통합 XML 스키마의 효율적인 추출

임태우, 강혜란°, 이경호

연세대학교 컴퓨터과학과

twrhim@icl.yonsei.ac.kr, hrkang@icl.yonsei.ac.kr°, khlee@cs.yonsei.ac.kr

An Efficient Extraction of An Integrated XML Schema

Taewoo Rhim, Haeran Kang°, Kyong-Ho Lee

Computer Science Department, Yonsei University

요 약

XML 스키마의 수가 급증함에 따라 동일한 도메인에 속하는 유사한 스키마를 통합하는 방법에 대한 관심이 증가하고 있다. 일반적으로 XML 스키마 통합 과정은 스키마 클러스터링과 통합 스키마 추출의 두 단계로 구성된다. 본 논문에서는 통합 스키마의 추출을 위한 효율적인 방법을 제안한다. 제안된 방법은 공통 구조 추출, 스키마 통합, 그리고 최적화의 세 단계로 이루어진다. 실험결과, 제안된 방법은 처리시간 및 정확도 측면에서 우수한 결과를 보였다.

1. 서 론

XML(eXtensible Markup Language)은 구조 정보를 표현할 수 있으며 플랫폼에 독립적이라는 특징 때문에 인터넷을 비롯한 다양한 분야에서 정보 표현 및 교환을 위한 표준으로 널리 사용되고 있다.

인터넷 상에 분포되어 있는 XML 문서를 검색하기 위해서는 XML 스키마를 통해 문서의 구조와 의미를 파악하여야 한다. 그런데 웹상에는 수많은 XML 스키마가 존재할 뿐만 아니라 계속하여 새로 생성되고 있다. 이에 유사한 스키마를 통합하는 스키마 통합(schema integration)에 대한 관심이 증가하고 있다. 일반적으로 스키마 통합 과정은 <그림 1>과 같이 스키마 클러스터링과 통합 스키마 추출의 두 단계로 구성된다 [1].

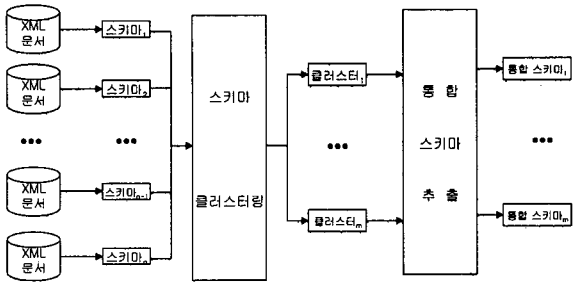


그림 1. 스키마 통합 과정.

본 논문에서는 통합 스키마의 추출을 위한 효율적인 방법을 제안한다. 한편, 기존에 통합 스키마 추출을 위한 여러 방법이 제안되었다 [2][3][4][5][6][7][8][9][10]. 스키마 통합을 위한 기존 연구의 대부분은 노드간의 유사도를 비교하여 공통점이 있는 두 노드 사이의 충돌을 해결하고 통합 노드를 만드는데 초점이 맞추어져 있었다. 이러한 방법은 각 노드들 간의 관계를 고려하지 않음으로서 생성된 통합 스키마가 기존 스키마의 노드들 간의 관계가 의미하던 부분을 완전하게 반영할 수 없으며 이로 인해 통합 스키마의 효율성이 저하될 수 있다. 또한 사용자의 개입이 충돌 해결 시에나 최적화 단계에서 필요한 경우가 많아 많은 시간이 소요된다.

제안된 방법은 공통 구조 추출, 스키마 트리 통합, 그리고 최적화의 세단계로 구성된다. 먼저 대응되는 경로간 통합을 통해 스키마 사이의 공통구조를 결정하고 이에 기반하여 유사 개념들을 통합한다. 노드간 이름, 데이터 타입, 빈도 지시자, 구조상의 차이를 공통 구조에 반영하고 최적화 단계를 거쳐 통합 스키마를 생성한다. 매칭 과정에서 정확성을 높이기 위해 축약어 사전과 동의어 사전을 적용한다. 최적화 단계에서는 중복되는 노드를 제거하여 최종적으로 통합 스키마를 생성한다.

실험 결과, 제안된 알고리즘의 장점은 경로간 통합 과정을 통해 스키마 통합의 시간 복잡도를 향상시키고 사용자의 개입 없이 통합 과정을 진행한다는 것이다.

2. 관련 연구

본 절에서는 XML 스키마 통합에 관한 기존의 연구 결과를 간략히 기술한다. Cruz 등 [2]은 XML 스키마를 RDF 온톨로지 형태로 모델링한 후, 각 온톨로지로부터 클래스 및 관계들간의 유사도에 기반하여 전역 온톨로지 생성한다[11]. Lehti와 Fankhauser [3]는 XML 스키마를 OWL(Web Ontology Language)형태의 온톨로지로 변환 및 통합하여 통합 스키마를 생성하는 방법을 제안한다. 제안된 방법은 OWL이 제공하는 개념과 관계 사이의 시맨틱 매핑을 정의하는 메커니즘에 기반한다. Jeong과 Hsu [4]는 클러스터링된 DTD(Document Type Definition)로부터 트리 문법 추론 기술을 기반으로 하여 통합 뷰를 추출하는 방법을 제안한다. 내재된 문법(nested grammar)을 추출하고 유사한 상태 및 관계를 통합 및 재구성하여 최적화된 통합 스키마를 생성한다. Lee 등 [5]은 XML 스키마의 복잡한 충돌 문제를 분류하고, XQuerySD를 이용한 엘리먼트 및 속성간 충돌 해결 방법을 제안한다. Mello 등 [6]이 제안한 방법은 DTD를 개념적으로 추상화한 객체 지향 기반의 표준적인 스키마를 입력으로 받는다. 객체지향 개념모델로 표현한 후 어휘적으로 유사한 개념들 사이의 충돌을 해결하여 통합 개념모델을 생성한다. Castano 등 [7]은 XML DTD를 제안된 X-class 형태로 변환하고, X-class간의 의미적 매핑을 기반으로 구조적인 클러스터링을 적용하여 사용자의 개입을 통해 유사한 클래스들 간의 충돌을 해결하고 노드 및 속성 간 차이를 조절하여 통합 클래스를 생성한다. Passi 등 [8]은 XML 스키마를 XSDM(XML Schema Data Model)이라는 객체 지향 데이터 모델로 표현한 후, 엘리먼트와 속성을 통합한다. XSDM으로 표현된 통합 스키

※이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2004-041-D00613)

마는 다시 XML 스키마로 전환된다. Zhang과 Liu [9]는 XML 스키마를 EUML(Extended Unified Modeling Language) 다이어그램으로 변환시킨 후 두 다이어그램 사이의 충돌을 해결하고 재구성 단계을 거쳐 최종적인 통합 개념모델을 생성하는 방법을 제안한다. Hakimpour과 Geppert [10]등은 각 로컬 온톨로지의 개념사이의 유사 관계를 바탕으로 온톨로지를 통합하고 온톨로지와 스키마 사이의 매칭관계를 이용하여 통합 스키마를 생성한다.

3. 제안된 방법

제안된 알고리즘은 <그림 2>와 같이 공통구조 추출, 스키마 트리 통합, 최적화의 세 단계로 구성된다.

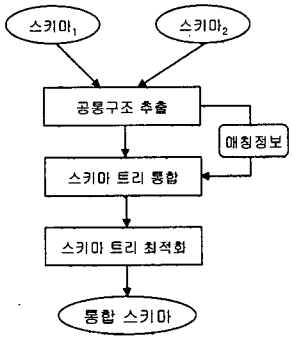


그림 2. 스키마 통합 과정.

3.1 공통구조 추출

본 절에서는 각 스키마의 경로를 추출하고 경로 구성 노드들 간의 매칭을 통해 스키마 사이의 공통 구조를 계산한다. 먼저 두 트리의 뿌리노드로부터 단말 노드까지의 모든 경로를 추출한다. 또한 서로 대응하는 경로들 간의 LCS(Longest Common Subsequence)를 계산한다. LCS는 두 대응 경로 사이의 최장 공통경로를 의미하며 각 경로의 노드 사이의 일대일 매칭을 적용하여 추출한다.

또한 두 스키마간의 공통구조를 계산하기 위해서 공통 경로의 합이 최대에 해당하는 경로간의 일대일 매칭을 찾는다. 이때 계산된 공통 경로의 합을 GCS(Greatest Common Subset)라고 정의한다. GCS를 계산하기 위해 경로 및 공통 경로를 각각 정점 및 간선의 가중치로 모델링하여 n개의 소스 경로와 m개의 타겟 경로로 구성된 가중치 이분 그래프(weighted bipartite graph) Kn,m을 생성하고, Kn,m에서 최대 이분 매칭(maximal bipartite matching)을 찾는다. 스키마간 공통 구조는 대응 경로간 충돌을 해결한 공통 경로의 합을 트리 형태로 나타낸 것이다.

매핑 테이블은 두 스키마와 통합 스키마 사이의 매칭 정보를 표현한다. 각 스키마에 대한 접근은 통합 스키마와 매핑 테이블을 이용하여 가능하다. 소스 스키마의 노드 n_i와 타겟 스키마의 노드 n_j가 매칭 관계를 형성하며 통합 스키마의 노드 n에 대응할 때 (n_i, n_j, n) 형태의 매칭 정보가 생성된다. 생성된 매칭 정보는 테이블의 형태로 저장되며 통합 및 최적화 단계를 거치면서 수정되어 최종적인 매핑 테이블을 생성한다.

3.2 스키마 트리 통합

본 절에서는 공통 구조 추출 단계에서 계산된 GCS 및 매칭 관계를 이용하여 스키마 트리를 통합한다. 소스 및 타겟 스키마의 대응경로간 통합을 통해 공통구조를 결정한다. 또한 생성된 공통구조에 기반하여 유사개념을 통합한다. 공통구조의 뿌리 노드로부터 트리를 너비우선 탐색하면서 소스 및 타겟 스키마의 차이를 공통구조에 반영한다. 소스 및 타겟 스키마간의 대응 경로는 매칭 관계를 형성하는 노드들로 구성된 공통 경로 외에 구조적인 차이를 포함할 수 있다.

중간노드가 존재하지 않을 경우, 공통 경로는 변하지 않으며, 한 경로만 중간노드를 포함할 경우 중간노드를 포함하는 형태로 공통 경로가 변환된다. 두 경로 모두 중간노드가 존재할 경우, 중간 노드간의 관계를 계산한다.

한 노드명이 다른 노드명을 대표하는 명사이거나 두 노드명이 부분-전체 관계를 갖을 경우, 두 노드간 관계는 포함관계로 정의된다. 또한 두 노드명 사이에 매핑관계 혹은 포함 관계가 존재하지 않을 경우, 두 노드는 비유사관계로 정의된다.

중간 노드 간에 포함관계가 존재할 경우, 두 노드는 상위개념과 하위개념으로서 부모자식 관계를 형성하며, 비유사관계의 두 중간노드는 각각이 중간노드로 존재하며 2개의 경로를 만든다. 이렇게 변환된 공통 경로들의 합이 스키마 사이의 공통 구조로 정의된다.

노드 사이의 충돌 해결은 Mello 등 [6]의 방법을 사용하였다. 노드간의 대응은 아래와 같이 나뉘어 지며 각각의 통합 방법은 다음과 같다.

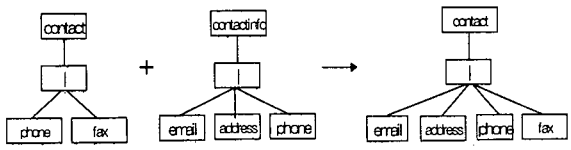
Case 1 : 심플 타입(simple type) 노드 사이의 통합

심플 타입 노드는 스키마 트리에서 단말 노드에 해당하며 엘리먼트나 속성을 나타낸다. 엘리먼트와 속성 사이의 통합일 경우, 엘리먼트로 통합 개념을 형성한다.

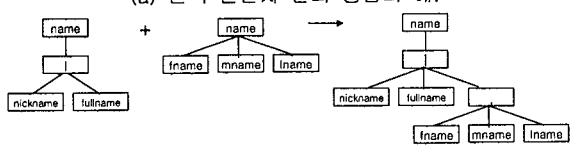
- 이름 변환 규칙 : WordNet [12]을 이용하여 노드간 관계를 파악하고 광의의 레이블을 선택한다.
- 데이터 타입 변환 규칙 : 두 타입 중 정보 손실 없이 변환을 통해 다른 타입을 포함할 수 있는 타입을 선택한다. 변환이 불가능할 경우, 데이터 타입으로서 문자열(string)을 갖는다.
- 빈도수 속성 변환 규칙 : 두 빈도수 쌍 중 최소값(minOccurs) 중의 작은 값과 최대값(maxOccurs) 중 큰 값을 빈도수 속성으로 갖는다.

Case 2 : 복합 타입(complex type) 노드 사이의 통합

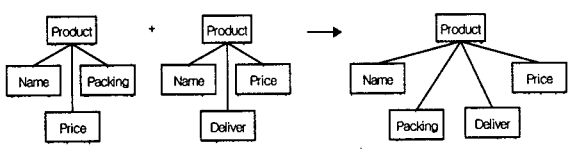
두 노드가 모두 복합 타입 노드일 경우, 통합을 위해 이름, 데이터 타입, 빈도수 외에 구조적인 충돌도 해결해야 한다. 복합 타입 노드는 하위 엘리먼트나 속성을 포함한다.



(a) 선택 연산자 간의 통합의 예.



(b) 선택 연산자와 순서 연산자의 통합의 예.



(c) 순서 연산자 간의 통합의 예.

그림 3. 복합 타입 노드간 통합의 예.

표 1. 스키마 통합에 관한 기존 연구와의 비교

	Cruz 등	Jeong과 Hsu	Lee 등	Mello 등	Castano 등	Zhang과 Liu	제안된 알고리즘
대상	XML 스키마	XML DTD	XML 스키마	XML DTD	XML DTD	XML 스키마	XML 스키마
클러스터링	어휘적	구조적	×	어휘적	어휘적	어휘적	어휘적
개념모델	RDF 온톨로지	CFG	스키마	객체지향 개념모델	X-class	UML 클래스 다이어그램	스키마 트리
이름 통합	○	○	○	○	○	○	○
타입 통합	×	×	○	○	○	○	○
빈도지시자 통합	×	○	○	○	○	×	○
구조간 통합	○	○	○	○	○	○	○
노드 - 속성간 통합	×	×	○	×	○	×	○
경로간 통합	×	×	×	×	×	×	○
최적화	○	○	×	○	△	△	○
자동화	○	○	○	×	×	×	○

통합 노드는 두 노드의 하위 노드의 합집합을 자식 노드로 가지며, 노드의 이름 및 빈도수 속성의 충돌은 심플 타입 노드의 경우와 같은 방법으로 해결한다. 하위 구조가 선택 및 순서 연산자가 교차된 구조일 경우, 바깥쪽부터 안쪽으로 구조적 충돌을 해결한다.

두 노드의 하위 노드가 각각 순서 혹은 선택 연산자일 경우 하위노드의 합집합의 형태로 통합 노드의 구조가 결정된다. 한편, 순서 연산자간의 충돌의 경우, 두 연산자를 선택 연산자로 묶어 결정한다. <그림 3>은 복합 타입 노드간 통합의 예를 보여준다.

Case 3 : 심플 타입 노드와 복합 타입 노드 사이의 통합

심플 타입 노드와 복합 타입 노드를 통합할 경우, 통합 노드는 복합 타입 노드로 생성된다. 새로운 노드는 복합 타입 노드의 하위 구조를 포함하며 심플 노드간의 통합과 같은 방법으로 노드 사이의 이름 및 빈도수 충돌을 해결한다.

각각의 경우 로컬 스키마의 접근을 위해 대응되는 노드 및 생성되는 통합 스키마의 노드 사이의 매칭을 매칭 테이블에 추가한다. 통합 스키마의 노드와 매칭관계를 갖지 않는 소스 및 타겟 스키마의 노드는 부모 노드와 대응되는 통합 노드의 자식 노드, 또는 자식 노드와 대응되는 통합 노드의 부모 노드로 대응된다.

3.3 최적화

제안된 휴리스틱을 통해 중복된 노드와 관계를 제거하고 재구성함으로서 최적화된 통합 스키마를 생성한다.

제안된 스키마 최적화 방법은 다음과 같다. 선택 연산자로 연결된 순서 연산식 중 LCS(Longest Common Subsequence)를 추출하여 선택 연산과 순서 연산으로 분리한다. 그리고 동일한 부모 노드를 가지고 공통의 레이블 및 하위 구조를 가진 경우, 하나의 서브트리로 통합한다. 이렇게 통합된 스키마 내의 사이클 및 transitive edge를 제거한다. 그리고 중복되는 빈도 지시자를 제거한다.

4. 실험결과

기존 연구 [2][3][6][9]에서 사용한 데이터로 실험한 결과, 제안된 방법은 기존 연구와 유사한 정확률과 축약률 [4]을 보였다. 또한 사용자의 개입 없이 스키마 경로간의 매칭을 수행함으로서 알고리즘의 시간 복잡도를 크게 줄이고 스키마 경로간의 직접적인 변환 비용을 고려하여 유사도를 계산할 수 있었다.

<표 1>은 스키마 통합에 관한 기존의 연구들과 제안된 알고리즘을 비교한 결과이다. 본 논문에서는 XML 스키마 통합 알고리즘을 제안하였다. 제안된 알고리즘은 어휘적 유사도에 기반하여 유사한 노드들을 클러스터링하고 그룹핑된 노드들 간의 이름, 데이터 타입 및 빈도 지시자간의 충돌을 제거하였다. 경로간 통합 과정을 통해 스키마 통합의 시간 복잡도를 향상시키고 여러 휴리스틱을 이용하여 사용자의 개입 없이 통합 과정을 진행하였다.

참고 문헌

[1] 임태우, 이경호, "XML 스키마 클러스터링을 위한 효율적인 알고리즘", 한국정보과학회 04 봄 학술발표논문집(B), pp.34-36, 2004.

[2] Isabel F. Cruz, Huiyong Xiao, and Feihong Hsu, "An Ontology-Based Framework for XML Semantic Integration," International Defence Exhibition & Seminar, pp. 217-226, 2004.

[3] Patrick Lehti and Peter Fankhauser, "XML data integration with OWL: experiences and challenges," Proc. Int'l Symposium Applications and the Internet, pp. 160-167, 2004.

[4] Euna Jeong and Chun-Nan Hsu, "View Inference for Heterogeneous XML Information Integration," Journal of Intelligent Information Systems, pp. 81-99, 2003.

[5] Kyong-Ha Lee, Mi-Hye Kim, Kyu-Chul Lee, Byung-Seob Kim, and Mi-Young Lee, "Conflict classification and resolution in heterogeneous information integration based on XML Schema," Proc. IEEE Conf. Computers, Communications, Control and Power Engineering, Vol. 1, pp. 93-96, 2002.

[6] Ronaldo dos Santos Mello, Silvana Castano, and Carlos A. Heuser, "A Method for the Unification of XML Schemata," Information & Software Technology, pp. 241-249, 2002.

[7] Silvana Castano, Alfio Ferrara, G. S. Kuruvilla Ottathycal, and Valeria De Antonellis, "A Disciplined Approach for the Integration of Heterogeneous XML Datasources," Proc. Int'l Workshop Web Semantics, pp. 103-110, 2002.

[8] K. Passi, S. Madria, Bipin S., S. Bhowmich, M. Mohania, "A Model for XML Schema Integration," Proc. 3rd ECWEB, pp. 193-202, 2002.

[9] Yan-Feng Zhang and Wei-Yi Liu, "Semantic Integration of XML Schema," Proc. Int'l Conf. Machine Learning and Cybernetics, pp. 1058-1061, 2001.

[10] Farshad Hakimpour, and Andreas Geppert, "Resolving semantic heterogeneity in schema integration," Proc. Int'l Conf. Formal Ontology in Information Systems, pp. 297-308, 2001.

[11] F. Noy and M. A. Musen. "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment," Proc. the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI 2000), pp. 450-455, 2000.

[12] George A. Miller, "WordNet: A Lexical Database for English," Communications of the ACM, Vol. 38, No. 11, pp. 39-41, 1995.