Journal of Korea Multimedia Society Vol. 8, No. 7, July 2005(pp. 857-868)

XML 스키마 클러스터링을 위한 효율적인 알고리즘

임태우†, 이경호††

요 약

스키마 클러스터링은 스키마의 통합을 위한 전처리 단계로서 중요하다. 본 논문에서는 XML 스키마를 클러스터링하기 위한 효율적인 방법을 제안한다. 제안된 방법은 먼저 스키마 사이의 유사도를 계산한다. 특히 두 스키마를 통합하는데 드는 비용이 적을수록 유사하다는 가정하에 스키마 사이의 유사도를 공통된 구조의 크기로 정의한다. 이를 위해서 경로 사이에 서로 대응하는 엘리먼트의 합이 최대가 되는 경로간의 일대일 매칭을 추출한다. 또한 계산된 유사도값에 기반하여 계층적 클러스터링 방법을 적용한다. 제안된 방법의 성능을 평가하기 위해서 다수의 XML 스키마를 대상으로 실험한 결과, 99%의 정확률과 93%의 클러스터링률을 보여 기존의 알고리즘보다 우수하였다.

An Efficient Algorithm for Clustering XML Schemas

Tae-Woo Rhim†, Kyong-Ho Lee††

ABSTRACT

Schema clustering is important as a prerequisite to the integration of XML schemas. This paper presents an efficient method for clustering XML schemas. The proposed method first computes similarities among schemas. The similarity is defined by the size of the common structure between two schemas under the assumption that the schemas with less cost to be integrated are more similar. Specifically, we extract one-to-one matchings between paths with the largest number of corresponding elements. Finally, a hierarchical clustering method is applied to the value of similarity. Experimental results with many XML schemas show that the method has performed better compared with previous works, resulting in a precision of 99% and a rate of clustering of 93% in average.

Key words: XML Schema(XML 스키마), Clustering(클러스터링), Schema Integration(스키마 통합)

1. 서 론

XML(eXtensible Markup Language) [1]은 문서와 데이터를 구조적으로 표현할 수 있는 메타언어이며 플랫폼에 독립적이라는 특징 때문에 인터넷을 비롯한 다양한 분야에서 정보 교환을 위한 표준으로 널리 사용되고 있다.

인터넷상에 분포되어 있는 XML 문서를 검색하기 위해서는 XML 스키마1)를 통해 문서의 구조와 의미를 파악하여야 한다. 그런데 웹상에는 수많은 문서 및 스키마가 존재할 뿐만 아니라 계속하여 새로 생성되고 있다. 이러한 문서들에 대한 검색 및 연산의 필요성이 증대되면서 유사한 도메인의 스키마를 하나로 묶는 스키마 통합(schema integration)에 대한 관심이 증가하고 있다. 특히 스키마 클러스터링은 통합 스키마를 생성하기 위한 전단계로서 매우 중요하다.

기존에 XML 스키마 클러스터링을 위한 여러 방

※ 교신저자(Corresponding Author) : 이경호, 주소 : 서울시 서대문구 신촌동 134(120-749), 전화 : 02)2123-5712, FAX : 02)365-2579, E-mail : khlee@cs.yonsei.ac.kr
접수일 : 2004년 5월 4일, 완료일 : 2005년 1월 3일
† 준회원, 삼성전자 정보통신총괄사업부 근무
(E-mail : rtaeng@hanmail.net)
†† 정회원, 연세대학교 컴퓨터과학과 조교수
※ 이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음.(KRF-2003-003-D00429)

1) 본 논문에서 XML 스키마는 XML Schema와 XML DTD(Document Type Definition)을 포함한 개념이다.

법이 제안되었다[2,3]. 그러나 기존 연구의 대부분은 단순히 엘리먼트(element)의 구조적 및 어휘적 유사도에 기반하기 때문에 스키마 통합 측면에서 문제점을 갖는다. 예를 들어 Person-Dog과 Dog-Person처럼 부모관계가 뒤집힌 경우, 기존 연구는 유사한 도메인의 스키마로서 계산하지만, 스키마 통합과정에는 많은 비용이 요구된다. 따라서 본 논문에서는 통합 스키마를 생성하는데 드는 비용에 기반하여 스키마간의 유사도를 제안하고, 계산된 유사도 값을 이용한 클러스터링 방법을 제안한다.

제안된 방법은 두 스키마를 통합하는데 드는 비용이 적을수록 스키마간의 유사도가 높다는 가정하에 스키마 사이의 공통된 구조의 크기를 계산한다. 이를 위해서 경로사이에 서로 대응하는 요소의 합이 최대가 되는 경로간의 일대일 매칭을 추출한다. 또한 계산된 유사도값에 기반하여 계층적 클러스터링 방법을 적용한다. 특히 보다 정교한 수준의 엘리먼트간 매칭을 위하여 축약어 사전과 동의어 사전을 이용한다. 제안된 알고리즘의 성능을 평가하기 위해서 다양한 범주에 속하는 XML 스키마를 대상으로 실험한 결과, 동일한 실험 데이터를 사용한 기존 연구보다 우수하였다.

본 논문의 구성은 다음과 같다. 2절에서는 스키마를 효과적으로 표현하기 위한 문서모델을 제안하고, XML 클러스터링에 대한 기존 연구를 간략히 기술한다. 3절에서는 제안된 스키마 클러스터링 방법을 자세히 기술한다. 4절에서는 제안된 알고리즘에 대한 성능 분석 및 기존 연구와의 차이를 기술한다. 마지막으로 5절에서는 결론 및 향후 연구방향을 기술한다.

2. 문서모델 및 관련연구

본 절에서는 XML 스키마를 효과적으로 표현할 수 있는 문서 모델을 제안한다. 또한 XML 클러스터링에 관한 기존 연구의 특징 및 문제점을 자세히 기술한다.

2.1 문서모델

본 논문에서는 XML 스키마를 효과적으로 표현하기 위한 문서 모델을 정의한다.

```
<!ELEMENT HOMEVISIT (PATIENT)>
<!ELEMENT PATIENT (NAME, ADDRESS, GENDER, PHONE, ID, SERVICES)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT GENDER (#PCDATA)>
<!ELEMENT PHONE (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT SERVICES (DATE, SERVICE*, PRODUCT*)>
<!ELEMENT DATE (#PCDATA)>
<!ELEMENT SERVICE (NAME, TIME, PRICE)>
<!ELEMENT TIME (#PCDATA)>
<!ELEMENT PRICE (#PCDATA)>
<!ELEMENT PRODUCT (#PCDATA)>
```

그림 1. 환자의 정보를 표현하기 위한 XML 스키마의 예

XML 스키마는 XML 문서에 포함될 엘리먼트의 이름과 구조, 애트리뷰트(attribute)의 이름 등 문서 형식을 정의한다. 논리적 구성요소인 엘리먼트의 이름과 계층구조 및 빈도수(cardinality)는 물론이고, 각각의 엘리먼트가 포함할 수 있는 정보의 데이터 타입을 정의한다. XML 스키마를 구성하는 엘리먼트는 하위 엘리먼트를 포함할 수 있고 애트리뷰트는 각 엘리먼트에 적용되어 엘리먼트의 특징을 기술한다. 그림 1은 환자의 정보를 표현하기 위해 사용 중인 XML 스키마의 예이다. 여기서 엘리먼트 SERVICE는 하위 엘리먼트로서 NAME, TIME, 그리고 PRICE을 포함하며 PRICE는 내용으로 문자열을 포함한다(#PCDATA는 엘리먼트가 내용으로 문자열을 포함함을 의미한다.).

본 논문에서는 XML 스키마를 표현하기 위해 트리구조에 기반한 문서모델을 제안하며 제안된 문서모델에 따라 표현된 트리구조를 스키마 트리라고 정의한다. 스키마 트리를 구성하는 각각의 노드는 스키마의 엘리먼트를 표현한 것으로서 엘리먼트의 이름을 레이블(label)로 갖고 노드의 속성 값으로 타입(type)과 빈도수를 갖는다. 스키마 트리에서의 노드는 다른 노드를 자식으로 가지는 중간노드와 자식을 갖지 않는 단말노드로 구성된다(그림 2에서 단말노드는 음영으로 나타내었다.). 중간노드는 타입 속성값을 갖지 않으며 단말노드는 해당 엘리먼트나 애트리뷰트의 데이터 타입을 값으로 갖는다. 빈도수 속성은 4개의 빈도 지시자(none, *, +, ?) 중 한 개 혹은 (최소값, 최대값)의 쌍을 값으로 갖는다. 그림 2는 그림 1의 스키마를 제안된 문서 모델로 표현한 결과이다.

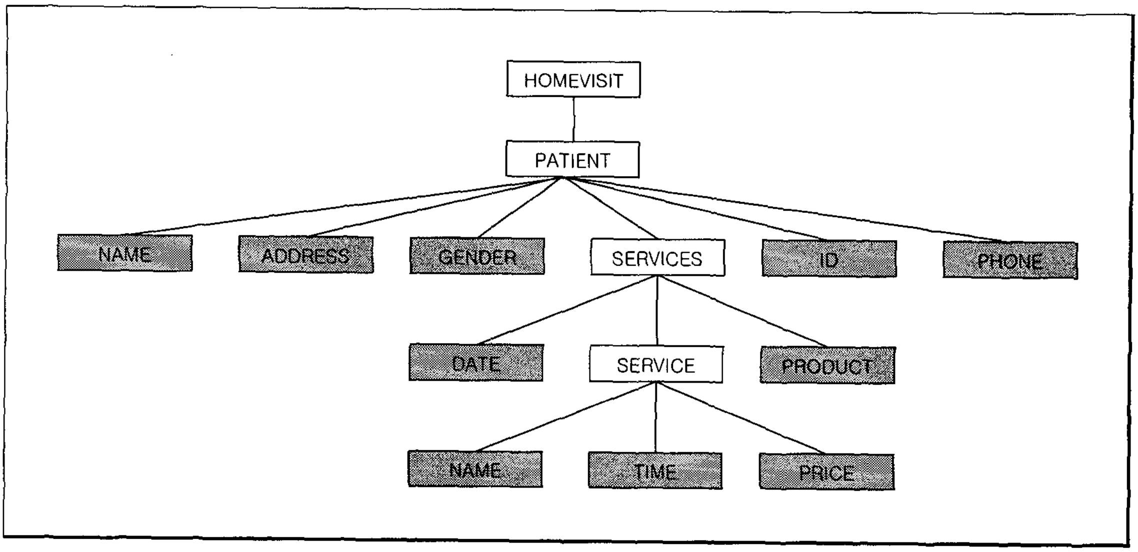


그림 2. 그림 1의 XML 스키마를 문서모델로 표현한 결과

2.2 관련연구

기존에 XML 클러스터링을 위한 연구결과는 크게 문서를 대상으로 한 방법과 스키마를 대상으로 한 방법으로 나누어진다. 본 절에서는 본 논문과 같이 XML 스키마를 대상으로 하는 방법은 물론이고 XML 문서를 대상으로 제안된 클러스터링 방법의 특징 및 문제점을 기술한다. 표 1은 XML 스키마 또는 문서 클러스터링에 대한 연구 결과를 요약한 것이다.

Lee 등 [2] 이 제안한 XClust는 두 DTD간의 유사도를 측정하고 그 유사도를 기준으로 계층적 클러스터링 기법을 사용하여 같은 도메인의 DTD들을 클러스터링한다. 이 시스템에서는 DTD를 트리 형태로 모델링한 후 DTD를 구성하는 각 노드의 이름과 속성, 뿌리노드(root node)로 가는 경로간 비교를 통해 언어적 및 구조적 유사도를 계산한다. 또한 직계자손 및 단말노드끼리의 유사도를 통해 두 노드간의 문맥적 유사성을 결정한다. 이렇게 구해진 노드간 유사도를 통해 DTD 사이의 유사도를 계산하고 그 유사도

표 1. XML 문서 및 스키마 클러스터링 방법

대상	저자	연도	특징
XML 스키마	Lee 등 [2]	2001	트리로 표현된 스키마를 구성하는 노드 사이의 언어적 유사도와 구조적 유사도를 정의 및 계산하여 두 스키마 사이의 유사도 계산
	Jeong과 Hsu [4]	2001	각 스키마가 통합 스키마로 통합되기 위한 변환연산의 양에 기반하여 스키마간 유사도 계산
XML Document	Wang 등 [11]	2004	각 문서의 구조 그래프를 제안하고 문서들 사이의 공통된 간선의 개수로서 문서간의 유사도를 계산
	Dalamagas 등[13]	2004	XML 문서의 재귀 및 반복노드를 제거한 후 두 문서간의 변환 연산의 크기로 문서간의 유사도를 계산
	Francesca 등 [12]	2003	문서를 나타내는 트리간의 매칭을 통해 구해진 공통구조의 크기에 기반하여 문서들을 클러스터링하는 방법을 제안
	Charnyote과 Hammer [3]	2003	XML 문서 사이의 거리함수를 정의하고 거리 값에 기반한 MCM (Multi-strategy Clustering Model) 클러스터링 방법을 제안
	Nierman과 Jagadish [6]	2002	XML 문서를 다른 문서로 변환하는데 필요한 변환연산의 비용을 이용하여 유사한 구조를 가진 문서들을 클러스터링 하는 방법을 제안
	Guillaume와 Murtagh [5]	2000	문서간의 참조관계를 그래프로 모델링한 후 그래프를 구성하는 간선(edge)의 가중치에 기반하여 파티셔닝하는 방법을 제안

행렬을 이용하여 유사도가 높은 DTD쌍으로부터 점진적으로 DTD들을 클러스터링한다. 이 방식은 각 DTD의 노드 사이의 문맥적인 요소까지 고려하여 높은 정확도를 갖지만 시간 복잡도가 높고 서로 크기가 다른 DTD들 사이에서는 정확한 유사도를 구할 수 없다는 문제점을 갖는다.

Jeong과 Hsu [4]가 제안한 XML IIA(Information Integration Agent)는 정보검색을 위한 시스템으로서 XML 문서의 효과적인 질의를 위해서 XML 스키마를 클러스터링 및 통합한다. 각 DTD 사이에 필요한 변환 연산의 길이를 이용하여 계산된 유사도에 기반하여 계층적인 클러스터링을 수행한다.

Charnyote과 Hammer의 MCM(Multi-strategy Clustering Model) [3]은 제안된 XML 문서 사이에 거리 함수를 적용하여 계산된 거리값을 이용하여 유사한 XML 문서들을 클러스터링한다. 단말노드의 경우, 이름 사이의 LCS(Longest Common Subsequence) [7]의 비율을 통해 거리를 계산한다. 또한 단말노드가 아닌 경우에는 자식노드 및 속성들의 거리값에 각각의 가중치를 곱해서 거리값을 계산한다. 두 XML 문서사이의 거리는 두 문서트리의 루트노드 사이의 거리값으로 결정된다. 그런데 이 시스템에서 사용한 실험 데이터는 20개의 기사(article)로부터 변경되었기 때문에 실제 웹상에 존재하는 XML 문서들을 반영한 결과라고는 볼 수 없다.

Guillaume와 Murtagh [5]는 XML 문서 클러스터링 방법을 제안한다. 이 논문에서는 XML 문서를 클러스터링하는 문제를 그래프 이론을 이용하여 최적의 파티션을 찾는 문제로 간주한다. 데이터베이스 내의 문서를 각각 하나의 노드로 모델링하고, 문서 사이의 링크를 그 노드간의 가중치를 가진 간선으로 모델링한 그래프에서 최적의 파티션을 찾아냄으로서 데이터베이스 내의 문서들을 여러 개의 클러스터로 파티셔닝할 수 있다. 또한 동일한 키워드를 공유하는 문서간에 키워드 링크를 추가함으로서 클러스터링의 정확도를 높인다.

Nierman과 Jagadish [6]는 구조적 유사도에 기반한 XML 문서 클러스터링 방법을 제안한다. 문서간의 구조적인 유사도를 구하기 위해 각 문서의 구조적인 정보를 트리 형태로 모델링한다. 그리고 삽입, 삭제 그리고 갱신의 변환연산을 정의하며 최소 비용의 변환연산의 집합을 문서간의 거리로 계산한다. 이렇게 계산된 거리에 기반하여 문서들을 계층적으로 클러스터링한다.

Wang 등 [11]은 효율적인 질의를 위해 XML 문서 클러스터링 방법을 제안한다. 각 XML 문서간의 공통된 간선의 비율로서 거리를 계산하고 임계값 이내의 거리를 가진 문서를 이웃 문서로 정의한다. 각 문서 및 클러스터 사이의 공통된 이웃 문서의 수를 통해 유사도를 구하고 이에 기반하여 XML 문서들을 계층적으로 클러스터링한다.

Francesca 등 [12]은 XML 문서를 나타내는 두 트리의 엘리먼트 사이의 매칭을 통해 공통구조를 추출한다. 공통구조의 크기로서 정의된 가중치 값에 기반하여 계층적 클러스터링 알고리즘을 적용한다. 또한 트리간의 병합과 단순화 과정을 통해 클러스터의 대표 구조를 추출한다.

Dalamagas 등 [13]은 XML 문서의 구조적 특성에 기반한 클러스터링 알고리즘을 제안한다. 이 논문에서는 재귀 및 반복 노드로 인한 구조적인 차이를 극복하기 위해 구조 요약(structural summary)를 제안한다. 다이내믹 프로그래밍을 이용하여 계산된 트리 변환 비용에 기반하여 XML 문서를 계층적으로 클러스터링한다.

한편 XML 문서는 XML 스키마의 인스턴스(instance)로서 스키마의 구조를 결정하는 빈도 지시자나 선택 연산자(choice operator) 등을 포함하지 않는다. 그러므로 XML 문서를 대상으로 하는 클러스터링 알고리즘을 스키마에 적용하는데는 적합하지 않다.

3. 제안된 클러스터링 알고리즘

본 절에서는 제안된 XML 스키마 클러스터링 알고리즘을 자세히 기술한다. 제안된 방법은 그림 3과 같이 단순화, 유사도 계산, 그리고 클러스터링의 세 단계로 구성된다.

3.1 단순화

XML 스키마간의 유사도 비교를 위해서 먼저 스키마를 스키마 트리로 모델링한다. 한편 스키마에서 선택 연산자로 연결되어 있는 부분은 모델링 과정에서 많은 후보 스키마 트리들을 생성한다. 그러므로 구조적 정보의 손실을 최소화하는 범위에서 선택 연

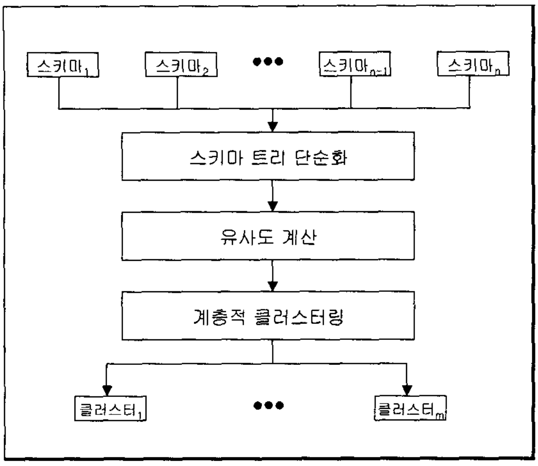


그림 3. 제안된 XML 스키마 클러스터링 과정

산자를 순서 연산자(sequence operator)로 변환한다. 이를 위하여 제안된 방법은 그림 4와 같이 Lee 등이 제안한 단순화 규칙을 적용한다.

단순화 규칙은 구조적 정보의 손실 정도에 따라 우선순위를 갖는다. 예를 들어, 규칙 E1~E5는 구조적 정보를 잃어버리지 않는 규칙으로서 높은 우선순위를 갖지만, 규칙 L6~L7은 변환 과정에서 정보 손실이 발생하여 낮은 우선순위를 갖는다.

3.2 유사도 계산

본 논문에서는 임의의 클러스터에 속하는 스키마

규칙	변환 방법	우선순위
E1	(a|b) ⇔ (a*,b*)	high
E2	((a|b)*)+ ⇔ ((a|b)+)* ⇔ (a*,b*) ((a,b)*)+ ⇔ ((a,b)+)* ⇔ (a,b)*	high
E3	((a|b)*)? ⇔ ((a|b)?)* ⇔ (a*,b*) ((a,b)*)? ⇔ ((a,b)?)* ⇔ (a,b)*	high
E4	((a|b)+)? ⇔ ((a|b)?)+ ⇔ (a*,b*) ((a,b)+)? ⇔ ((a,b)?)+ ⇔ (a,b)*	high
E5	((E)*)* ⇔ (E)* ((E)?)? ⇔ (E)? ((E)+)+ ⇔ (E)+	high
L6	(a|b) ⇒ (a,b) (a|b)+ ⇒ (a+,b+) (a|b)? ⇒ (a?,b?)	low
L7	(a,b)* ⇒ (a*,b*) (a,b)+ ⇒ (a+,b+) (a,b)? ⇒ (a?,b?)	low

그림 4. 단순화 규칙 [2]

들을 포괄하는 스키마를 통합 스키마라고 정의한다. 본 논문에서 제안된 스키마 사이의 유사도는 통합 스키마를 생성하는데 필요한 비용에 기반한다. 즉, 임의의 두 스키마를 통합 스키마로 변환할 때 변환 비용이 적게 들수록 유사하다고 간주한다. 따라서 공통되는 부분을 보다 많이 포함할수록 두 개의 스키마는 더 큰 유사도를 갖는다.

그러므로 제안된 유사도는 식 (1)과 같이 두 스키마간의 공통된 구조의 비율로 정의한다.

$$\text{스키마 트리간 유사도} = \frac{\text{공통구조}}{|T_s| + |T_t| - \text{공통구조}} \quad (1)$$

$|T_s|$: 소스스키마트리의노드들의집합,

$|T_t|$: 타겟스키마트리의노드들의집합

스키마 트리간의 구조적 유사도를 구하기에 앞서 먼저 두 노드 Ns와 Nt에 대한 노드 유사도(Ns, Nt)를 정의한다. 노드 유사도는 수식 (2)와 같이 노드의 이름, 빈도수 속성 및 데이터 타입 속성의 유사도 값을 이용해 계산한다. 유사도 계산에 사용되는 스키마가 XML DTD일 경우, 노드간의 데이터 타입이 존재하지 않으므로 w_3은 0이 된다.

$$\text{노드 유사도}(Ns, Nt) = w_1 \times \text{이름 유사도}(Ns, Nt) + w_2 \times \text{빈도 유사도}(Ns, Nt) + w_3 \times \text{데이터 타입 유사도}(Ns, Nt) \quad (\text{단, } w_1 + w_2 + w_3 = 1) \quad (2)$$

이름 유사도는 노드의 레이블간의 어휘적인 유사도를 나타낸다. 먼저 각 노드명을 대문자나 특수문자를 기준으로 토큰화하여 각 토큰 사이의 유사도를 계산한다. 노드명 사이의 이름 유사도는 토큰들 사이의 유사도를 이용해 수식 (3)과 같이 토큰 사이의 유사도의 합을 전체 토큰수로 나눈 값으로 정의한다.

$$\text{이름 유사도}(Ns, Nt) = \frac{2 \times \sum \text{유사도}(Ns_i, Nt_j)}{|Ns| + |Nt|} \quad (3)$$

Ns_i: 소스 노드의 토큰, $1 \le i \le n$

Nt_j: 타겟 노드의 토큰, $1 \le j \le m$

토큰 사이의 정확한 유사도를 계산하기 위해 축약어 사전과 동의어 사전을 이용한다. 축약어 사전은 축약어와 축약어의 전체이름을 가진 테이블 구조로 해당 토큰명과 같은 이름의 축약어가 있을 경우에

토큰명을 해당 축약어의 전체이름으로 대체한다. 축약어 대체를 끝낸 토큰들은 다른 노드의 토큰들과 문자열 비교를 통해 동일한 토큰의 경우 1.0의 유사도를 반환한다.

동일하지 않은 토큰의 경우 동의어 사전[8]을 이용한다. 본 논문에서 사용된 동의어 사전은 같은 의미를 가진 여러 단어들 사이의 관계를 정의한다. 문자열이 동일하지 않은 두 토큰에 대하여 동의어 관계가 존재할 경우, 두 토큰은 0.8의 유사도값을 갖는다.

스키마 트리를 구성하는 단말노드는 다양한 데이터 타입을 타입 속성으로 갖는다. 그런데 타입 속성이 서로 다른 노드간의 변환은 노드에 포함된 정보의 손실을 가져올 수 있다. 이러한 변환에 의한 정보 손실의 정도에 기반한 타입 속성간의 유사도를 제안한다. 본 논문에서는 노드의 타입 속성간의 유사한 정도를 해당 노드가 포함하는 정보의 손실 정도에 따라 '동일', '비손실 변환가능', '손실 변환가능', 그리고 '변환불가'의 4단계로 구분한다. 동일한 타입 속성을 가진 노드 사이의 변환의 경우, 가장 큰 유사도를 부여하고, 비손실 변환가능, 손실 변환가능, 그리고 변환불가의 단계의 따라 점점 낮은 유사도를 부여한다. XML 스키마 트리에서 노드가 가지는 주요 타입 속성간의 유사도는 표 2와 같다.

두 노드의 빈도 유사도는 표 3과 같이 각 노드의 빈도수 속성 사이의 유사도로 나타낸다. 한편 XML Schema의 경우, 빈도수를 구체적으로 명시할 수 있다. 이때 표 3에 의하여 계산될 수 없는 경우, 해당 빈도수의 최대값이나 최소값이 동일할 경우 0.9를, 두 값이 모두 다를 경우 0.7을 할당한다.

이와 같은 노드 유사도를 이용하여 트리간의 유사도를 계산하기 위해서 먼저 두 트리의 뿌리노드로부터 단말 노드까지의 모든 경로 $SPath_i$, $0 \le i \le n$와 $TPath_j$, $0 \le j \le m$를 추출한다. 또한 모든 경로들 간의 LCS를 계산한다. 경로간 LCS는 두 경로 사이의 최장 공통경로를 말하며 각 경로의 노드 사이의 1:1 매칭을 통하여 추출한다. 특히 두 노드의 유사도가 임계값 Th_{node} 이상일 때 두 노드 사이에 매칭관계를 생성한다. 어떤 경우에도 경로의 노드들 사이의 순서 관계는 유지되며, 경우에 따라 두개 이상의 LCS가 존재할 수도 있다. 예를 들어, 두 경로 {PurchaseOrder - DeliverTo - Address - Phone}과 {Employee - Address - Contact - Phone}의 LCS는 {Address - Phone}에 해당한다. 또한 두 경로 {Item - Name}과 {Name - Item}의 경우, LCS는 Item 혹은 Name이 된다.

제안된 알고리즘은 두 스키마간의 공통구조를 계산하기 위해서 LCS의 합이 최대에 해당하는 경로간의 일대일 매칭을 찾는다. 이때 계산된 LCS의 합을 GCS(Greatest Common Subset)라고 정의한다. 제안된 알고리즘에서는 GCS를 계산하기 위해 소스 및 타겟 트리들의 각 경로를 정점(vertex)으로, 경로간 LCS 관계를 간선으로 갖는 그래프 G(V,E)를 모델링

표 3. 빈도수 속성간 유사도

	*	+	?	none
*	1	0.9	0.8	0.7
+	0.9	1	0.7	0.8
?	0.8	0.7	1	0.8
none	0.7	0.8	0.8	1

표 2. 타입 속성간의 유사도

타겟노드의 데이터 타입 / 소스노드의 데이터 타입	string	decimal	double	float	boolean	duration	dateTime
string	1.0	0.4	0.4	0.4	0.0	0.4	0.4
decimal	0.7	1.0	0.7	0.7	0.4	0.0	0.0
double	0.7	0.4	1.0	0.4	0.0	0.0	0.0
float	0.7	0.4	0.7	1.0	0.0	0.0	0.0
boolean	0.7	0.7	0.0	0.0	1.0	0.0	0.0
duration	0.7	0.0	0.0	0.0	0.0	1.0	0.4
dateTime	0.7	0.0	0.0	0.0	0.0	0.4	1.0

Algorithm FindGCS
입 력 : 소스 스키마 트리 T_s와 타겟 스키마 트리 T_t
출 력 : T_s와 T_t 사이의 GCS
1. T_s와 T_t의 루트로부터 각 단말노드로의 경로 $SPath_i$, $0 \le i \le n$,와 $TPath_j$, $0 \le j \le m$,를 추출한다.
2. 각 경로 $SPath_i$와 $TPath_j$간의 LCS를 계산한다.
3. 각 경로 $SPath_i$와 $TPath_j$를 정점으로 하며 각 경로간 LCS의 크기를 간선의 가중치로 갖는 가중치 이분 그래프 $K_{n,m}$를 생성한다.
4. $K_{n,m}$에서 최대 이분 매칭을 찾는다. 이때의 LCS의 합이 GCS가 된다.

그림 5. 제안된 GCS 계산 알고리즘

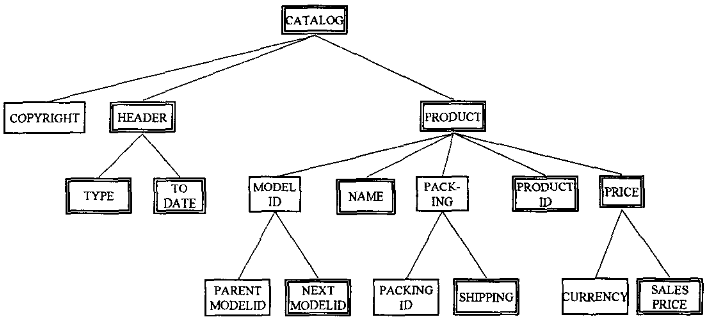
한다. 그래프 G(V,E)에서 V는 정점의 집합으로서 소스 및 타겟 트리의 각 경로를 의미하며 E는 정점간의 간선으로서 경로간의 LCS의 크기를 가중치 값으로 갖는다. 그래프의 정점은 소스트리 및 타겟트리에 해당하는 두 종류로 구분된다. 즉, n개의 소스 경로와 m개의 타겟 경로로 구성된 그래프 G(V,E)는 가중치 이분 그래프(weighted bipartite graph) $K_{n,m}$에 해당한다. 따라서 GCS를 계산하는 문제는 $K_{n,m}$에서 최대 이분 매칭(maximal bipartite matching) [9]을 찾는 문제에 해당한다. 제안된 알고리즘에 대한 자세한 기술은 그림 5와 같다.

이렇게 구해진 GCS의 크기를 두 트리의 전체 노드의 수로 나눈 것이 두 트리의 유사도 값이 된다.

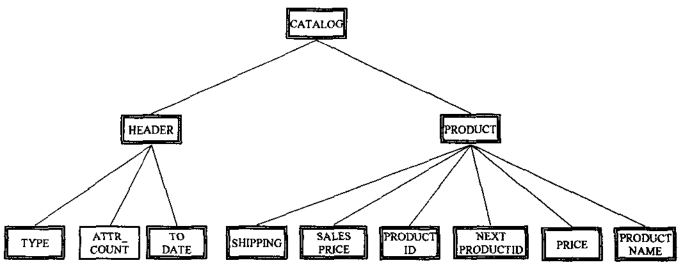
수식 (1)을 제안된 GCS에 따라 표현하면 다음 수식 (4)와 같다.

$$\text{스키마 트리간 유사도} = \frac{|T_{gcs}|}{|T_s| + |T_t| - |T_{gcs}|} \tag{4}$$

예를 들어, 그림 6의 소스 스키마 트리와 타겟 스키마 트리의 구조적 유사도를 계산해 보자. 두 스키마 트리의 단말노드가 각각 11개, 9개이므로 총 11개의 소스 경로와 9개의 타겟 경로가 존재한다. 이때의 GCS는 이중선으로 표시된 부분으로부터 계산되며 두 스키마 트리간의 유사도는 식 (4)에 의해 0.61(= 11 / (17+12−11))이다.



(a) 소스 스키마 트리



(b) 타겟 스키마 트리

그림 6. 스키마 트리의 예

3.3 클러스터링

제안된 유사도 계산 알고리즘은 클러스터링을 위해 모든 스키마들 사이의 유사도 행렬을 생성한다. 제안된 방법은 높은 유사도를 가진 스키마들간의 클러스터링을 위하여 계층적 클러스터링 기법 [10]에 기반한다. 제안된 방법은 그림 7과 같이 각 스키마 트리들의 집합과 유사도 행렬을 입력값으로 받아 각 스키마들의 클러스터를 나타내는 클러스터 배열을 생성한다. 임의의 두 클러스터간의 유사도가 임계값 $Th_{cluster}$보다 큰 경우, 단일의 클러스터로 통합한다. 만일 $Th_{cluster}$가 큰 값으로 설정될 경우 많은 스키마들이 독립적으로 클러스터링되고, 반대의 경우 적은 개수의 큰 클러스터로 통합될 것이다.

먼저 입력된 XML 스키마들을 각각 하나의 클러스터로 할당한다. 그리고 주어진 스키마간의 유사도 행렬에서 가장 높은 유사도를 지니는 두 개의 스키마를 검색한다. 유사도가 $Th_{cluster}$보다 클 경우에만 클러스터링 과정을 적용한다.

만일 두 스키마가 서로 다른 클러스터 배열에 포함되어 있을 경우, 각 스키마를 포함한 두 개의 클러스터를 병합하고 두 스키마 사이의 유사도값을 0으로 변경한다(9~11번째줄 참조). 클러스터링 과정에서 유사도 행렬을 변경하게 되는데 유사도 행렬의 최대값이 $Th_{cluster}$보다 작아질 때까지 위의 과정을 반복한다. 유사도 행렬의 모든 유사도 값이 $Th_{cluster}$보다 작아지면 클러스터링 작업을 끝내고 클러스터 배열을 결과값으로 반환한다.

4. 실험 결과

제안된 방법의 성능을 평가하기 위해 표 4와 같이 Lee 등의 연구와 동일한 데이터를 이용하여 실험하였다. 실험 데이터는 109개의 DTD로 구성되어 있으며 여행(Travel), 환자(Patient), 출판(Publication) 그리고 호텔(Hotel)의 4개의 도메인에 속한다.

```
Algorithm Clustering

입  력 : 스키마 트리의 집합 T
         유사도 행렬 Sim[n][n]                  // n개의 스키마간의 유사도값을 저장
         임계값 Th_cluster

출  력 : 클러스터 배열 C[n]                     // 유사한 스키마들을 포함하는 클러스터를 표현

1 :  입력된 XML 스키마 각각을 클러스터 배열에 할당한다.

2 :  max = 유사도 행렬에 포함된 값 중에서 최대값

3 :  While (max > Th_cluster)                         // 최대값이 임계값보다 큰 경우
4 :  {
5 :    if ( (C[row] == C[col] ) // 유사도가 max인 두 스키마 row와 col가 이미 동일한 클러스터에도 포함되
                                 어 있는 경우
6 :        Sim[row][col] = 0;  // 유사도 행렬에서 두 스키마의 유사도를 0으로 변경
7 :    else                           // row와 col 두 스키마가 다른 클러스터에도 속해있는 경우
8 :    {
9 :        C[row] = C[row] + C[col];      // 스키마 col을 포함하는 클러스터를 row를 포함하는 클러스터로 병합
10 :       C[col] = 0;
11 :       Sim[row][col] = 0;
12 :   }
13 : }
```

그림 7. 제안된 클러스터링 알고리즘

표 4. 실험 데이터

	호텔	환자	출판	여행
스키마수	26	20	20	43
스키마의 평균 노드수	9.8	27.5	15.8	9.8
스키마의 평균 단말노드수	7.5	21.4	10.1	7.3
각 노드의 평균 자식수	4.6	4.4	2.85	4.9

알고리즘의 성능 분석을 위해 클러스터링의 정확도와 시간 복잡도를 분석했다.

4.1 성능 분석

실험에 사용된 스키마는 XML DTD로서 DTD는 타입 속성이 존재하지 않으므로 노드 유사도에 사용되는 가중치 w1, w2, 그리고 w3는 각각 0.7, 0.3, 그리고 0으로 설정하였다. 또한 클러스터링에 사용되는 $Th_{cluster}$값을 바꿔가며 여러 번의 실험을 진행하였다. 본 논문에서 클러스터의 도메인은 이에 포함된 다수의 스키마의 도메인으로 정의한다. 제안된 성능평가 기준은 표 5와 같다.

정확률은 본래 속하는 클러스터로 클러스터링된 스키마들의 비율로서 정확률이 높을 경우, 스키마 통합시 더 적은 비용으로 통합 스키마를 생성할 수 있다. 또한 클러스터링률은 전체 스키마중에서 실제로 클러스터링된 비율로서 클러스터링 알고리즘이 잘 작동하는지를 나타내는 수치이다. 특히 임계값을 지나치게 높게 설정하면 스키마들은 거의 그룹핑되지

표 5. 성능평가 기준

기 준	정의
정확률	$\dfrac{\sum_{i=1}^{n} \text{클러스터}_i \text{ 의 정확률}}{\text{클러스터개수}}$
클러스터링률	$\dfrac{\text{스키마수} - \text{단일클러스터수}}{\text{스키마수}}$

않아 단일 클러스터로 존재하여 낮은 클러스터링률을 보일 것이다. 다양한 임계값에 따른 클러스터링의 정확률과 클러스터링률은 표 6과 같다.

임계값 $Th_{cluster}$가 증가함에 따라 정확률은 증가하고 클러스터링률은 감소하였다. 제안된 방법은 $Th_{cluster}$가 0.23일 때, 100%의 정확률을 보였으며 이때 전체 클러스터의 수는 9개이며 이중 단일 클러스터는 2개였다. 제안된 방법은 0.2에서 0.7사이의 임계값에서 유효하며 평균적으로 99%의 정확률과 93%의 클러스터링률을 보인다.

4.2 기존 연구와의 비교

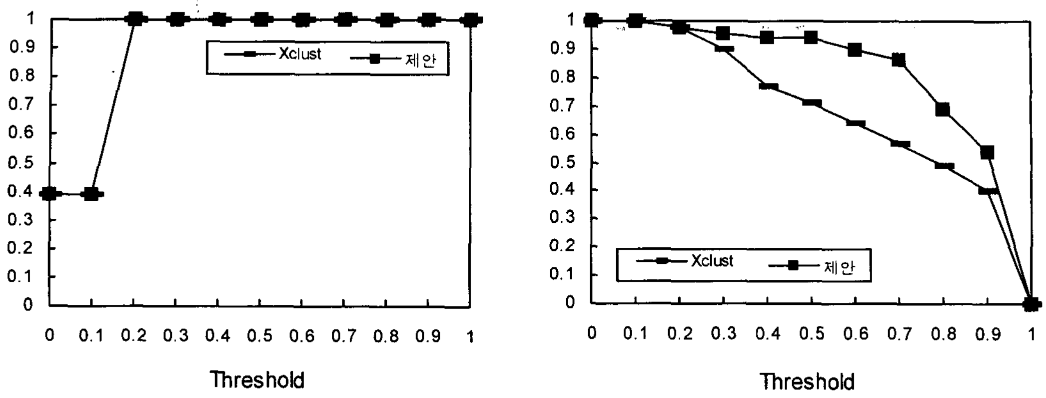
제안된 방법은 기존의 Lee 등의 방법과 비교하여 정확률 면에서 유사한 성능을 보였다. Lee 등이 제안한 XClust는 $Th_{cluster}$가 0.20일 때, 100%의 정확률을 보였으며 이때 전체 클러스터의 수는 9개이며 이중 단일 클러스터는 1개였다.

서로 다른 임계값에 따른 두 알고리즘의 정확률과 재현율을 그림 8에 나타내었다. 정확률은 클러스터링이 본래의 도메인을 얼마만큼 반영하는지를 나타내는 수치이며, 클러스터링률은 실제 클러스터링 과정을 거쳐 그룹화된 스키마의 비율에 해당한다. 결과적으로 두 알고리즘은 임계값의 변화에 대하여 거의 유사한 정확률을 보였다. 한편 클러스터링률 면에서 제안된 방법이 임계값의 변화에 보다 덜 민감함을 알 수 있다. XClust의 경우 임계값이 증가함에 따라 클러스터링률이 급격히 떨어져 유효한 결과를 얻지 못하였다.

한편 각 알고리즘의 클러스터링의 질을 평가하기 위하여 수식 (5)와 같이 대표 클러스터의 재현율을 정의한다. 대표 클러스터는 병합된 클러스터중 도메인별로 가장 많은 스키마를 포함한 클러스터로서 그림 9에서 보이는 바와 같이 제안된 알고리즘은 1, 2, 3, 그리고 4번 클러스터, XClust는 1, 2, 3, 그리고 5번

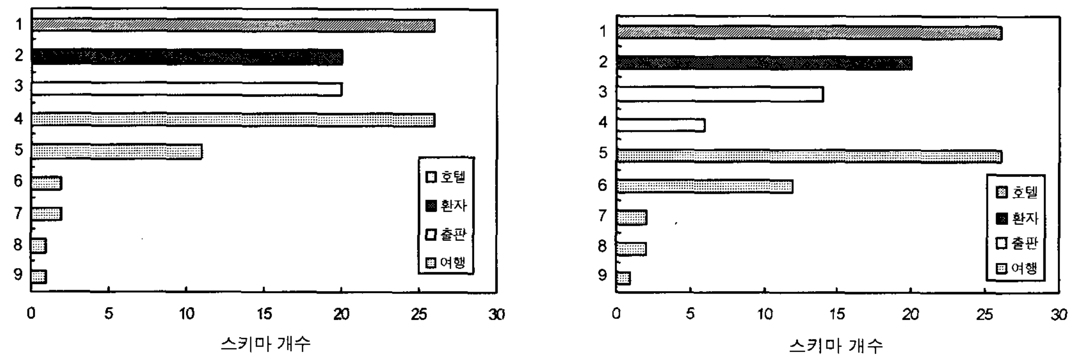
표 6. 임계값에 따른 제안된 알고리즘의 정확률과 클러스터링률

임계값($Th_{cluster}$)	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
클러스터수	1	2	8	11	17	17	25	34	56	75	109
싱글 클러스터수	0	0	2	4	7	7	11	15	34	50	109
잘못 클러스터링된 스키마 개수	66	66	2	0	0	0	0	0	0	0	0
정확률	0.39	0.39	0.98	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
클러스터링률	1.0	1.0	0.98	0.96	0.94	0.94	0.90	0.86	0.69	0.54	0.0



(a) 정확률 (b) 클러스터링률

그림 8. 제안된 방법과 XClust의 정확률 및 클러스터링률



(a) 제안된 알고리즘 ($Th_{cluster}$ - 0.23) (b) XClust ($Th_{cluster}$ - 0.20)

그림 9. 정확률이 100%일 때 클러스터의 분포

클러스터를 대표 클러스터로 생성하였다. 대표 클러스터의 재현율은 전체 스키마중 대표 클러스터에 포함된 스키마의 비율로서 재현율이 높을수록 같은 도메인의 스키마들이 같은 그룹으로 클러스터링되는 경향을 보인다. 실험 결과, 대표 클러스터의 재현율은 제안된 알고리즘이 0.84로서 XClust의 0.79보다 나은 결과를 나타냈다.

$$\text{재현율} = \frac{\sum \text{대표 클러스터에 포함된 스키마수}}{\text{전체 스키마 수}} \quad (5)$$

한편 Jeong과 Hsu의 방법의 경우, 실험 데이터와 클러스터링 결과를 입수할 수 없어서 제안된 알고리즘과 비교할 수 없었다. 한편 Jeong과 Hsu는 소수의 기본 DTD를 이용하여 도메인별로 100개의 DTD를 만들었다. 이 DTD들을 다양하게 변형하여 실험데이터를 마련하였으며 변형율에 따라 75%에서 100% 사이의 정확률을 보였다.

제안된 방법과 XClust, 그리고 Jeong과 Hsu의 알고리즘의 시간 복잡도를 표 7에 나타내었다. 제안된 알고리즘은 Jeong과 Hsu의 알고리즘과 비교하여 스키마의 종류에 따라 차이가 있지만 유사한 시간 복잡도를 갖는다. 반면에 XClust는 노드간의 언어적, 구조적, 문맥적인 유사도를 고려하여 DTD간의 유사도를 계산하기 때문에 매우 큰 시간 복잡도를 갖는다. 이에 따라 제안된 알고리즘은 XClust보다는 크게 향상된 시간 복잡도를 보인다. 최근 들어 스키마의 구조가 점차 복잡해지고 스키마의 사용범위가 넓어짐에 따라 여러 분야에서 스키마 통합의 필요성이 증가

표 7. 시간 복잡도 비교

	XClust	Jeong과 Hsu의 알고리즘	제안된 알고리즘
유사도 계산	$O(n^2m^2e^3)$	$O(n^2Km^2)$	$O(n^2e^3)$
계층적 클러스터링	$O(n^3)$		

n : 실험에 사용된 스키마의 수, m : 스키마가 포함한 노드의 수, e : 스키마의 단말 노드수, K : 각 노드의 자식수

하고 있다. 이에 따라 스키마 통합에 있어서는 정확률도 중요하지만 통합에 소요되는 시간적인 복잡도도 중요하다. 제안된 방법은 단말노드로부터 경로를 추출하는 방법을 사용하여 $O(n^2e^3)$의 시간 복잡도를 갖는다. 또한 경로간의 매칭을 통해 두 스키마간의 공통부분을 구함으로서 스키마 통합에 필요한 변환의 양을 정확하게 구해낼 수 있다. 이런 측면에서 본 논문에서 제안하는 방법은 기존 연구보다 스키마 통합에 적합하다.

5. 결론 및 향후 연구방향

본 논문에서는 XML 스키마의 통합을 위한 선행과정으로서 클러스터링 알고리즘을 제안하였다. 기존의 스키마 클러스터링에 관한 연구는 스키마를 구성하는 노드의 언어적 및 구조적 정보를 이용하여 유사도를 계산하였으며 클러스터링 후 실제 통합 스키마를 생성하기에는 한계를 갖는다. 제안된 방법은 스키마간의 최대 공통 구조를 계산함으로서 통합 스키마로의 변환이 용이한 클러스터를 생성한다.

제안된 방법은 스키마에서 추출한 경로간의 관계를 고려하여 유사도를 계산한다. Lee 등의 방법은 노드간의 1:1 매칭을 통해 스키마의 유사도를 계산한다. 이 방법은 실제 스키마의 문맥적 및 언어적인 유사도를 정확하게 계산할 수 있으나 그 유사도가 변환의 용이함을 보장하지는 않는다. 예를 들어, 소스 스키마는 {person - pet - name}으로 구성되어 있고 타겟 스키마는 {pet - person - name}으로 구성될 때 두 스키마는 0.67의 유사도를 갖지만 통합 스키마로 변환하기는 쉽지 않다. 따라서 본 연구에서는 서로 대응하는 노드간에 부모-자식 관계를 유지하면서 경로간의 유사도를 구하는데 초점을 맞추었다.

또한 제안된 방법은 노드가 아닌 각 스키마 경로간의 매칭을 수행함으로서 알고리즘의 시간 복잡도를 크게 줄이고 스키마 경로간의 직접적인 변환 비용을 고려하여 유사도를 계산할 수 있다. 또한 매칭결과의 정확성을 높이기 위해 축약어 사전과 동의어 사전을 적용하였다. 제안된 방법의 우수성을 입증하기 위해 기존 연구와 동일한 데이터를 대상으로 실험한 결과, 정확률 면에서는 기존 방법과 유사하였으며 클러스터링률과 클러스터링의 질적인 면에서는 제안된 방법이 우수하였다.

향후 보다 정확하고 빠른 클러스터링을 위해 경로간은 물론이고 서브트리간의 매칭관계를 효과적으로 추출할 수 있는 방법을 연구할 것이다. 또한 추출된 클러스터로부터 통합 스키마를 생성하는 방법을 연구할 것이다.

참 고 문 헌

[1] World Wide Web Consortium, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, http://www.w3c.org/TR/REC-xml, 2000.

[2] M. Lee, L. Yang, W. Hsu, and X. Yang, "XClust : Clustering XML Schemas for Effective Integration," *Proc. 11th Int'l Conf. Information and Knowledge Management*, pp. 292-299, Nov. 2002.

[3] Charnyote Pluempitiwiriyawej and Joachim Hammer, "Element Matching across Data-oriented XML Sources Using a Multi-strategy Clustering Model," *Data & Knowledge Engineering*, Vol. 48, Issue 3, pp. 297-333, Mar. 2004.

[4] Euna Jeong and Chun-Nan Hsu, "Induction of Integrated View for XML Data with Heterogeneous DTDs," *Proc. 10th Int'l Conf. Information and Knowledge Management*, pp. 151-158, 2001.

[5] Damien Guillaume and Fionn Murtagh, "Clustering of XML Documents," *Computer Physics Communications*, Vol. 127, Issue 2-3,

pp. 215-227, May. 2000.

[6] A. Nierman and H. V. Jagadish, "Evaluate Structural Similarity in XML Documents," *Proc. Fifth Int'l Workshop on the Web and Databases*, pp. 61-66, 2002.

[7] Claus Rick, "Simple and Fast Linear Space Computation of Longest Common Subsequence," *Information Processing Letters*, Vol. 75, Issue 6, pp. 275-281, Nov. 2000.

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

[9] Robert Sedgewick, *Algorithm in C++, Part 5 Graph algorithm, 3rd edition*, Addison Wesley, 2001.

[10] E. Gose, R. Johnsonbaugh, and S. Jost, *Pattern Recognition and Image Analysis*, Prentice Hall, 1996.

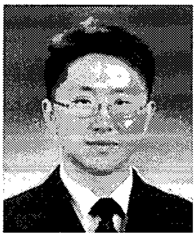
[11] W. Lian, W. W. Cheung, N. Mamoulis, and S. Yiu, "An Efficient and Scalable Algorithm for Clustering Xml Documents by Structure," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, Issue 1, pp. 82-96, Jan. 2004.

[12] F. De Francesca, G. Gordano, R. Ortale, and A. Tagarelli, "Distance-based Clustering of XML Documents," *Proc. First Int'l Workshop on Mining Graphs, Trees and Sequences*, pp. 75-78, Sep. 2003.

[13] Theodore Dalamagas, Tao Cheng, Klaas-Jan Winkel, and Timos Sellis, "Clustering XML documents using structural summaries," *Proc. EDBT Workshop on Clustering Information over the Web*, Mar. 2004.

임 태 우

2003년 연세대학교 컴퓨터과학과 졸업(학사)
2005년 연세대학교 컴퓨터과학과 졸업(석사)
2005년~현재 삼성전자 정보통신총괄사업부 근무
관심분야 : XML 스키마 클러스터링 및 통합



이 경 호

1995년 연세대학교 전산과학과 졸업(학사)
1997년 연세대학교 컴퓨터과학과 졸업(석사)
2001년 연세대학교 컴퓨터과학과 졸업(박사)
2001년 National Institute of Standards and Technology(NIST) 객원연구원
2002년~현재 연세대학교 컴퓨터과학과 조교수
관심분야 : 멀티미디어 문서처리, XML, 웹 서비스