matching", The VLDB Journal, (10) pp.334-350, 2001.

[3] A. Doan, P.Domingos and A.Levy, "learning Source Descriprions for Data Integration", Proc.on WebDB 2000, pp.81-86, 2000.

[4] D.Lee and W.Chu, "Constraints-preserving Transformation from XML Document Type Definition to Relational Schema", UCLA-CS-TR-200001, 2001.

[5]J.Shanmugawundaram, K.Tufte, G.He, C.Zhang, D.Dewitt and J.Naughton, "Relational Databases for Querying XML Documents: Limitations and Opportunities," Proc. on VLDB, Edinburgh, Scotland, pp. 302-314, 1999.

[6] C.S.Kim, "Systematic Generation Method and Efficient Representation of Proximity Telations for Fuzzy Relational Database Systems," Proc. of the 20th EUROMICRO Conference, Licerpool, England, IEEE Computer Society Press, pp. 549-555, 1994.

[7] C.S.Kim, "A Complex Matching of XML Schema", Proc. of the 20th International Conferenceon Internet Computing, Las Vegas, Nevada, U.S.A., CSREA Press, pp.484-489, 2002.